# Fall detection and activity recognition using stereo low-resolution thermal imaging

Yannick Zoetgnande

UNIVERSITÉ DE RENNES 1

# THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE RENNES 1

ÉCOLE DOCTORALE N° 601
*Mathématiques et Sciences et Technologies de l'Information et de la Communication*
Spécialité : *Signal, Image, Vision*

Par

**Yannick Wend Kuni Zoetgnande**

## Fall detection and activity recognition using stereo low-resolution thermal imaging

**Thèse présentée et soutenue à Rennes, le 16 décembre 2020**
**Unité de recherche : Laboratoire Traitement du Signal et de l'Image (LTSI), UMR Inserm 1099**

**Rapporteurs avant soutenance :**

| | |
|---|---|
| Christine Fernandez-Maloigne | Professeure à l'Université de Poitiers |
| Mohamed Daoudi | Professeur à IMT Lille Douai |

**Composition du Jury :**

| | | |
|---|---|---|
| Président : | Antoine Manzanera | Professeur à l'Institut Polytechnique de Paris |
| Examinateurs : | Christine Fernandez-Maloigne | Professeure à l'Université de Poitiers |
| | Mohamed Daoudi | Professeur à IMT Lille Douai |
| | Mireille Garreau | Professeure à l'Université de Rennes 1 |
| | Vincent Gauthier | Directeur de Neotec Vision |
| Dir. de thèse : | Jean-Louis Dillenseger | Maître de conférences à l'Université de Rennes 1 |

# Résumé en français

## Motivations

Selon l'Institut National de la Démographie, le nombre de personnes âgées ne cessera d'augmenter et doublera d'ici 2050 (par exemple le nombre de personnes de plus de 75 ans passera de 6 millions en 2020 à 12 millions en 2050[1]). Dans ce contexte, mêmes si des infrastructures sont créées pour les accueillir, il devient essentiel de trouver des solutions permettant aux personnes âgées de vivre à leur domicile, le plus longtemps et avec le plus d'autonomie possible.

Dans ce cadre là, les chutes sont à surveiller tout particulièrement, puisqu'il s'agit de la première cause de mortalité, hors maladie, pour les individus âgés de plus de 75 ans [1] et représente donc un réel enjeu sociétal. Cette surveillance concerne deux aspects : la détection et la prévention des chutes. La détection de la chute permet de rapidement porter secours à l'individu, ce qui est d'autant plus important que la rapidité de prise en charge permet de limiter les conséquences de la chute qui vont s'aggravant avec le temps passé au sol, pouvant mener au décès de l'individu. La prévention permet de diminuer le nombre d'occurrences des chutes et de retarder l'apparition de la première chute. Dans notre cas, la prévention consiste à faire un suivi de l'activité de la personne et à en déduire des signes de fragilité par une analyse de l'évolution de cette activité.

Notre travail de Thèse s'est inscrit dans le cadre d'un projet financé par l'ANR : PRuDENCE (ANR-16-CE19-0015-01). PRuDENCE pour PRévention Et DEtectioN des ChutEs. Ce projet est une collaboration entre une société de vision, NeoTec-Vision (Pacé), le LTSI (Université de Rennes 1), l'ECAM Rennes (Bruz), l'UTT (Troyes) et l'université de Lille. L'objectif du projet PRuDENCE est de proposer un nouveau dispositif à bas coût à base de capteurs de profondeurs et/ou thermiques permettant de prévenir le risque de chutes par l'analyse de l'activité des individus. Ces types de capteurs ont été choisis car ils répondaient à différentes caractéristiques d'acceptabilité par les personnes surveillées (cf. chapitre 1): le dispositif est purement *passif*, ces capteurs permettent un *fonctionnement de jour comme de nuit* et assurent le *respect de l'anonymat* des personnes observées.

Notre travail de Thèse est directement lié à une des voies que voulait explorer le projet : la détection de chutes et le suivi d'activité à l'aide de caméras thermiques.

1. https://www.ined.fr/fr/tout-savoir-population/chiffres/france/evolution-population/projections/

Comme le prix du dispositif est un critère important pour le déploiement aux lieux de vie des personnes âgées, il a été choisi d'utiliser des caméras thermiques de bas-coût. Le principal inconvénient est toutefois que les caméras à faible coût ont une résolution d'image médiocre ($80 \times 60$ pixels dans notre cas), et que les images elles-mêmes sont pauvres en information. Les informations sont purement bidimensionnelles, or la détection de chutes et/ou le suivi d'activités nécessitent une estimation précise de la pose de la personne dans la pièce. Nous avons donc envisagé une solution de *détection de chutes et de suivi de l'activité à partir d'une paire stéréo de caméras thermiques basse résolution*.

## Système de vision

Afin de détecter les chutes, nous avons décidé de combiner deux caméras thermiques en mode stéréo. Mais si les caméras thermiques avec une bonne résolution sont relativement chères [2], récemment, certains fabricants ont proposé des caméras thermiques à très bas prix, comme par exemple le Lepton 2 de FLIR [2] que nous avons choisi dans notre étude. En contrepartie, ces caméras bas-prix ont une très faible résolution ($80 \times 60$ pixels), produisent des images bruitées avec des dérives en niveau de valeurs dans le temps (les caméras corrigent de temps en temps cette dérive ce qui a pour effet un saut temporel brutal des valeurs de l'image) et n'ont pas de calibration en température.

Le système d'acquisition est composé d'une paire de caméras FLIR lepton 2 (Fig.1). Le champ de vision horizontal est de $51°$ et le champ de vision diagonal est de $63.551°$. La fréquence d'images maximale des caméras est de huit images par seconde. Les deux caméras ne sont pas synchronisées. Elles sont placées en parallèle (leurs axes optiques sont parallèles) afin de favoriser le champ de vue. La distance entre les deux caméras (la ligne de base) a été définie à $16\,\mathrm{cm}$ afin de pouvoir inclure les deux caméras dans un boîtier pas trop encombrant. Le dispositif sera placé en hauteur (au plafond ou sur un mur juste sous le plafond) et dirigé de telle sorte à surveiller une pièce entière.

Les deux caméras sont pilotées à l'aide d'une carte micro-contrôleur adaptée et programmée par la société NeoTec-Vision. Cette carte permet de récupérer les images sur un PC par USB et dans le futur à faire l'interface avec la carte de traitement embarquée.

2. https://www.flir.com/globalassets/imported-assets/document/lepton-2.5-family-datasheet.pdf

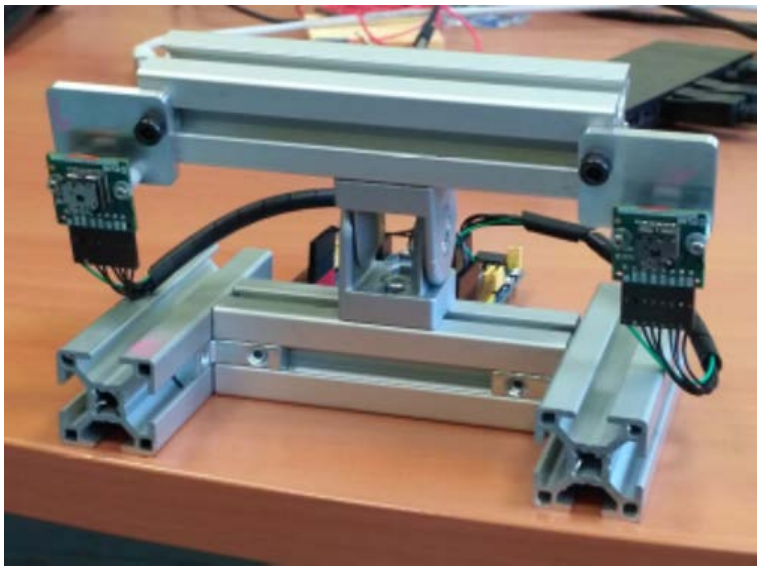


**Figure 1:** Le système stéréo composé de deux caméras Lepton 2 placées en parallèle.

## Organisation du travail de Thèse

L'objectif du projet étant double (1 - détection de chutes et 2 - monitoring de l'activité), nous avons décidé de décomposer notre étude en plusieurs sous-études (Fig. 2) :

1. Dans un premier temps, nous nous sommes intéressés à la *détection de chutes par reconstruction stéréoscopique*. Notre idée était d'adapter le pipeline classique de vision stéréoscopique aux spécificités de nos images thermiques. Cette étude a permis de mettre en place des solutions originales de calibration robuste [3], d'extraction et appariement de points caractéristiques [4] et une méthode de super-résolution [5], elle n'a pourtant pas réussi à atteindre le degré de précision nécessaire à la détection de chutes.
2. Nous avons donc opté pour une nouvelle *approche par apprentissage de détection de chutes à partir de la paire stéréo* mais ne nécessitant pas la reconstruction 3D de la scène.
3. Et, afin de répondre au deuxième objectif, nous avons menés quelques essais préliminaires pour l'*analyse par apprentissage de l'activité de la personne* (debout, couchée, marche, assise,...).

## Détection de chutes par reconstruction stéréoscopique

Notre idée première de détection de chutes était d'estimer la pose de la personne dans la scène 3D par stéréo-vision et comparer cette pose par rapport au niveau sol

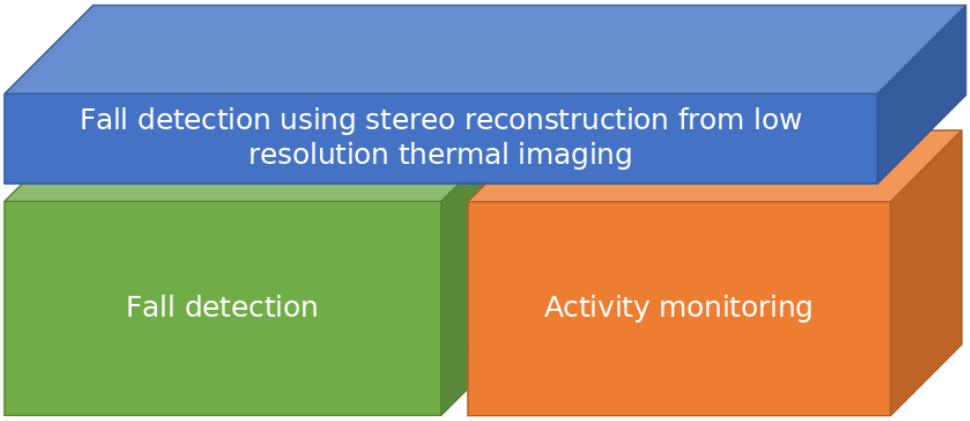



**Figure 2:** Organisation du travail de Thèse

de la pièce également estimé en stéréovision. Si par exemple le centre de gravité de la personne était proche du sol, la chute serait détectée.

Nous avons donc repris le pipeline classique de la stéréovision:

1. Calibration stéréo des caméras thermiques. Nous somme face à deux difficultés : d'une part, la mire de calibration doit être adaptée à l'information physique recueillie par ce type de caméra et, d'autre part, la faible résolution et le bruit inhérents à nos caméras thermiques bas-coût. Pour cela nous avons proposé une *méthode robuste de calibration de caméras thermiques stéréos basse-résolution* [3].
2. L'extraction d'amers dans les deux images de la paire stéréo. Les images thermiques sont très peu texturées et, de surcroît, le faible résolution rend flous les détails de l'image.
3. La mise en correspondance des amers communs aux deux images et l'estimation de la disparité entre ces amers. Là, également, la faible résolution de nos images rendait extrêmement imprécise l'estimation de la disparité. Afin de résoudre ce problème et le problème précédent, nous avons proposé une méthode d'*extraction et de mise en correspondance de points avec une précision sous-pixel* [4].
4. La reconstruction 3D qui doit nous permettre d'estimer 1) le plan du sol à partir d'amers appartenant ou étant placés sur le sol, et 2) la pose 3D de la personne à partir des vues stéréoscopiques.

   Dans ces deux cas, la précision de la reconstruction est directement liée à la précision des étapes précédentes.
5. Afin d'améliorer la précision des étapes de calibration, d'extraction et d'appariement d'amers, nous avons également proposé une méthode de *super-résolution* permettant d'agrandir les images par un facteur 4 tout en *préservant voire en améliorant par apprentissage les contours* contenus dans les images thermiques [5].

De manière classique, l'étape de calibration consiste à trouver les paramètres d'un modèle de calibration à partir de points réels donnés par une grille de calibration. Dans notre cas, nous avons opté pour le modèle de caméra pinhole et nous avons utilisé les fonctions de la librairie OpenCV afin d'estimer les différents paramètres du modèle. Toutefois, nous nous sommes retrouvés confrontés à deux problèmes : 1) la fabrication d'une grille de calibration adaptée aux propriétés physiques mesurés par le capteur et à sa résolution et 2) s'assurer que malgré la faible résolution de nos caméras, le processus de calibration soit robuste, c'est-à-dire précis et reproductible.

**Grille de calibration.** La grille de calibration doit donner une information de température. Comme le processus de calibration peut être assez long, nous avons opté pour la fabrication d'une grille dont les points sont chauffés de manière active. De manière plus détaillée, nous avons placé 36 ampoules d'automobiles, sous la forme d'une grille de $6 \times 6$, sur un panneau en bois (Fig. 3). Chaque ampoule est séparée de sa voisine d'une distance de $160\,\mathrm{mm}$, ce qui nous fait une panneau de $1\,\mathrm{m} \times 1\,\mathrm{m}$.

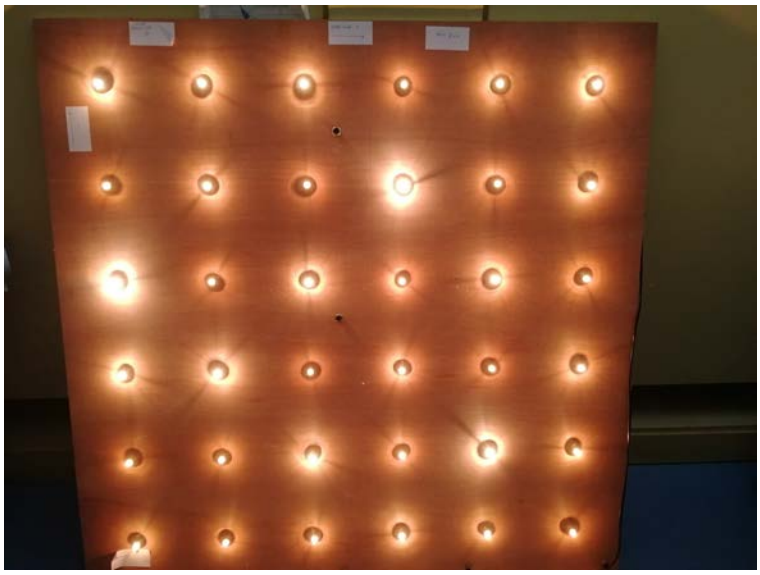


**Figure 3:** Grille de calibration

**Calibration robuste.** Cette robustesse va dépendre, d'une part, de la précision de l'estimation des points de la grille dans les images et, d'autre part, du nombre d'acquisitions différentes de cette grille.

Du fait de la faible résolution, nous avons remplacé le module d'extraction de points, précis au pixel près, proposé dans OpenCV par notre propre méthode où le centre des halos lumineux est localisé à une précision sous-pixel. L'autre paramètre qui va influencer la précision de la calibration est le nombre d'images de la grille utilisées pour estimer les paramètres du modèle. Un nombre trop petit a non seulement comme

conséquence une calibration peu précise mais également non-reproductible. Dans ce cas, les paramètres estimés par deux calibrations successives peuvent être assez différents. Par contre, un trop grand nombre d'images alourdit considérablement la manipulation lors de l'acquisition de la grille. Nous avons mené une étude de type bootstrap afin d'évaluer l'évolution de la précision et de la reproductibilité de la calibration en fonction du nombre d'images. Nous avons conclu que pour notre protocole, l'acquisition de 35 paires d'images était nécessaire et suffisante pour fournir une calibration précise et robuste.

### Extraction et appariement de points caractéristiques à une précision sous-pixel

Une fois les caméras calibrées, la vision 3D nécessite 3 étapes : 1) l'extraction de formes caractéristiques dans les deux images, 2) l'appariement des même formes pour en estimer la disparité entre les deux images, et 3) la triangulation pour estimer la position 3D en fonction de la disparité. Une bonne reconstruction est alors tributaire du nombre de points remarquables extraits des images et de la précision de l'estimation de la disparité entre les points vus par les deux images. Afin de maximiser ces deux critères, nous avons proposé le schéma suivant (Fig. 4) :

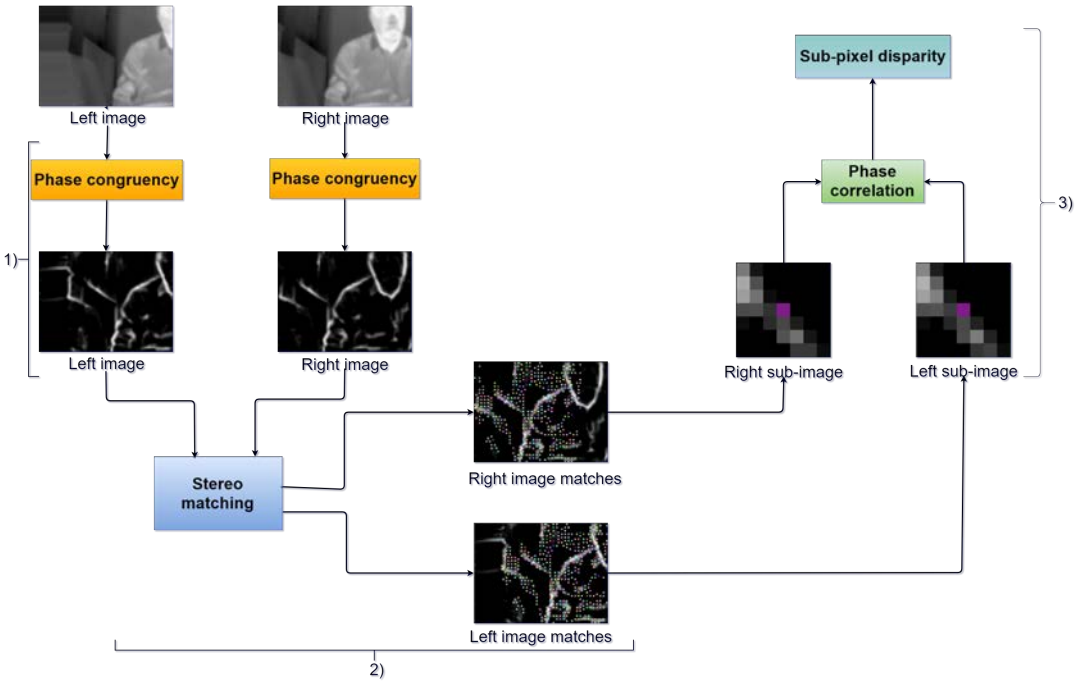



**Figure 4:** Suite de traitements pour l'appariement à une précision sous-pixel : 1) extraction robuste de points remarquables ; 2) appariement robuste à une précision du pixel ; 3) estimation plus précise de la disparité à une précision sous-pixel.

**Extraction de formes caractéristiques.** Comme nous l'avions mentionné précédemment, les images thermiques sont très peu texturées par rapport à des images de caméras

travaillant dans le spectre visible. Ils est donc beaucoup plus difficile d'extraire un grand nombre d'amers (coins, points de contours,...) de ces images. De surcroit, la faible résolution rend beaucoup plus flous ces amers, ce qui entraine une localisation beaucoup moins précise. La littérature nous donne plusieurs méthodes classiques pour les images RGB : le détecteur ed coins de Harris [6], KLT [7], FAST [8], BRIEF [9], la congruence de phases [10, 11]. Certains auteurs ont proposé d'adapter ou de simplifier ces méthodes pour l'extraction d'amers dans le cas d'images thermiques (mais sur des images de plus grande résolution). Hajebi et al. ont démontré dans leur papier que la congruence de phases pouvait extraire plus de points caractéristiques que les autres méthodes [12]. Nous avons donc exploré et adapté cette méthode à nos images.

En deux mots, la congruence de phases reflète le comportement de l'image dans le domaine fréquentiel. Il a été noté que les éléments de type contour ou coin ont plusieurs de leurs composantes fréquentielles dans la même phase. L'idée est alors de chercher la congruence de phases à différentes orientations et échelles. Nous avons opté pour la variante qui calcule les congruences à l'aide de filtre monogénique au lieu d'un banc de filtres log-Gabor [13]. Pour chaque pixel, un moment maximum est estimé par combinaison des différentes congruences. Un pixel ayant un moment maximal supérieur à un seuil $\gamma$ sera considéré comme point remarquable.

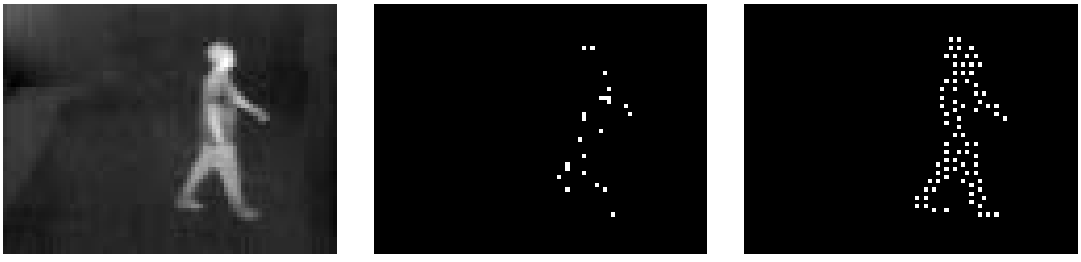
Une évaluation menée sur nos images de basse résolution a permis de démontrer que : la congruence de phases permettait d'extraire plus de points remarquable que les autres techniques classiques et que la congruence de phases était insensible aux variations temporelles brusques de l'intensité des images. Nous avons également constaté qu'en faisant varier la valeur du seuil $\gamma$, nous pouvions obtenir, soit des points remarquables robustes mais peu nombreux, soit beaucoup de points remarquables mais au détriment de la robustesse.

**Appariement des points remarquables.** La calibration des caméras permet de rectifier les images. L'appariement est alors simplifiée par la recherche de points similaires selon la ligne épipolaire. La similarité est estimée à l'aide de la mesure de similarité de Lades [14] menée dans une fenêtre $5 \times 5$. L'appariement est ensuite confirmé à l'aide de critères de cohérence entre les images (contraintes d'unicité ou d'ordre des points le long de la ligne, orientation similaire des amers, bijectivité des appariements entre vues droite et gauche,...). Nous avons évalué visuellement les appariements proposés pour 15 paires d'images et nous avons constaté moins de $1\%$ d'appariements incorrects.

**Estimation de la disparité à un niveau sous-pixels.** L'appariement permet d'estimer la disparité pour chaque paire de points à la précision du pixel. Cette précision est toutefois insuffisante car nous avons démontré que dans la configuration de notre montage, une erreur d'appariement de 1 pixel conduisait à une incertitude de plus de

$50\,\mathrm{cm}$ en profondeur. La corrélation de phase est une méthode classique d'estimation d'une translation par la recherche de la position du maximum du pic de corrélation. L'idée est alors de modéliser ce pic à l'aide d'un modèle (un sinus cardinal dans notre cas) et d'estimer la position du maximum à un niveau sous-pixel par l'ajustement du modèle aux données. Après avoir estimé la meilleure taille de fenêtre de calcul pour la corrélation, nous avons évalué notre méthode sur des images thermiques de haute résolution et sur les images acquises par nos caméras. Sur les images haute-résolution, plus de $99\%$ des points étaient appariés avec une précision inférieure à 0,5 pixels. Le taux d'appariement diminuait toutefois si l'on souhaitait une précision plus grande : $90\%$ de pixels appariés avec une erreur inférieure à 0,25 pixels, $55\%$ avec une erreur inférieure à 0,1 pixels et $33\%$ avec une erreur inférieure à 0,05 pixels. Nous avons observé un comportement identique sur nos images basse-résolution avec des taux d'appariement de $97\%$, $83\%$, $55\%$ et $34\%$ pour respectivement des erreurs inférieures à 0,5, 0,25, 0,1 et 0,05 pixels.

**Reconstruction 3D.** Comme nous n'avions pas de vérité terrain nous avons simplement comparé la reconstruction 3D des points extraits, appariés et reconstruits par triangulation en utilisant notre méthode (congruence de phases et corrélation de phase) avec ceux reconstruits en utilisant des fonctions plus classiques fournies par OpenCV (extraction de point par ORB car réputée robuste [15], appariement par k plus proches voisins -KNN- du fait de la faible abondance des points puis triangulation) (voir Fig. 5). Nous constatons clairement que notre méthode reconstruit plus de 3D que la méthode classique. Si l'on regarde la distribution de la profondeur en $z$ des points 3D reconstruits pour une personne qui se trouve à environ $3\,\mathrm{m}$ des caméras : ORB+KNN situe les points à $1523,94 \pm 1612,76\,\mathrm{m}$ alors que notre méthode donne $3648,10 \pm 256,43$ $\mathrm{m}$. Notre mesure semble plus précise et fournit des points avec moins de dispersion.



(a) Image thermique, vue de gauche

(b) Points 3D reprojetés (ORB+KNN) (#24)

(c) Points 3D reprojetés (notre méthode) (#80)

**Figure 5:** Points 3D reprojetés sur un plan image.

Nous avons également appliqué cette méthode à des points chauds placés au sol. L'idée était alors de déterminer le plan du sol afin de détecter les chutes par l'analyse de la distance entre les points extraits des personnes et le plan du sol. Malheureusement, même si notre méthode donnait des points 3D plus précis, la dispersion des points

était trop importante pour estimer le plan du sol de manière robuste.

### Super-résolution

Une solution pour remédier à cette faible précision induite par la reconstruction stéréo consiste à augmenter la taille des images, ce qui a pour effet de diminuer la taille du pixel et donc devrait permettre de localiser et apparier plus précisément les points remarquables dans les images. Un tel processus, appelé super-résolution, est un problème inverse mal posé. En effet, un nombre infini d'images haute-résolution peut correspondre à une même image basse-résolution.

Lorsqu'une seule image est utilisée au cours de ce processus, on parle de super-résolution d'image unique (SISR). Parmi les différentes méthodes de super-résolution, celles basées sur l'apprentissage nous paraissaient les plus adéquates car elles permettent d'ajouter l'information manquante perdue lors de l'acquisition basse-résolution. Lors de ma thèse, j'ai eu l'opportunité d'effectuer une mobilité de recherche de 4 mois, à Ryerson University, Toronto, Canada. Lors de cette mobilité, j'ai exploré une méthode de super-résolution basée sur des réseaux de neurones convolutifs.

Notre objectif a été de développer une méthode permettant d'augmenter la résolution d'un facteur 4 dans les deux directions ($80 \times 60 \longrightarrow 320 \times 240$) tout en améliorant le niveau de détail de l'information utile à la reconstruction stéréo, les contours présents dans l'image dans notre cas. Les questions ont été alors : 1) Quel est le réseau le mieux adapté à notre problématique ? 2) Comment inciter le réseau à s'intéresser explicitement aux contours ? et 3) Comment faire l'apprentissage du réseau alors que nous ne disposions pas de paires d'images basse/haute-résolutions de la même scène ?

Plutôt que d'utiliser un réseau qui a déjà été appliqué sur des images thermiques (thermal enhancement network TEN [16] ou CNN with skip connections [17]), nous avons choisi de nous inspirer de l'architecture Residual Dense Block (RDN) qui a fait ses preuves sur de la super-résolution d'images du spectre visible [18] (Fig. 6).

Nous voulions également mettre en saillance les contours contenus dans l'image afin que le réseau les traite de façon spécifique. Pour cela, nous avons placé en début de réseau des modules d'extraction de contours (Sobel, Kirsch et Laplacien de Gaussiennes) et injecté cette information en parallèle de l'image dans le réseau.

L'apprentissage était le point crucial de notre méthode. En effet, du fait du prix des caméras thermiques haute-résolution, nous ne disposions pas de paires d'images basse/haute résolutions pour l'apprentissage. Nous avons donc pris des images haute-résolution de personnes en intérieur d'une base de données proposée par [19]. Nous avons alors créé des images basse-résolution $I_{lr}$ en application un

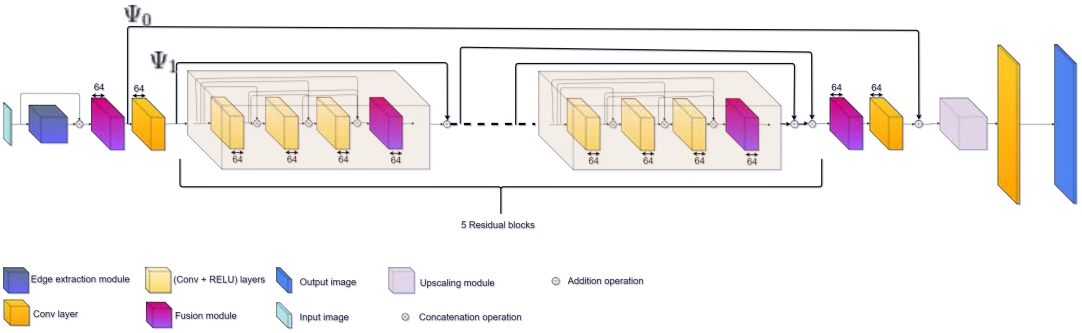



**Figure 6:** Edge Focused Thermal Super-resolution (EFTS)

modèle de dégradation aux images haute-résolution $I_{hr}$ :

$$I_{lr}(m,n) = d\left(h\left(I_{hr}\left(x,y\right)\right)\right) + \sigma\left(m,n\right)$$

avec $h$ une fonction de filtrage flou (filtre Gaussien), $d$ l'opérateur de sous-échantillonnage et $\sigma$ du bruit additif (bruit Gaussien). Nous avons fait de l'apprentissage aveugle en créant des images basse-résolution à partir des images hautes-résolution en faisant varier de manière aléatoire les variances du filtre Gaussien et du bruit Gaussien. Ceci nous garantissait un apprentissage robuste, non lié à une caméra particulière.

Notre méthode a été évaluée sur une autre base de données que la base d'apprentissage. Les résultats qualitatifs et quantitatifs (Peak Signal to Noise Ratio -PSNR-, Structural Similarity Index -SSIM- et Edge Preservation Index -EPI-) ont permis de démontrer que notre méthode avait de meilleures performances que les autres méthodes de super-résolution.

De même, la méthode appliquée sur nos propres images produit des images haute-résolution avec des détails plus fin (Fig. 7).

## Détection de chutes par apprentissage

Comme il a été mentionné précédemment, les méthodes classiques de stéréovision basées sur la reconstruction 3D n'ont pas permis de reconstruire le plan du sol, ni d'estimer la pose de la personne à partir de nos paires d'images thermiques basse-résolution. Ceci, malgré le fait que nous avions élaboré une méthode de calibration robuste, et une extraction de formes et un appariement à un niveau de précision sous-pixel.

Nous avons alors pensé à une autre stratégie : est-il possible de définir la relation entre un point 3D et ses projections sur les deux images de la paire stéréo, non de manière analytique par un modèle, mais par apprentissage ?

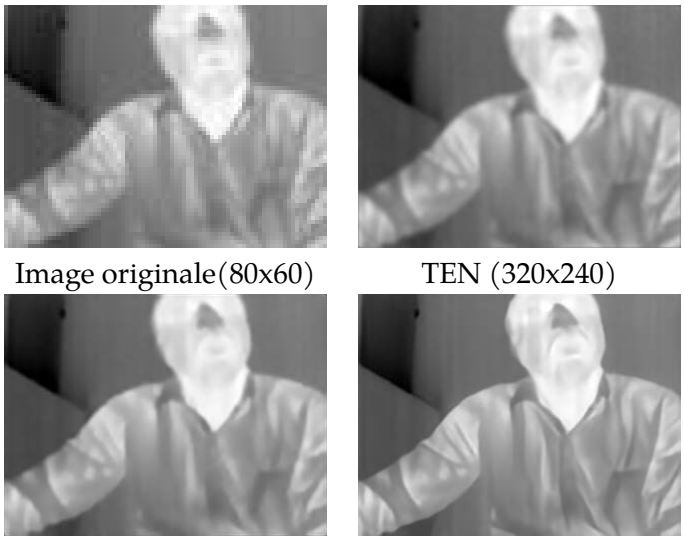



**Figure 7:** Super-resolution de l'image d'une personnes assise en face de la caméra thermique basse-résolution

L'idée est alors d'utiliser un réseau de neurones convolutif ou un classifieur plus classique et de lui apprendre si un point 3D vue par les deux images de la paire est au sol ou non.

La procédure de détection de chutes sera alors la suivante (Fig. 8 ) :

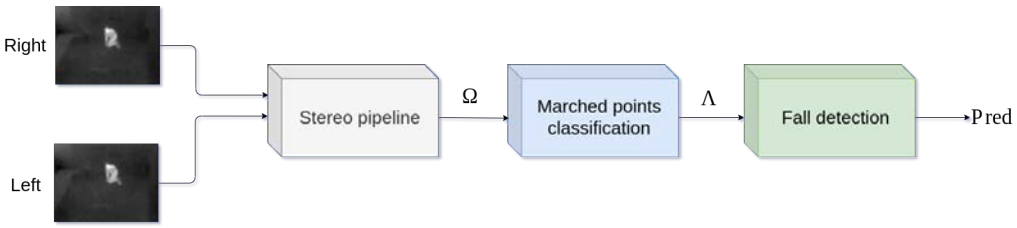



**Figure 8:** Procédure de détection de chutes

1. Une procédure de mise en correspondance. Nous avons repris la procédure décrite dans le chapitre "Détection de chutes par reconstruction stéréoscopique" avec :
   (a) Une extraction de points remarquables par congruence de phases ;
   (b) Une simple segmentation de ces points pour ne garder que les points appartenant potentiellement à une silhouette. Cette segmentation est aisée sur les images thermiques du fait de la chaleur dégagée par les personnes ;
   (c) L'appariement de ces points remarquables par simple mesure de similarité et quelques critères de cohérence entre les images. Il est noter que, comme nous n'avons pas calibré les caméras, nous ne pouvons pas utiliser la géométrie épipolaire. Par contre, du fait de la pauvreté en information

des images thermiques, il y a finalement relativement peu de points à apparier.

2. La classification des points appariés en ”point 3D au sol”/”point 3D au-dessus du sol” par inférence du réseau ou du classifieur ;
3. La détection de la chute par une analyse des points classifiés.

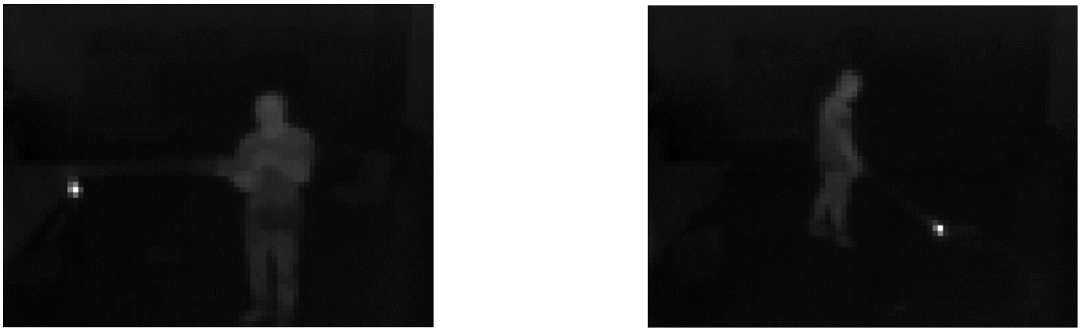
Il est a noter que nous faisons une détection de chutes basée sur l'analyse d'images statiques. Nous ne faisons pas intervenir une analyse temporelle à ce niveau.

Les points clés de la procédure sont alors le choix du réseau ou du classifieur, l'apprentissage de ce réseau/classifieur et l'analyse des points classifiés.

**Données pour l'apprentissage.** L'objectif est de savoir si une paire de points appariés est sur le sol ou au-dessus. Pour cela, nous avons déplacé un point chaud (une ampoule allumée) sur le sol et dans l'espace de la pièce au-dessus du sol et nous avons acquis les images correspondantes (Fig. 9). La lampe est facile à segmenter car c'est généralement l'objet le plus ”lumineux”. Nous avons donc une première base d'apprentissage composée de paires d'images de la lampe associées soit à une classe ”point 3D au sol” soit à une classe ”point 3D au-dessus du sol”.

Afin de simplifier l'apprentissage, nous avions également extrait la position 2D des ampoules dans les images (centre de gravité en niveau de gris dans une ROI autour des ampoules). Nous avons donc une seconde base d'apprentissage composée de paires de coordonnées 2D associées soit à une classe ”point 3D au sol” soit à une classe ”point 3D au-dessus du sol”.

Notre stratégie d'apprentissage nous a permis de disposer d'un ensemble de données équilibré contenant autant de points au sol que de points non au sol.



(a) Au-dessus du sol (b) Sur le sol

**Figure 9:** Vue gauche de l'image d'une lampe

**Choix du réseau ou du classifieur.** Nous avons exploré deux stratégies en fonction de l'information sur les points remarquables : paire de coordonnées 2D ou paire d'images thermique.

Si les points 2D se présentent sous la forme de leurs coordonnées 2D dans les images, un simple classifieur de type SVM suffit.

Si les points 2D se présentent sous la forme d'images thermiques, nous avons proposé un modèle basé sur l'apprentissage profond, inspiré de DenseNet (DGD) [20]. Cette solution a l'avantage d'associer les caractéristiques de l'image thermique (résolution, bruit, halo, . ..) dans le processus d'apprentissage.

**Analyse des points classifiés.** Pour une personne au sol, nous supposons qu'une grande partie des points remarquables sera classée "au sol". De même, pour une personne debout, assise, voire allongée dans un lit ou sur un canapé, seule une faible partie des points remarquables (ceux des pieds par exemple) devraient être classés "au sol". Un simple seuil sur le pourcentage de points classés "au sol" devrait suffire pour détecter ou non la chute.

**Résultats.** Pour l'apprentissage, nous avons acquis 6000 images de la lampe au sol et 6000 images de la lampe au-dessus du sol. Cette acquisition se fait en moins d'une demi-heure car il suffit de déplacer la lampe sur le sol ou dans l'espace avec les caméras en mode vidéo (8 images/sec). Une telle acquisition est à faire une fois pour une configuration donnée de caméras.

La performance de notre détecteur de chutes a été testée sur quatre bases de données. Chaque base de données était composée d'images acquises sur une personne différente. Sur ces images les personnes étaient dans une des configurations suivantes : debout, marchant, assises, couchées sur un lit ou un canapé ou tombées sur le sol. Ces bases de données n'étaient pas équilibrées dans le sens qu'il y avait relativement peu de chutes par comparaison aux autres activités.

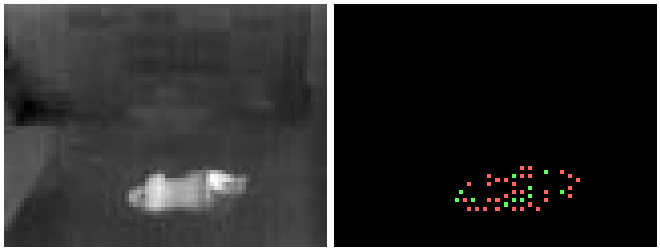
Dans un premier temps, nous avons comparé nos deux stratégies de classifieurs : SVM sur coordonnées 2D et DGD sur données image. Pour cela nous avons appliqué nos classifieurs sur les données d'apprentissage sous la forme de 5 réplications d'une validation croisée à 2 blocs (5x2cv). Pour chaque réplication, nous avons mélangé les données de manière aléatoire. Cependant, nous avons appris et testé le SVM et le DGD avec respectivement les mêmes données. La valeur médiane du taux de classification de DGD (0,976) surpasse un peu celle de SVM ( 0,965) ($p = 0,00028$). La prise en compte de l'information image apporte un léger gain de précision de classification. Cependant, ce gain semble un peu marginal par rapport à la complexité de mise en œuvre de la DGD.

Nous avons alors réglé les paramètres de notre méthode (le seuil de congruence de phases qui agit sur le nombre de points remarquables et le seuil de décision sur les points classés) sur des données de personnes. Comme nous connaissions la vérité terrain "chute/pas chute", nous avons pu évaluer et optimiser les différents réglages

en terme de sensibilité, spécificité, précision et score-F1. Les résultats obtenus ont montré de très bonnes performances avec des scores supérieurs à 0,9 pour les différents indicateurs.

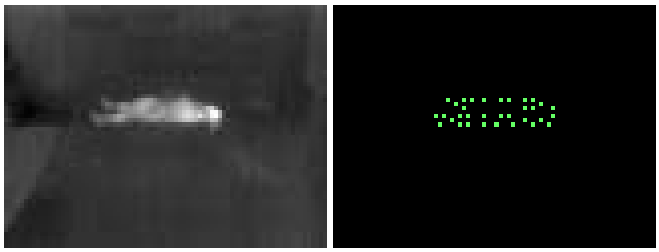
Nous avons également comparé de manière informelle nos scores avec ceux donnés dans la littérature pour les méthodes [21] et [22] qui étaient appliquées sur des images thermiques. Nos scores sont supérieurs aux leurs mais sur des données différentes.

Par contre, nous avons constaté que notre méthode était capable de discerner une personne couchée, d'une personne au sol. Ce cas est généralement difficile à discriminer pour des méthodes mono-caméra, même basées sur de l'apprentissage (Fig. 10).



(a) Personne après une chute au sol.

(b) 76.66% de points classés au sol.



(c) Personne couchée sur un canapé).

(d) 0% de points classés au sol.

**Figure 10:** Différence entre une personne au sol (a et b) et une personne couchée sur un canapé (c et d)

## Reconnaissance d'activité

Ce chapitre présente l'état de nos réflexions et les travaux que nous avons menés sur la détection de l'activité de la personne à partir de la paire de caméras thermiques en stéréo à l'aide d'apprentissage profond. Une de nos difficultés vient du fait que nous n'avons pas assez de données pour entraîner un réseau. En effet, à notre connaissance, personne n'a essayé d'estimer des activités à partir d'images thermiques, et a fortiori à partir de paires d'images thermiques basse-résolution. Par contre, nous avions accès à une base de données publique mise à disposition par Tran

*et al.* [23] et axée sur la détection de chutes avec des modalités comme l'image de profondeur, l'accéléromètre et des images RGB. Nous n'avons gardé que cette dernière modalité. Pour cette modalité, Tran *et al.* ont mené des acquisitions sur 50 sujets avec 7 points de vue et 20 activités différentes.

Nous avons alors voulu essayer 4 stratégies différentes pour estimer l'activité de personnes sur des images thermiques à partir d'un apprentissage mené sur des images RGB :

1. Apprentissage sur des contours RGB et inférence sur les contours des images thermiques $(Model_n)$.

   Nous avons considéré que les cartes des contours dans une image RGB et dans une image thermique devaient être globalement identiques (aux contours de textures près). L'idée est alors d'extraire les contours (Sobel et/ou Laplacien) sur les images RGB et d'entrainer le réseau sur ces données de contours. Lors de l'inférence, il suffit d'extraire les contours de l'image thermique et de fournir cette information au réseau.

2. Apprentissage sur du flot optique RGB et inférence sur le flot optique des images thermiques $(Model_{flow})$.

   Cette approche est assez proche de la précédente, sauf que nous avons entrainé le réseau sur le flot optique extrait des images RGB. L'inférence se fait alors sur le flot optique extrait de l'image thermique.

3. Apprentissage sur des images thermiques simulées à partir des images RGB $(Model_s)$.

   Pour cela nous avons entrainé des réseaux adverses génératifs (GAN model) à simuler des images thermiques à partir d'images RGB. Nous avions à notre disposition des paires d'images fournies par CAMEL [24], LITIV 2017 [25] et Bildeau [26]. Cela nous a permis d'essayer les simulations suivantes : RGB vers thermique, contours image RGB (Sobel) vers thermique, RGB vers contours image thermique et contours image RGB vers contours image thermique. Le réseau de reconnaissance d'activité est alors entrainé sur les images thermiques ou contours image thermique simulés à partir de RGB.

4. Généralisation de domaine $(Model_{dm})$.

   Dans ces classes de méthodes, l'idée est d'apprendre un modèle à partir de plusieurs domaines observés afin de le rendre performant sur tout autre domaine non vu. Classiquement, un premier réseau apprend à extraire des caractéristiques robustes des images quel que soit le domaine d'acquisition. Ce réseau est alors suivi d'autres réseaux de classifieurs de tâches et classifieurs de domaines. Dans notre cas, nous avons repris l'idée d'Albuquerque et al. [27] mais en prenant une autre architecture de réseau.

Par contre, nous avons dû adapter la généralisation de domaine du fait des données déséquilibrées. Nous disposions d'un grand ensemble de données de vidéo RGB et d'un tout petit ensemble de données de vidéo d'images thermiques que nous ne pouvions pas diviser en ensembles de données d'apprentissage et de test. Nous avons décidé d'augmenter artificiellement le nombre de domaines en fournissant les vidéos RGB de départ, deux domaines fabriqués par l'application du filtre de Sobel sur les images RGB plus ou moins floutées et deux domaines fabriqués par l'application du fitre Laplacien sur les images RGB plus ou moins floutées. Comme domaine cible nous avons choisi les images thermiques après application du filtre de Sobel.

Comme nous n'avions pas d'a priori sur les réseaux, nous avons testé deux architectures, I3D et R(2+1)D, que nous avons dû fortement adapter pour traiter nos images thermiques.

Nous avons alors entrainé les deux réseaux en fonction des 4 stratégies sur les images de la base de données de Tran *et al.* mentionnée précédemment [23]. Nous n'avons retenu que 8 activités (marche, saisie d'objets par terre, assis puis se lever, chute de la position debout, ramper, debout et chute, assis et chute, couché et chute).

Nous avons alors comparé les résultats de détection d'activité de ces 4 stratégies avec les deux réseaux sur une base de données d'images thermiques acquises par nous (caméra gauche pour éviter la redondance) avec 3 sujets et les 8 activités décrites ci-dessus. La comparaison s'est faite sur l'analyse visuelle des plans obtenus par t-SNE (t-distributed stochastic neighbor embedding) et des matrices de confusion. Nous avons également mesuré la précision et le score F1 de ces différentes variantes. Globalement nous avons constaté que le réseau R(2+1)D donnait de meilleurs résultats que I3D. De même, ce sont les stratégies $Model_n$ (Apprentissage contours image RGB et inférence sur contours image thermique) et $Model_s$ (La variante apprentissage sur images thermiques simulées à partir d'images RGB) qui semblaient donner les meilleurs résultats.

Ces résultats ne sont que préliminaires, un examen plus détaillé des différents modèles doit être poursuivi.

## Conclusion

Dans cette thèse, l'objectif était de mettre en œuvre une solution de détection de chutes et de suivi d'activité à partir d'une paire de caméras thermiques basse-résolution.

Dans un premier temps, nous avons suivi une stratégie classique de vision par ordinateur basée sur la reconstruction 3D de la scène. Pour cela nous avons développé

une solution de calibration robuste de caméras thermiques stéréo, une méthode d'extraction et d'appariement à une précision sous-pixel de points remarquables et une méthode de super-résolution avec saillance sur les contours. Par contre le degré de précision atteint n'était pas suffisant pour une détection de chutes.

Dans un second temps, nous avons proposé une solution de détection de chutes par apprentissage profond. L'originalité de la méthode est que le réseau n'apprend pas les chutes mais la position de points dans l'espace, ceci sans calibration des caméras. Les résultats obtenus surpassent ceux de techniques de la littérature appliqués sur des images similaires.

Finalement nous avons effectué une étude préliminaire sur la reconnaissance d'activité par apprentissage profond. Nous avons exploré plusieurs stratégies qui permettaient de contourner le fait qu'il n'existe pas de données stéréo thermiques suffisantes pour faire l'apprentissage. Les résultats préliminaires semblent prometteurs.

# Acknowledgements

First of all, I would like to thank my parents Zoetgnande Dimitri and Zoetgnandé Martine who have always supported me. They have always been there for me and believed in me when I wanted to do research. I would also like to thank my brothers and sisters in particular: Laure Bouda, Gladis Dipama and Franck Zoetgnandé. I also thank my wife Christelle Sorgho who has always been there for me and who encouraged me in difficult moments. I can only thank my dear friends Stéphane Beogo and Benjamin Thionbiano for their assistance.

Secondly, I would like first of all to thank Professor Christine Fernandez-Maloigne and Mohamed Daoudi for dedicating their time and expertise to the revision of this manuscript. I am also grateful to Professors Antoine Manzanera and Mireille Gareau for their interest in this work, and I greatly appreciated the constructive discussion that followed the presentation of my thesis. I also want to thank Mr. Vincent Gauthier for his support.

I would like to express my deep and sincere gratitude to my research supervisor in the LTSI laboratory: Jean-Louis Dillenseger, associate professor. Not only their precious criticism, advice and close supervision play an important role in the progress of my research, but their support, availability and kindness made this life experience unforgettable.

This project would not have been possible without the help and financial support of ANR PRuDENCE. I would like to thank the partners of the project, specially Geoffroy Cormier.

I would also like to thank my colleagues from LTSI for their help: Pablo Alvarez (alias Gringo), Soumaya Msaad, Bertille MetMontot, Hellene Feuillâtre. Special thanks also go to my friends from the laboratory Pierre-Antoine Chantal, Marouane Arrais and Majd Saleh.

*Dedicated To my parents, sisters, brother and wife.*

# Table of Contents

# List of Figures

# List of Tables

# Introduction

According to the National Institute of Demography, the number of seniors will continue to increase and will double by 2050 ([3]). In this context, even if specialized institutes such as retirement homes are created to support them, it becomes essential to find solutions allowing the seniors to live at home, as long as possible and with as much autonomy as possible.

In this context, falls should be monitored in particular, since it is the leading cause of death, apart from illness, for individuals over 75 years old [1] and therefore represents a real societal issue. This surveillance concerns two aspects: detection and prevention of falls. The detection of the fall makes it possible to quickly bring rescue the person. This point is crucial because the rapidity of the intervention limits the consequences of the fall as much as possible. It is well known that the time spent on the ground increases the consequences that can lead to death. Prevention makes it possible to reduce the number of occurrences of falls and to delay the onset of the first fall. In our case, prevention consists of monitoring the activity of the person and deducing signs of fragility by analyzing the evolution of this activity.

This work was part of a project funded by the ANR: PRuDENCE (ANR-16-CE19-0015-01). PRuDENCE stands for **PR**évention **E**t **DE**tectio**N** des **C**hut**E**s. This project is a collaboration between a computer vision company NeoTec-Vision (Pacé), LTSI (University of Rennes 1), ECAM Rennes (Bruz), UTT (Troyes), and the University of Lille. The objective of the PRuDENCE project is to propose a new low-cost device based on depth and/or thermal sensors, making it possible to prevent the risk of falling by analyzing the activity of individuals.

First, we implemented a classic computer vision strategy based on a 3D reconstruction of the scene. For this, he had to develop original solutions for robust camera calibration, extraction and matching with sub-pixel precision of remarkable points, and super-resolution with salience on the contours. On the other hand, the degree of accuracy achieved was not sufficient for the detection of falls.

Secondly, we proposed a fall detection solution through deep learning. The originality of the solution is that the network does not learn the falls but the position of points in space, without calibrating the cameras. The obtained results surpassed those of techniques applied to similar cameras.

At the end of the thesis, we tried several strategies for recognizing activities

3. https://www.ined.fr/fr/tout-savoir-population/chiffres/france/evolution-population/projections/

through deep learning. These strategies circumvented the fact that there is not sufficient thermal stereo data for training. The first results seem promising.

The thesis is structured as follows. The Chapter 1 describes the state of the art about fall detection. Before detecting or preventing fall, we wanted to correctly define what is a fall. Besides, to provide an appropriate device, it is important to answer to questions: what is a fall?, why seniors are falling?, where seniors are falling?, what are the consequences of seniors falls? and how to prevent these falls? The Chapter 2 details our main contributions and explain our choices. The Chapter 3 describes the device we used for fall detection and activity monitoring. Chapter 4 details the features extraction method, the stereo-matching process, and the sub-pixel matching algorithm. Then, the Chapter 6 uses some of the solutions described in Chapters 3 and 4 in order to estimate the fall or not of a person by to 1) learn if a point in on the ground and not and 2) to detect if features of a person are on the ground or not. Finally, the Chapter 7 investigates different solutions to deal with the lack of annotated thermal dataset. In this chapter, we compare four solutions respectively based on training on visible edges, then inferring on thermal edges, training on simulated thermal images, then inferring on real thermal edges, training on visible optical flow then inferring on thermal, optical flow and finally domain generalization.

# 1 If fall detection is the answer, what is the question?

-'What if I fall?', Tim cried.
-Maerlyn laughed. 'Sooner or later, we all do.'

Stephen King, The Wind Through the Keyhole

## 1.1 Introduction

In most developed countries, the share of the population over the age of 65 years is increasing. This aging sometimes results in a loss of autonomy and increasing incidents such as falls. Many researchers have already tackled the problem of preventing the consequences of falls. But, we will try to go oppositely to find out the real questions that arise when a fall occurs [28]. Nowadays, in most of the developed countries, the population is becoming older and older. This part of the community is confronted with isolation and weakness. Such weakness can lead to serious injury when a fall occurs. In retirement homes, there are 2 to 3 falls per person per year [29].

We think that questions are as important as answers, in particular in research. Finding the right questions lead us to find the right answers. Once Albert Einstein said: "If I had an hour to solve a problem and my life depended on the solution, I would spend the first 55 minutes determining the proper question to ask, for once I know the proper question, I could solve the problem in less than five minutes." Indeed finding the right question allow us to have a global vision of the problem to solve.

This chapter will try to figure out the main questions that arise before solving Fall detection issues. Even if it obvious, we will see that sometimes the definition of a fall is not clear.

## 1.2 What is a fall?

There are many definitions of falls. One of them is "Any unintentional change in position where the person ends up on the floor, ground, or other lower-level; includes falls that occur while being assisted by others" from [30]. From this definition there are many terms that are important: *unintentional*, *position* and *floor, ground, or other lower level*. Unfortunately, this definition is not clear. As pointed by the authors in

[31], the definition of fall is different from the researcher community to seniors and health care providers. While researchers consider the above definition, seniors and healthcare providers are more focused on the consequences of falls when defining it. In [32], the authors point out this disparity and suggested that it is crucial to define a fall to avoid confusion. Sometimes the words slip, trips, and falls are used to define the same concept. These latter events can have the same consequences, even if they do not respect the definition of [30]. There are thus several other definitions of a fall:

- "*A fall is an event which results in a person coming to rest inadvertently on the ground or other lower level and other than as a consequence of the following: Sustaining a violent blow, loss of consciousness, Sudden onset of paralysis, as in a stroke, an epileptic seizure.*" [32]
- "*A fall is an unexpected loss of balance resulting in coming to rest on the floor, the ground, or an object below knee level.*" [33]
- "*A fall is an unintentionally coming to rest on ground, floor, or other lower-level; excludes coming to rest against furniture, wall, or other structure.*" [34]
- "*A fall is Any involuntarily change from a position of bipedal support (standing, walking, bending, reaching, etc.) to a position of no longer being support by both feet, accompanied, by (partial or full) contact with the ground or floor.*" [35]
- "*A fall is losing your balance such that your hands, arms, knees, buttocks or body touch or hit the ground or floor.*" [36]
- "*A fall an unintentional change in position where the elder ends up on the floor or ground.*" [37]
- "*A fall is inadvertently coming to rest on the ground or other lower level with or without loss of consciousness and other than as the consequence of sudden onset of paralysis, epileptic seizure, excess alcohol intake or overwhelming external force.*" [38]
- "*A fall is a sudden loss of gait causing the hit of any part of the body to the floor.*" [39]

In [31], Zecevic et al. conduct telephone interviews with seniors and in-person meetings with three physicians, three pharmacists, one optometrist, two community-care coordinators, and four therapists (occupational, physical, recreational, and rehabilitation), eleven nurses, five personal support workers, and two health administrators. There were two main questions: "How would you describe or define a fall?" and "What do you think are the three main reasons for falling?". They found that there are two definitions of falls: antecedent-based and consequence-based. Antecedent explains why, where, and how, while consequences explain the body position or injury. The authors noticed that even if there are some similarities in seniors' and researchers' definitions of falls, seniors are more focused on antecedents and consequences.

In our opinion, all these definitions are correct. But to define a fall effectively in the context of seniors, we must correlate it with the senior's background. We need

to go deeper to see how dangerous a given fall is. The danger of the fall is linked to the questions as follows: Why is a senior falling?; Where is he falling?; and Is there a solution to prevent it?

## 1.3 Why

In [31], Zecevic et al. interview health care providers and seniors to define the main reasons for falling. Researchers focused on muscle weakness, history of falls, gait deficits while seniors pointed balance, weather, inattention, and medical conditions. The responses of health providers are closer to those of seniors. Indeed, they put forward reasons like medical conditions, stability, medications, and indoor obstacles. Among these three groups, two common reasons stand out: vision and balance. Thus, the disparity between the answers is not limited to the definition of a fall but also concerns the fall's reasons.

There are two types of risk factors: intrinsic risk factors and extrinsic risk factors. Intrinsic risk factors include demographic (age, gender), system (balance, strength, vision, and cognition), and disease (dizziness/vertigo, cardiovascular disease, dementia, and depression). As for extrinsic, they include medications, home, and footwear.

In [40], Ambrose et al. report the main risks that can lead to falls. The risk factors are neurological, neuromuscular, osteoarticular, and visual. We can summarize them as:

— Age: Age is one of the main risks of falls. According to Bergen, in [41], seniors are likely to experience falls in their daily activities. Based on interviews and surveys, they show that in 2014, 28% of elderly in the United States reported having had at least one fall in the previous 12 months. In most cases, aging is associated with the apparition of diseases such as muscle problems, cardiovascular disease, diabetes, arthritis, etc. [42].

— The balance and gait are considered one of the most critical risk factors of falls [43]. The worsening of gait and balance can be linked-to usual aging effects or pathological aging [44], [45]. Regarding the usual aging effects, seniors may face decreased coordination and muscle strength and the loss of body-oriented reflexes. This can lead to a trip or slip [46]. In [46], the authors perform experiments to disturb the balance of 16 younger (21 to 35 years) vs. 19 older (68 to 88 years) adults. They find that the later ones are more likely to fall.

Regarding the pathological aging effect, they are linked to neurological gait pathologies such as hemiparetic, frontal, Parkinson, neuropathic, etc. [47]. In [47], the authors study 632 seniors to determine the neurological gait abnormalities. They found that subjects with neurological problems have more risk to fall than other subjects.

— Vision: Vision is essential for balance control and obstacle avoidance. If the vision is not clear, such an issue can lead to distance misjudgment of spatial information. In [48] Salonen and Kivelä analyze the available state of the art data and conclude that there is a correlation between recurrent falls and eye disease. In [49], Freeman et al. show that falls likelihood is increased with both central visual impairment and peripheral visual impairment.

— Cognition: In [50], Yogev-Seligmann et al. show that walking and gait is a complex process covering many fields such as physiology, biomechanics, brain mapping, physics, and neuropsychology. Gait is associated with the executive function (EF), referring to higher cognitive processes. These processes use and modify information from many sensory systems. With aging, the executive function worsens, and the attention (the ability to dual-tasking) decreases.

— Cardiovascular diseases: Cardiovascular diseases can be hypotension, hypertension, or atrial fibrillation. Hypotension is affecting 30% of people over 65 and, more importantly, 70% of people in nursing homes [40]. In [51], Hausdorff et al. show that there is a link between blood pressure and risk of falls. Regarding atrial fibrillation, in [52], Sanders et al. perform experiments on 442 patients over 65 and analyze *accidental* and *non-accidental* falls. They find a higher prevalence of non-accidental falls for patients having atrial fibrillation.

— Medications: Medications can also a risk factor of falls. In the state of the art, studies report psychotropic [53], diabetes medications [54], nonsteriodal anti-inflammatory [55] and anti-epileptics [56] which have a direct impact on the prevalence of falls.

— Depression: Depression is common to many diseases such as strokes, Parkinson, Alzheimer's, dementia [57, 58]. In [59], Paleacu et al. show through a small study that major depressive disorder (MDD) can influence gait speed, dual tasking, and executive function.

— Environment: Environment is considered as an extrinsic factor. Poor lighting and furniture (combined with visual impairment) can increase the risk of falls. Footwear is also a significant cause of falls. In [60], Menant et al. rank type of shoes accordingly to risk of fall. They show that slippers have a higher fall risk, followed by barefoot/fastened shoes. Athletics shoes and canvas shoes present a lower risk of falls. They also show that, unfortunately, most of the time, seniors like to wear slippers. They finally show that the design of shoes is essential. Indeed, shoes with heels more than 2.5 cm are associated with a high risk of falls.

## 1.4 Where

To analyze falls, it is also essential to know where falls occur most of the time. In [61], Kochera compiles data from the National Health Interview Survey. He finds that 55% of falls occur at home in the house, 23% close to the house, and 22% away from home. Regarding the places where falls occur, he notices that 43% of them are at ground level while 14% of them occur on stairs, 11% on the sidewalk, and 9% from chair/bed.

In [62], Gill et al. monitor a cohort of 1088 persons for three years. The experiment is conducted with people living in buildings without stairs. They find that 10% of falls occur in hallways, and for those happening at home, the distribution is 13% in bathrooms (toilets), 19% in kitchens, 30% in bedrooms, and 31% in living rooms.

In [63], Le Dain and Cormier analyze the localization of fall in two institutes for seniors: a retirement home named Nymphéas and a hospital named Polyclinique Saint Laurent. Regarding Nymphéas, falls occur most of the time in rooms (65%) and the bathroom (15%). There is no fall outside in the reported results, which suggests that seniors do not come out of their rooms very often. These results are similar to those recorded at Polyclinique Saint Laurent. Indeed, they find 71% of falls occur in the room, 15% in the bathroom, and 6% in other places.

By analyzing these studies, we can say that most of the time for seniors dwelling with the community, falls occur at home (55%), and among falls at home, most of them (80%) occur in kitchens, bedrooms, or living rooms. Regarding retirement homes or hospitals, most of the falls happen in the bedroom and the bathroom.

## 1.5 Consequences

There are physical and psychological consequences when a fall occurs. A large number of studies have been conducted on the physiological consequences (injuries) of falling. In [64], Stokes and Lindsay show that from 1983 to 1992, falls were the leading cause of hospitalization for people over 65.

Fractures are the most common consequences of falls. Many types of fractures may arise from a fall: hip fracture is the most serious and frequent one. In [65], Kannus et al. state that in the 1990s, there were 250 000 hip fractures per year in the USA, costing $5.4 billion. They also find that hip fracture is more frequent for women than for men. But falls can also lead to trunk, neck, upper limbs, humerus, chest, broken knee, and Cervical Spine fractures. Falls can introduce physiological effects such as death or morbidity, functional decline, inactivity, functional dependency, loss of autonomy, depression, loss of self-confidence, and self-efficacy loss.

Falls can cause significant psychological and social consequences. When a

senior falls, she develops a fear of falling, and many authors show that there is a correlation between falls and fear of falling [66–68]. In [69], Friedman et al. observe 2 212 seniors (from 65 to 84) during 20 months to establish a temporal relationship between these two syndromes. They find that there is a strong correlation between these two phenomena. The fear of fall can reduce mobility, muscle weakness, and so increase fall risks [70].

There are also financial consequences for seniors, families, and governments. There are many studies to calculate the cost of falls, but it isn't easy to compare. In [71], Florence et al. analyze the data from Medicare Current Beneficiaries Survey (MCBS), and they state that, in the USA, non-fatal falls cost $50 billion per year while fatal falls cost $754 million. In the USA, 30 million seniors fall each year, resulting in 30 000 deaths, and each year 3 million seniors are treated for a fall injury [72].

Regarding Europe, in [73], Alekna et al. perform a survey on 878 community-dwelling women during 12 months in Vilnius (Lithuania). They find that 35.3% of women have felt during this period. Among these falls, 90.3% of falls cause injuries, while 15.3% cause bone fractures. They calculate that a fall cost 254 € per patient. According to the Public Health Outcomes Framework (PHOF), falls were the ninth highest cause of disability-adjusted life years (DALYs) in England. The cost of falls each year is 4.4 billion € [74]. Regarding France, the situation is comparable to these European countries. Indeed the cost is around 2 billion € per year [75].

The consequences of falls are innumerable, not only for the senior's mental and physical health but also for the economies of countries. It is, therefore, essential to find solutions to mitigate the consequences of these falls.

## 1.6 How to detect or to prevent?

Prevention of falls is essential to reduce the consequences. The intuitive solution can be to mitigate fall risk factors(as mentioned in Section 1.3). So most of the time, by detecting these risk factors very early, we may prevent falls. This prevention can be done through the analysis of orthostatic balance and posture, activity analysis, gait analysis, and intervention action therapeutic (Fig 1.1). If detection can be dissociated from prevention, most of the time, detection is part of prevention. This is why, in this section, we will treat detection and prevention jointly. Fall detection is, most of the time, a part of activity analysis. Indeed, in many studies, fall is considered as an activity to detect.

### 1.6.1 Solutions

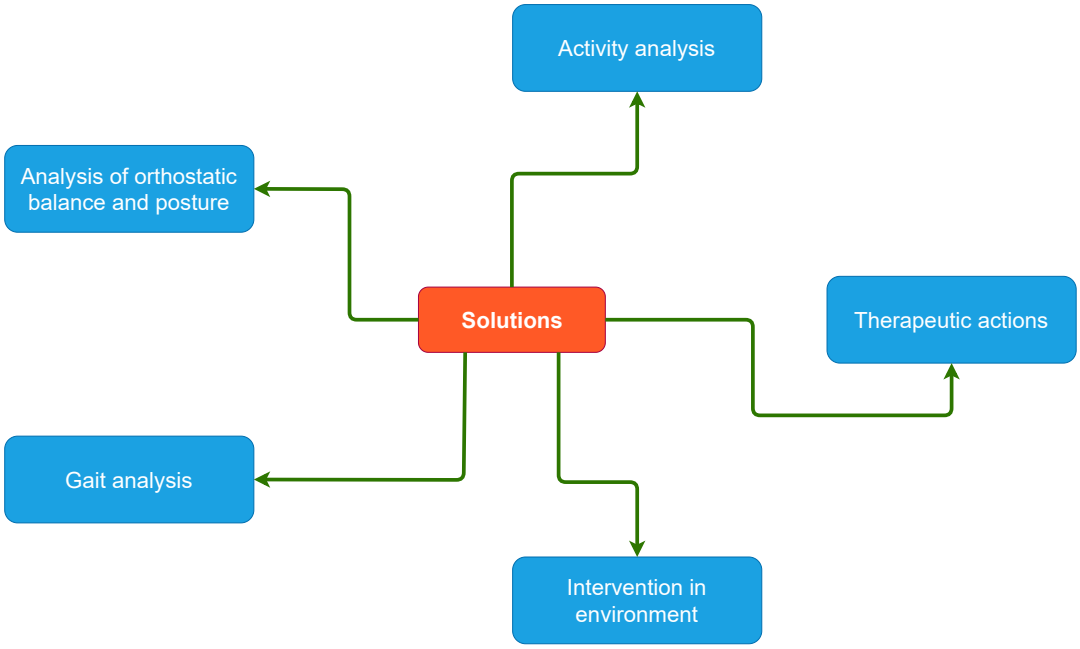



**Figure 1.1:** List of solutions to detect or prevent a fall

#### 1.6.1.1 Therapeutic actions

When intrinsic risk factors have been identified through orthostatic balance and posture analysis, activity analysis, and gait analysis, it is sometimes necessary to contain these risk factors through therapeutic actions. One of them is muscle strengthening through physical activity. Most of the time, these exercises are part of a broader fall prevention program. In [76], Shimada et al. conduct an observation of thirty-four frail seniors before and after rehabilitation. They find that there were some improvements in certain tests such as One Leg Standing, Functional Reach, and Functional Balance after exercises.

#### 1.6.1.2 Intervention in environment

Environmental hazards are also one of the fall risk factors. By re-organizing the room, it may be possible to mitigate fall risk. Falls may be prevented by:

— removing rugs

— making the room tidier by removing obstacles

— avoiding steps that are too steep or too long

— lighting well the rooms (not too dim nor too bright)

— avoiding pets and, unstable chairs and tables

— by not setting the toilet seats too low

In [77], Gillespie et al. show that there were fewer falls when removing the environmental hazards.

#### 1.6.1.3 Analysis of orthostatic balance and posture

The orthostatic balance and posture analysis are linked to intrinsic risk factors such as balance, vision, and cognition. These ones can ultimately weaken muscles and degrade balance. In [78], Delbaere et al. conduct a prospective cohort study of 500 seniors for 12 months to see the risk factors that can be used to predict a fall. With these selected risk factors, it could be possible to prevent falls. They use classification tree analysis [79] to discriminate factors. They perform many assessments, such as:

— Physical assessment through a Physiological Profil Assessment (PPA) [80], which estimates the fall risk through many parameters (vision, proprioception, quadriceps strength, simple reaction time, ...). They also include a one-leg balance test for 10 seconds.

— Cognitive assessment: using Trail Making Test (TMT) with language skills, memory performance, and visuoconstructional ability

— Psychological Assessment: using depression symptoms and anxiety symptoms.

— Disability, Physical Activity, and Quality-of-Life assessment: they also include levels of disability and quality of life.

After counting the number of falls for each senior, they analyze how each of these factors is related to the observed fall. During the experiments, 166 seniors fell while 50 had a low risk of falling (PPA $< 0.6$). After selecting impairment in balance-related as the first partitioning variable, two factors emerge as the most important to understand why a fall occurs. The first is a general disability as the model emphasizes that seniors with low physiological issues and no disability are those who felt the least. The second is coordinated stability.

The analysis of the postural attitude also helps to prevent falls. Indeed, a worsening of the postural attitude can lead to a sudden deterioration of the individual's cognitive capacity. Such risk factor can be evaluated through a standardized geriatric assessment [81, 82] to construct a Frailty Index (FI). This assessment contains some measurements such as Berg balance scale [83], uni-pedal stance [84], Sensory organization test [85], Functional Reach test [86], Tinetti balance scale [87], Time up and go [88] and Stop Walking when talking [89].

With these assessments, it should be possible to follow each senior during months and prevent falls when these measures are worsening.

#### 1.6.1.4 Activity analysis

In addition to assessing risk factors, daily activity analysis can also provide indicators for fall prevention. Indeed, most fall risk factors also influence a senior's routine. For example, weaker muscles lead the senior to avoid certain activities. Through activity analysis, it could be possible to detect dangerous activities. Activity

analysis is most often carried through activity recognition using smart devices, smart homes, and machine learning algorithms. Activity analysis can be done from two viewpoints: the senior posture (sitting, walking, falling, etc.) or her mobility rate (how many times the senior stayed lying, etc.).

By analyzing the senior's daily activity, the main idea is to assess her ability to carry out daily activities such as cooking, walking, etc. When this study is carried out over a long time, the worsening of frailty may manifest itself in an abrupt cessation of activity, motor impairment, or cognitive impairment. Such a task is linked to anomaly detection [90]. Indeed, the main idea is to point out certain changes in the subject's routine. In [90], Duong et al. use Switching Hidden Semi-Markov Model to identify deviation in the daily routine of a person with high/low-level activities. They define the routine as entering-the-room & making-breakfast, eating-breakfast, washing-dish, making-coffee, reading-morning-newspaper & having-coffee, and leaving-the-room. To recover activities, they placed four cameras in the ceiling corners. Then, they established some routines using sequential successions of these activities. Their model can classify sequences and detect anomalies.

#### 1.6.1.5 Gait analysis

Contrary to activity analysis, gait analysis focuses only on how seniors walk and detect anomalies by analyzing a long-term data set. In [91], Blanke and Hageman compare the gait of twelve young men and twelve older men through three 14-meters walking sequences. Using eight gait characteristics, they find that both groups have similar gait characteristics except for the stride, which is more significant for younger adults. These experiments have been conducted for healthy people.

In [92], Toulotte et al. perform a single leg balance test and gait parameters analyzing to exhibit the relationship between falling and these parameters. The subjects were composed of forty women (21 fallers who were 70.4 years (+6.4) and 19 non-fallers aged 67.0 years ($\pm$4.8)). To get gait parameters, they use Vicon 370 System [93] with six cameras and place markers on subjects' bodies. In the context of dual tasks, they find a strong relationship between the fall and some gait parameters such as cadence, walking speed, stride time, step time, and single support. But they were not a significant difference for a single task. These results reinforce the idea that the gait should be analyzed during daily activities. Indeed, during an ordinary day, seniors may be brought to perform many activities while walking. Unfortunately, most of the time, it is impossible to install a Vicon System in every senior's room. As for [91], less expensive sensors can be used.

In [94], Jiang et al. use a mobile phone and a three-axis accelerometer to get gait features that can be used to prevent falls. They define a gait model based on the assumption that a normal gait should be cyclic. They compute a similarity measure

between cycles. In [95], Senden et al. perform a study to compare the Tinetti scale (which is subjective) with gait characteristics as fall risk measures. Gait characteristic has been retrieved using an accelerometer. Experiments have been performed on one hundred subjects, 50 with the risk of falling and 50 without. Tinetti scale is an assessment of fall risk using visual gait and balance. They prove that it is necessary to combine algorithmic-based gait assessment to subjective gait assessment such as Tinetti.

### 1.6.2 Which solution?

As we have mentioned in Section 1.6.1, there are many solutions to detect or prevent falls. Some solutions, such as the analysis of the orthostatic balance and posture or therapeutic actions, require long-term proximity of doctors. However, in France (as in many countries), the geographical distribution of doctors is uneven. As pointed out by [96], in some regions such as Corse or Franche-Comté, there are less than two geriatricians per inhabitant. When going more in-depth, in certain small towns, there are no geriatricians. Thanks to sensors, geriatricians might follow up with their patients even hundreds of kilometers away. When there are no geriatricians close to the population, all the solutions we presented in section 1.6.1 can be performed using sensors. When there is a geriatrician, she can use sensors as an aid to support or invalidate her diagnoses.

Anyway, all these solutions might be used together. For example, sensor-based activity analysis will not be efficient if there too many environmental hazards in the senior's room. Many sensors can be used to perform gait analysis or activity analysis, but it is not clear which one is the best.

## 1.7 Sensors

Most of the fall prevention methods are based on analyzing results obtains from seniors' daily living activities. Activities can be monitored using sensors. We will investigate each type of sensor (advantages and drawbacks) and determine which one is the most suitable for our context.

### 1.7.1 Type of sensors

They are three main categories: wearable sensors [97], ambient sensors [98] and vision-based sensors [99]. Based on this classification into three categories, we made a rough taxonomy of the different sensors (Fig 1.2). In this section, we will present these sensors and how they are used for fall detection.

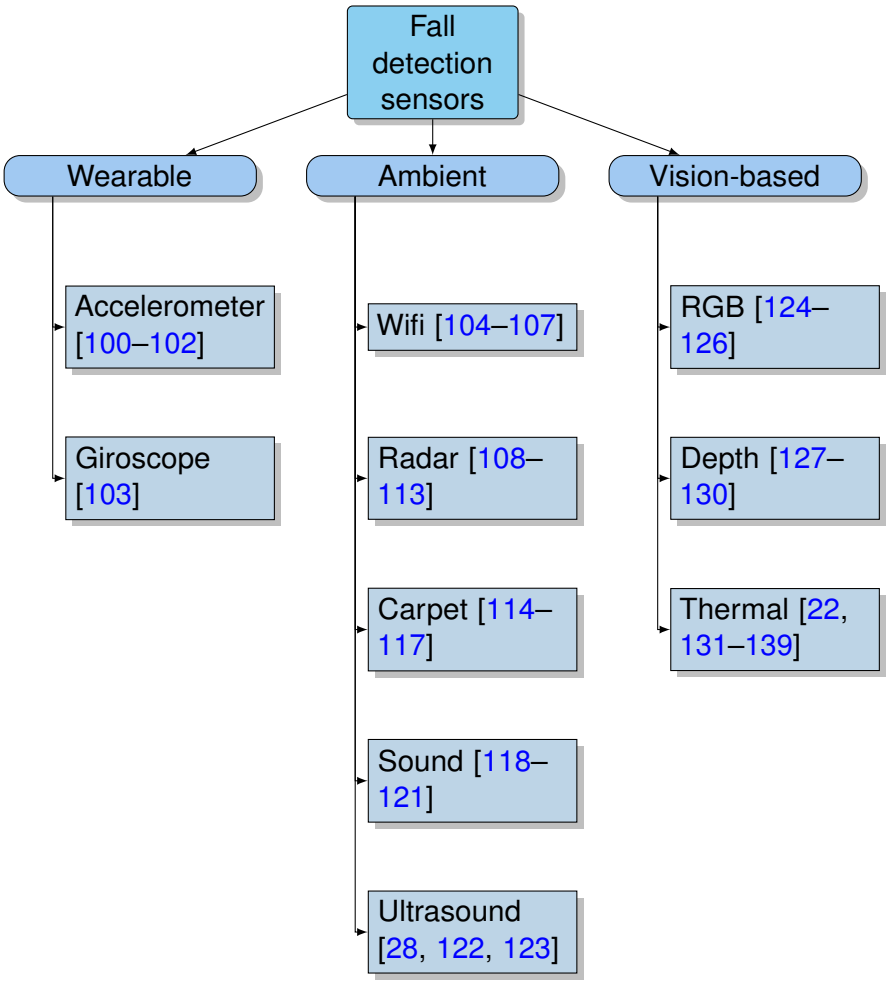



**Figure 1.2:** Light taxonomy of fall detection using different sensors

#### 1.7.1.1 Wearable sensors

Wearable sensors are sensors that are worn by the people like watch, necklace, included in clothes. Firstly, wearable sensors have been developed as a panic button [140] on medical alarm devices. Thanks to integrating embedded sensors and wireless communication devices, they have been widely used in medicine to monitor patients' physiological activities. It has become a way for doctors to monitor the health of their patients. Portable sensors are now manufactured to be very light so that they can be worn easily. The use of mobile sensors is no more specific to the medical field as it is currently gained great popularity thanks to sports [141].

In the context of assisted living for seniors, recently, wearable sensors have been introduced to monitor some activities [142–144]. Their performance is boosted by the rapid improvements of microelectronics and micromechanics, so now there are many wearable sensors with low energy consumption and less processing resources. One way to monitor seniors' activities is body temperature. Human body temperature can give some clues about various health conditions such as stroke or heart attacks

[145]. Heart rate is also an indicator of health conditions. Heart rate can be retrieved by Photoplethysmography (PPG) [146] or sound-based sensor [147].

For fall detection, one of the most widely used wearable sensors is the accelerometer. In [100], Bourke et al. evaluate a threshold-based method to detect falls using a tri-axial accelerometer. They also want to determine the best place to set the sensor. They test two positions: on the trunk or the thigh. They collect data from actions performed by 20 subjects (ten young subjects and ten seniors subjects). Within these actions, the subjects perform four types of falls: forward, backward, left, and right. They find that placing the accelerometer on the trunk gives better results by classifying 480 actions.

In [101], Kangas et al. also evaluate many positions (waist, head, and wrist) to see which position gives better results. They also take into account the same type of falls as [100]. They find that placing the accelerometer on the head is very efficient. Unfortunately, most seniors will not accept such a location. Based on the observation that most fall detection methods in the literature are evaluated on simulated fall data, Bagalà et al. show, especially in the context of the accelerometer, that most fall detection methods give poor results in the real case [102]. They apply 13 techniques from the literature on real-world data and found 3 to 85 false alarms per day, depending on the methods.

As accelerometers, the gyroscope can also be used for fall detection. For example, in [103], Bourke et al. acquire 480 movements and classify these actions using three thresholds.

Sometimes, to increase the model's performance, it is possible to combine accelerometer and gyroscope [148]. Nowadays, accelerometers tend to be widely adopted. This is because current smartphones contain such systems. In [149], Dai et al. are the first to propose a phone-based fall detection system. Accelerometers can be used for activity detection (fall) [144], subject's movement analysis or postural orientation [150].

#### 1.7.1.2 Ambient sensors

Ambient sensors are sensors placed in the person's familiar environment. Even if the vision-based sensors are also placed in the senior's environment, we will study them separately in the section.

Ambient sensors are usually used to monitor the whole room or a house. This kind of sensors can be classified into three categories: those that detect motion (WiFi [104–107], Radar [108–113, 151], ultrasound [28, 122, 123]), those that record the ambient sound [118–121] and those that record pressure and which are placed on the floor (smart carpet) [114–117].

**WiFi :** The main idea is to use some sensors that can implicitly or indirectly detect and analyze the motion. With this in mind, Palipana et al. had the idea to analyze the WiFi stream, specifically features extracted by Short Time Fourier Transform (STFT), and propose a sequential forward selection algorithm to detect the falls [105]. Their method comprises three modules: data collection and prepossessing features extraction and classification using SVM. Their SVM classifier has an accuracy (80%) even if the environment changes.

In [104], Zhang et al. use the CSI level (Channel State Information) of WiFi to detect falls. They use both phase and amplitude of CSI. Their method was also composed of three modules: pre-processing, segmentation, and fall detection. During pre-processing, they successively apply interpolation and low pass filter to CSI stream. Then during segmentation, they select fall-like activities then segment the activity stream. Finally, the last pipeline comprises features extraction followed by an SVM classifier and a fall alarm. The precision of their method is 89%. In [107], Yang et al. use similar modules as [104, 105], and get a detection precision of 89% and a false alarm rate of 8%.

**Radar:** Like WiFi, Radar can also be used to detect falls. In [152], Van Dorp and Groen propose a method to quickly extract features to estimate human motion parameters using a radar sensor. They use the Boulic's model [153], which describes human motion with three parameters.

In [108], Liu et al. use an inexpensive fall detection system based on Doppler Radar. They employ Mel-Frequency Cepstral Coefficients (MFCC) to detect activities such as walking, bending down, and falling distributed as 109 falls and 341 no falls. Their method obtains good performance with the Area Under the Receiver Operating Characteristics (AUROC) of 0.91.

In [154], Kim and Ling propose to classify human activity using a MicroDoppler. They select seven activities. Twelve subjects perform these activities collected with a Doppler Radar. They extract features using the short-time Fourier transform (STFT), and then an SVM classifier is used. In [111], Jokanovic et al. use deep learning-based methods to detect Radar falls. Their approach is composed of three modules: pre-processing, features extraction, and classifiers. They show that using a deep learning-based method gives better results than a PCA-based method.

**Ultrasound and sound:** Sensors based on sound and ultrasound can also be considered as ambient sensors. In [118], Zigel et al. propose a fall detection based on floor vibration and sound sensing. To detect falls, they perform signal processing and pattern recognition to discriminate between falls from other events. Their method produces results with a sensitivity of 97% and a specificity of 98.6%. The main problem

of sound sensors is the high number of false alarm. Indeed, simple falling furniture could be considered as falls.

In [120], Popescu et al. propose a method called FADE to reduce false alarms. They use a vertical sound sensor that can locate the origin of the sound. They discard false alarms by discriminating sound located at a height higher than 2 feet. This method allows them to reduce the hourly rate of false alarms from 32 to 5. To discriminate pets, they incorporate a motion detector.

In [28], Dobashi et al. propose a fall detection system using ultrasound sensors. They set two sensors on the ceiling of a bathroom, considering that many falls occur in this place. They use discriminant analysis to classify data between intention data (sitting) and accident data (falling).

**Smart carpet:** Since, most of the time, a fall occurs when a person is on the floor, many authors propose to detect falls using smart carpet. In [114], Chaccour et al. propose a smart carpet with differential pressure sensors. They obtain a sensitivity of 88.8% and a specificity of 94.9%.

In [115], Aud et al. also propose a smart carpet which does not require an external power supply or batteries. They collect data set from 11 volunteers walking or falling on the carpet. In [116], Muheidat et al. use a carpet containing 128 sensors. They acquire data from 10 volunteers and classified it using the Weka Classifiers [155].

#### 1.7.1.3 Vision-based sensors

Vision-based sensors are widely used in many fields. For fall detection, vision-based sensors can be used either for body shape analysis, inactivity detection, head motion analysis, or activity recognition.

There are many types of vision sensors: visible cameras [1], depth cameras, and thermal cameras. Visible sensors are common cameras that retrieve images from visible electromagnetic radiation. Depth sensors are used to determine the distance from any object to the given depth sensor. Thermal cameras, they collect the infrared electromagnetic radiation.

The most commonly used sensors are visible cameras. They are cheaper, can record a huge amount of information, and their setup is straightforward. Here are some examples of fall detection algorithms based on visible cameras. In [156], Solbach et al. propose a Convolutional Neural network-based method to estimate the human pose using stereo cameras. Given an image pair, they perform stereo processing, 2D human pose detection, 3D ground plane detection, 3D pose calculation, and fall

1. by visible cameras, we mean cameras in the visible spectrum

detection. Due to privacy issues and inability to monitor during the night, they have tended to be abandoned for real-world applications in favor of depth and thermal cameras.

In [157], Dubois et al. compare three gait parameters retrieved using a pressure carpet with these recovered by their method based on a depth sensor. They ask eleven subjects to walk, and they compute the step length, the step duration, and the walking speed. They show that the results obtained with their method are very close to those obtained using the carpet.

As far as thermal cameras are concerned, there is a lot of work in the literature. In [131], Vadivelu et al. propose a fall detection system based on thermal images. Their method, based on optical flow, also uses Spatiotemporal Interest Point (STIP) and Fisher vector framework. In [132], Hayashida et al. propose a real-time fall detection method using thermal images. They place their camera on the ceiling. In [22], Nogas et al propose to detect falls as anomalies. Thus, they compare three deep learning-based methods: DAE (Deep auto-encoder), CAE (Convolutional Auto Encoder), and ConvLSTM-CAE (LSTM autoencoder). They find that CAE offers the best results showing the importance of 3D convolutions for video processing (we will talk more about 3D convolution in Chapter 7). Unfortunately, most of these work concern only fall detection and does not involve activity recognition. We think that the monitoring of senior's activities should be done through a more fine-grained framework. In [158], Tao et al. propose to use low resolution infrared sensors ($8 \times 8$) for activity recognition. Their framework is based on three steps: sequence pre-processing (background subtraction and frame re-sampling), features extraction extraction (Temporal and Spatial features) and finally classification. In the next section, we will select a sensor and explain why.

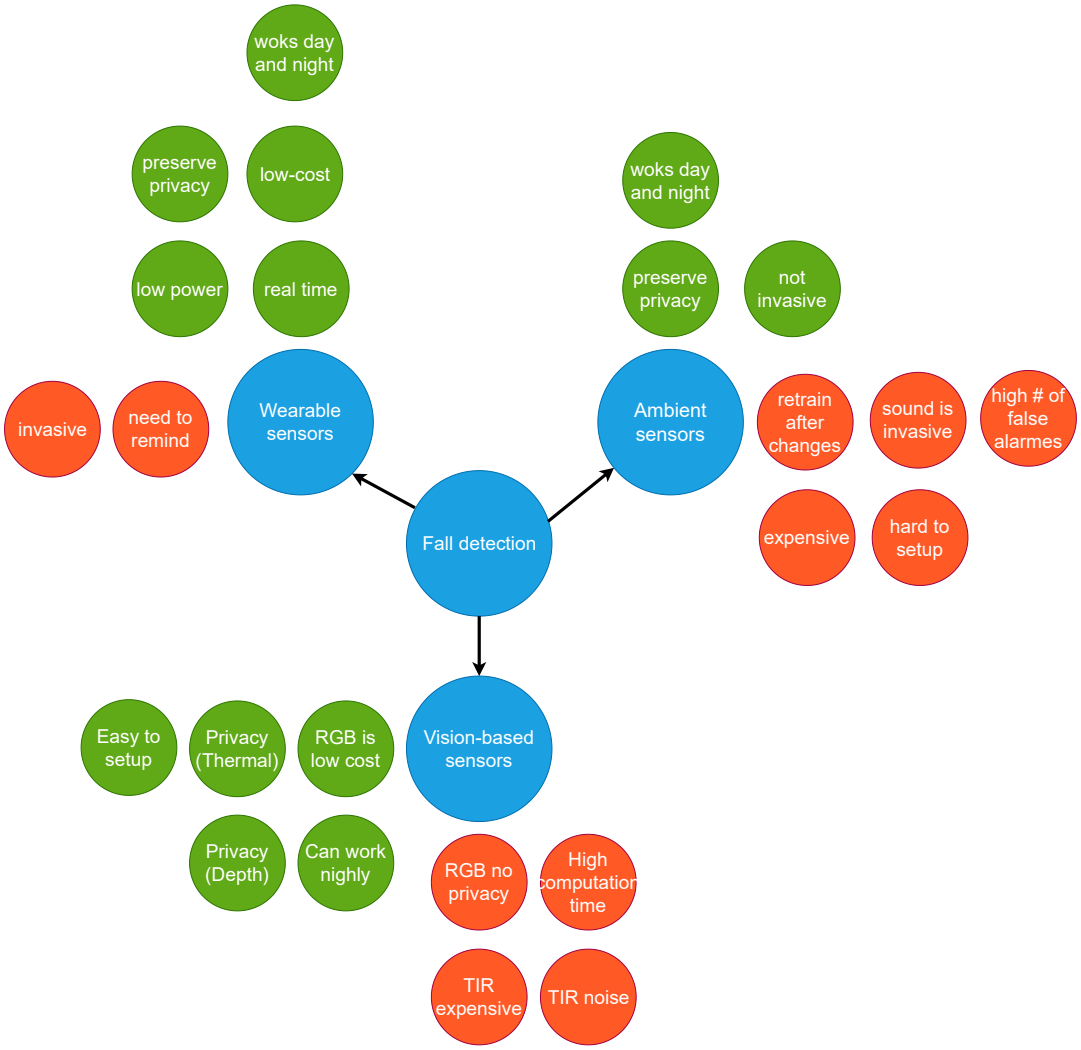



**Figure 1.3:** Advantages and drawbacks of fall detection sensors (TIR : Thermal infrared)

### 1.7.2 Which sensor?

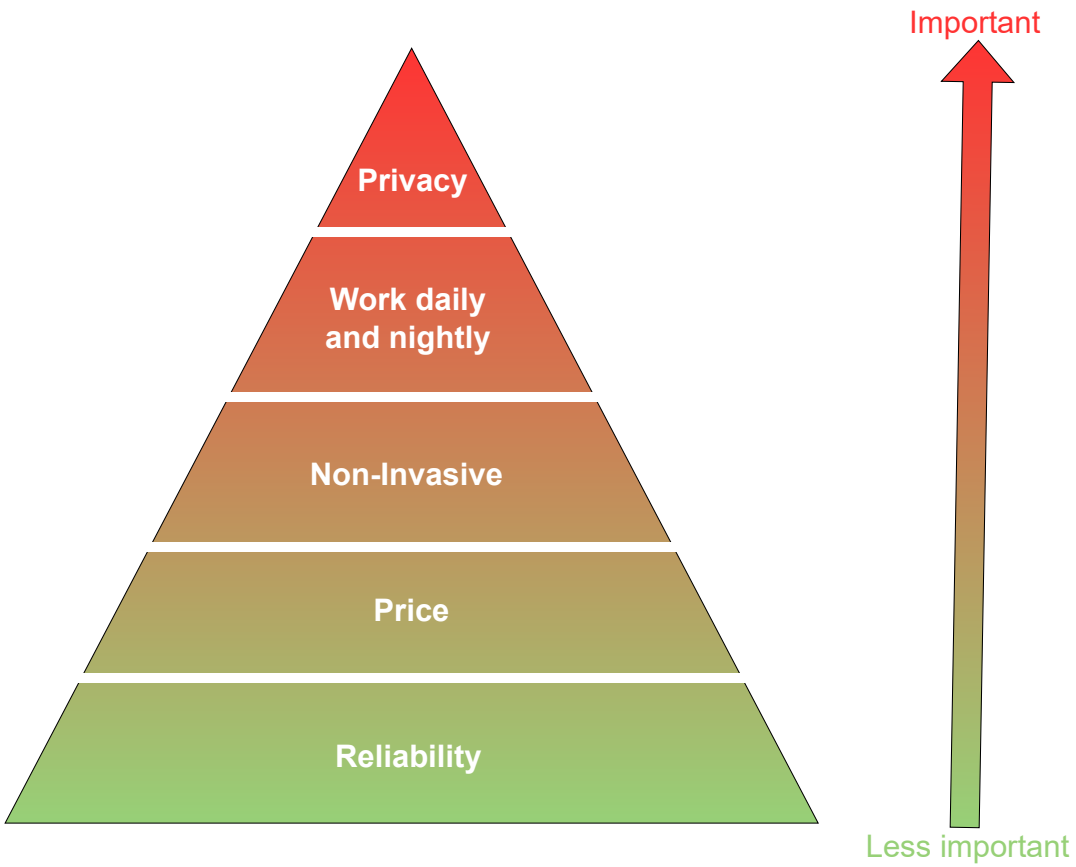



**Figure 1.4:** The way we choose the sensor

Each of the sensors we have described above has advantages and disadvantages (Fig 1.3).

Due to the rapid development of electronic technologies, wearable sensors are inexpensive and energy efficient. Most of time, wearable data can be transmit easily and the processing time is fast. In addition, they can monitor senior activities every where not only in the room. However, we have noticed that to be effective, these sensors must be worn systematically, day and night, and in places that are not always obvious to an elderly person. This is extremely restrictive, and some people forget or even refuse to wear them. For these reasons, we believe that this type of sensor does not guarantee sufficient surveillance for the detection of falls.

Some of the ambient sensors, such as carpets, require expensive materials and significant room modifications. Others, such as WiFi, are technologies widely used in the world and already equip the rooms. On the other hand, they require setting up a model that has to be updated when there are any modifications of the furniture places in the room. Some of the ambient sensors protect privacy (carpet, WiFi), but others are very intrusive. This is the case with sound-based technologies that have

access to many types of private conversations. Implementing these solutions is quite complicated; current solutions based on ambient sensors are very sensitive with a high rate of false positives. That's why we didn't take into account the ambient sensors.

Regarding wearable sensors, if they are very efficient, they are invasive. They can not be used to the fall detection of seniors people who can forget to wear them. So there remain vision-based sensors. Their reputation is that these sensors are intrusive. Even if it has been proven that the more a person is accustomed to being surrounded by technological devices, the more he will be inclined to accept the installation of surveillance tools [159], we will review the acceptability of vision sensors for surveillance and fall detection.

In [63], Le Dain and Cormier conduct a study on seniors (and seniors relatives) and retirement home staff to assess the acceptance of a fall detection system based on Kinect sensors. They use the model Unified Theory of Acceptance and Use of Technology (UTAUT) [160]. Evaluating the extent to which new technology will or will not be accepted. In the reported interviews, they find that seniors' needs are sometimes very different from those of accommodation managers. For example, managers want a system that will be proposed to senior but not imposed. Moreover, seniors do not want their bathroom to be in the system field of view. As for health and retirement home staff, they emphasized the system's usefulness but did not highlight the privacy issue. They want to keep the possibility to deactivate the system. Moreover, they want the system to respect their privacy and intimacy, and the system should not be installed without their agreement. In [161], Arning and Ziefle compare different vision sensors depending on medical safety, privacy, type of camera, and camera location. They gave a questionnaire to 194 participants and analyzed the data set using Conjoint Analysis [162]. They find that privacy is an essential issue for participants, and many of them reject face recognition. So, user acceptance of sensors surveillance is not only linked to their effectiveness.

Privacy is the first concern in the surveillance of seniors. As pointed out above, this is the main disadvantage of visible cameras. However, this is not only the case for cameras. There are also some privacy issues for depth, thermal, wearable devices, or ambient sensors. Whatever the type of data set, user data should be preserved or anonymized. The fact that we are highlighting more privacy issues for visible sensors is related to how raw visible images show more information than others. This is why a lot of work has been done to dis-identify people on visible images. For example, in [163], Erdélyi et al. propose a method to de-identify people using blurring, Sobel edge detection, and Mean-Shift Clustering to cartoon visible images. Other authors show that it is possible to get a good trade-off between privacy and efficiency [164, 165].

While visible images must be pre-processed to preserve a certain privacy level,

other types of cameras are more privacy-friendly. For example, thermal and depth sensors are more likely to prevent an individual from being identified in a video.

As we have already stated, the most critical aspect for us is privacy. Indeed, even though people nowadays easily share their data on social networks, they are less inclined to share their habits and what they are doing at home. Given privacy issues, we cannot take into account visible cameras. Besides, they cannot work nightly. This is our second criterion for choosing a sensor because most falls occur at night. The privacy issue also forbids us to build our system on sound.

Our third criterion is that a sensor should be non-invasive. This is not the case with ambient sensors because people do not need to wear them even on nights. Reliability also forbids us to ambient sensors such as WiFi or Radar because they are hard to setup. It remains Thermal and Depth sensors. Some manufacturers recently proposed some very low price thermal cameras, e.g., the FLIR Lepton 2 [2], with very low resolution ($80 \times 60$ pixels). Using two thermal sensors, we think that we could recover the depth information (that depth sensors provide) in addition to the temperature information (that depth sensors do not provide). Therefore, our choice is to develop solutions for fall detection and monitor seniors' activity using a stereo pair of thermal cameras. In the rest of this chapter, we will explain the characteristics of a thermal camera.

## 1.8 What is a thermal camera?

Thermal cameras are cameras that are sensitive to infrared radiation. This electromagnetic radiation (Fig 1.5) is in the 700 nm-1mm bands, between radio waves and visible light. Although it is part of electromagnetic radiation, infrared radiation was unknown before 1800. Sir William Herschel (1738 - 1822) discovered them by searching through an optical filter to reduce the brightness of the image of the Sun during solar observations. By the way, Sir William Herschel also discovered the planet Saturn.

Infrared is associated with heat because, at ordinary room temperature, objects spontaneously emit thermal radiation in the infrared range. Figure 1.5 shows spectral photon-emission as a function of temperature and wavelength.

The infrared radiations are usually divided into three categories:

— Short wavelength IR (SWIR or near IR): going from 0.35 $\mu m$ to 2.5 $\mu m$. The SWIR sensors work well for the high atmosphere but give poor performances for human environments. They are usually used for night vision and IR spectroscopy.

2. https://www.flir.com/globalassets/imported-assets/document/lepton-2.5-family-datasheet.pdf

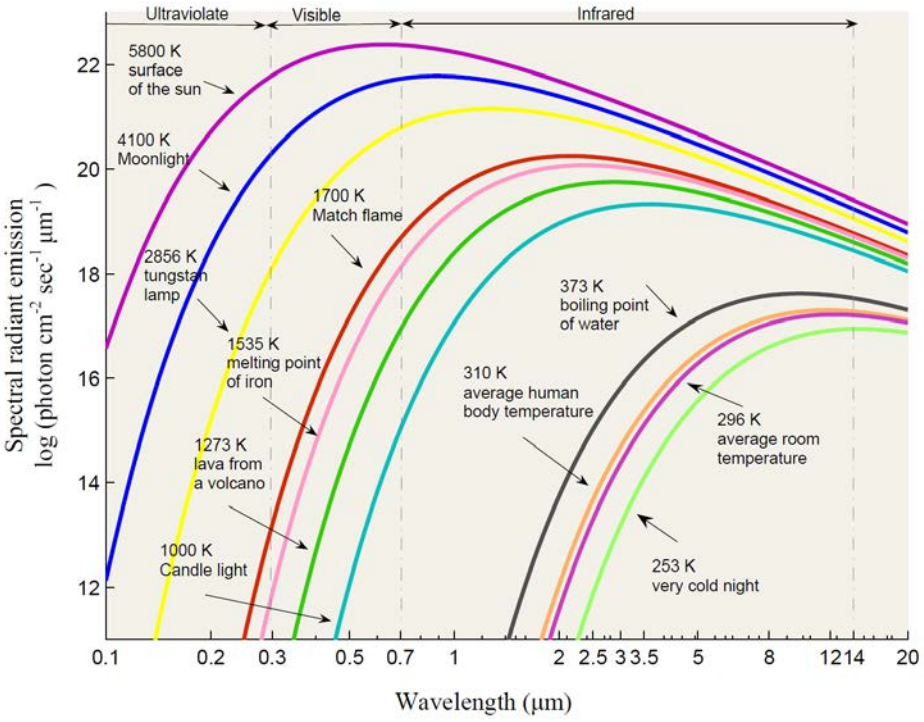



**Figure 1.5:** Example of infrared radiation reflection (from [166])

— Medium wavelength IR (MWIR or Medium IR): going from 3.3 $\mu m$ to 5 $\mu m$. The MWIR sensors present the advantage of providing less noisy images compared to LWIR sensors.

— Low wavelength IR (LWIR or far IR): going from 8 $\mu m$ to 14 $\mu m$. Most of LWIR sensors are adapted to human environments. This band covers the thermal emissions of bodies. So, in this band, cameras can produce thermal images of a scene without requiring external illumination such as the Sun, moon, or infrared illumination.

In 1880, the bolometer was invented by Samuel Pierpont Langley to study solar radiation. The first infrared-sensitive sensors have been created by the Hungarian physicist Kálmán Tihany, in 1929, for the British anti-aircraft defense following World War I. Due to these early sensors' success, the US Army and Texas Instrument created the first infrared line scanner in 1947. Since then, infrared sensors have undergone many improvements. Nowadays, they are widely used for night vision, medical testing, roof inspection, security, hobby photography, etc.

Nowadays, the thermal sensors market is increasing more and more and will reach 4.6 billion $ by 2025 [167]. There are two types of thermal sensors [168]:

— Sensors detecting the change in heat induced by the absorption of IR radiation. They use two types of detectors: bolometric detector and pyroelectric detector.

— Quantum based sensors. They are specialized in measuring low temperatures ( $\approx 77$ K) and require cryogenic cooling.

While the second type of sensors is more sensitive than those from the first type, they are, on the other hand, more expensive, bulkier, and heavier than the first ones. Moreover, those from the first type do not need cryogenic cooling. As one of the criteria is the price, we chose to use a sensor of the first type for monitoring seniors. Besides, we need sensors that can be used in human environments, day and night, so we have chosen thermal imaging (far infrared).

Compared to other sensors, thermal sensors have many characteristics:

— High noise and low resolution: thermal cameras are characterized by a lower signal to noise ratio compared to visible cameras. When there is noise, it is difficult to apply certain classical computer vision algorithms, such as edge detection. Also, high-resolution thermal cameras are very expensive. So most of the thermal cameras have a low-resolution, which adds difficulties to image processing.

— IR reflection: Thermal cameras are also affected by IR reflection. Due to the longer wavelength of infrared radiation, this phenomenon can lead to low or miss-detection, and some objects may be detected twice. In [169], Goubet et al. address this issue in their proposed method for tracking pedestrians.

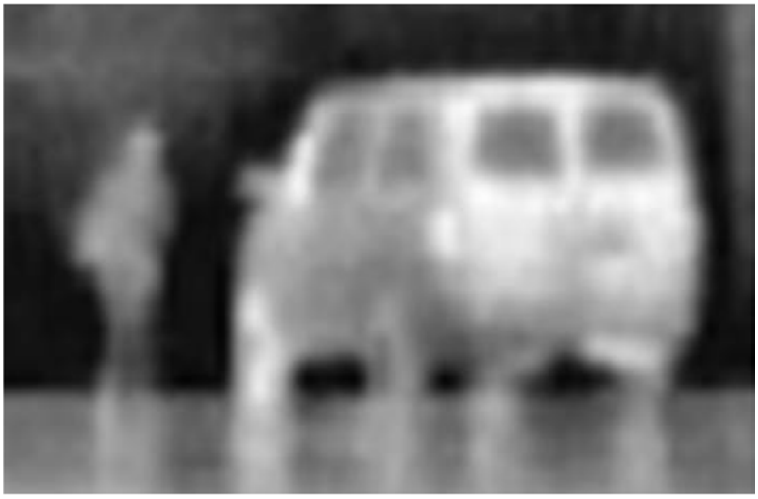


**Figure 1.6:** Example of infrared radiation reflection (from [169])

— IR halo effect: A halo effect also characterizes infrared image. This effect appears when a very cold or very hot object is contrasted with the background. Some computer vision techniques may not work properly with the halo effect. Indeed, the halo could be considered as part of the object. Although the halo can facilitate segmentation, other processing steps such as matching could give worse results.

— History effects: Temperature variations do not change quickly. The propagation does not take effect instantly, so in a video, the current frame may be affected by the previous frames. Some computer vision techniques, such as optical flow computation, will not work well. Indeed, the assumption about brightness constancy is no longer correct. When the temperature varies slowly, previous frames may leave a "ghost image" in the current frame.

It is important to note that by choosing high-quality thermal cameras, some of these constraints disappear.

## 1.9 Conclusion

Fall detection and activity monitoring are essential nowadays. In this chapter, we answer the questions, "*what?*" "*why?*" "*where?*" and "*how to prevent/detect?*". The consequences can be costly not only for the faller but also for the public finances. Fall can be linked to intrinsic or extrinsic causes. Fall detection or prevention consists of alleviating some causes. This detection or prevention can be performed through many strategies such as intervention in the environment, analysis of orthostatic balance and posture, activity analysis, or gait analysis.

Most of these solutions can be combined, but for our purpose, we choose activity analysis. To monitor senior activities, we choose thermal infrared sensors according to a priority list of properties such as anonymity, working daily and nightly, etc. Our device is composed of two thermal cameras set in stereo to take advantage of 3D triangulation.

Unfortunately, thermal cameras are characterized by many drawbacks, such as low resolution, noise, etc. Our work's main purpose is to detect falls and monitor senior activities despite these advantages and disadvantages. In the next chapters, we will dive in this way.

# 2 On how many roads must a man fall, before you call him a man?

"How many roads must a man
walk down
Before you call him a man?"

Bob Dylan

## 2.1 Introduction

In the previous chapter, we highlighted the need to detect falls and monitor senior's activity to detect frailty. These two aspects are the main concerns of the ANR project PruDENCE (ANR-16-CE19-0015-02).

PruDENCE is a collaboration between the NeoTecVision (Pacé), the LTSI team at the University of Rennes 1, ECAM (Bruz), a Living Lab at UTT (Troyes), and Université Lille II (Lille). PruDENCE is focused on creating a new low-cost device based on depth and/or thermal sensors to prevent the risk of falls by analyzing the activity of seniors. PruDENCE is divided into several work-packages:

1. First, the detection of activities from a depth sensor and a low-resolution thermal camera. This work is the following of a previous Ph.D. work [63];
2. Secondly, the detection of falls and the detection of activities from a pair of a stereo low-resolution thermal camera. This pair of stereo low-resolution thermal camera should be preferred to the pair of a depth sensor and a low-resolution thermal camera because of a lower cost;
3. Thirdly, the statistical analysis of the activities to predict senior frailty;
4. Then, the validation of the solutions in the Living Lab;
5. Then, the implementation as an industrial product that could be commercialized;
6. Finally, the study of the regulatory aspect, mainly from a data protection point of view.

The choice of the sensors (depth and thermal cameras) was directly linked to the analysis made in Chapter 1. Both sensors responded to the different characteristics of acceptability to monitor people: they are purely passive. They work during day and night and ensure respect for the anonymity of the observed people.

Our work is directly linked to the second work-package: implementing image processing solutions to detect falls and monitoring the activity from a pair of thermal cameras in stereo mode. The main difficulty comes from the fact that the sensors chosen on a cost criterion are very low resolution ($80 \times 60$ pixels) and that the images themselves are poor in information. The use of a camera pair was also justified because the information of one camera is purely two-dimensional and that fall detection and/or activity tracking require a precise estimation of the person's pose in the room.

This chapter will first present the global pipeline and the stereo setup. Then it will present the framework we developed for the detection of falls and activity monitoring from the pair of low-resolution thermal cameras. Throughout the implementation of image processing solutions, we were confronted with different problems that led us to change our strategy many times to solve the final problem. This chapter will also present and justify the several bypass strategies that led us to new contributions.

## 2.2 Global pipeline

To monitor seniors activity through fall detection, we tried to define three steps:

— First, it could be important to exploit the two views to get a 3D pose by simple information extraction on the images (location of the person, estimation of its 2D center of gravity or bounding box) followed by a 3D stereoscopic reconstruction of this information. The expected result is the detection or not of a fall.

— Second, the activity analysis will require more information, so the images' improvement must be envisaged. Due to the low-resolution, we chose super-resolution.

— Third, to perform the monitoring of the people activity, a finer analysis of the posture (standing, lying, walking, sitting) must be performed. The problem will be how to integrate complementary information to the low-resolution pair of stereoscopic images.

First, we considered a 3D scene reconstruction solution using conventional stereoscopic vision adapted to our low-resolution thermal images (see Section 2.4.1 and Section 2.4.2). So at the beginning of the Ph.D., we developed solutions that gave better precision than traditional techniques to compensate for our images low-resolution. These solutions should work in a perfect world. Unfortunately, even these higher precision methods did not give the necessary accuracy for a 3D reconstruction to detect falls. We decided then to tackle directly, without stereo reconstruction, the fall detection from a pair of thermal images (see Section 2.4.3). We also did some prospective research to estimate an activity from a thermal video sequence (see Section 2.4.4).

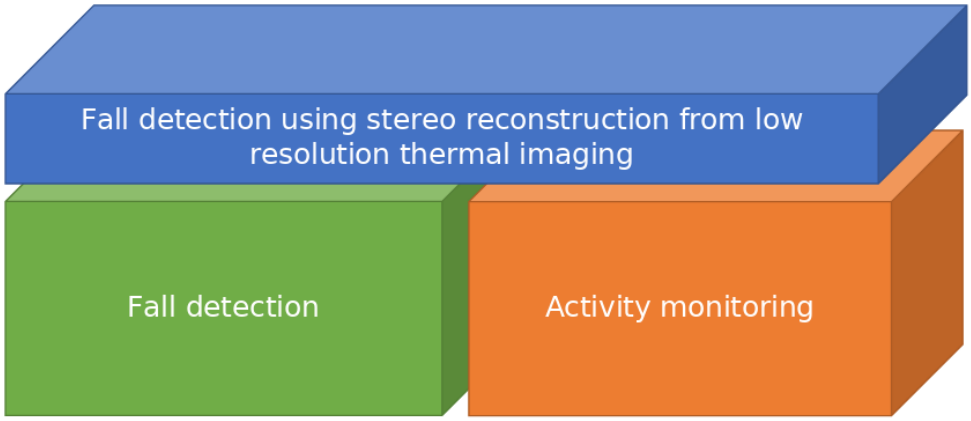



**Figure 2.1:** Overview of the thesis

The Fig 2.1 represents our global pipeline divided by two modules: fall detection and activity monitoring pipelines.

## 2.3 Thermal cameras and stereo setup

| | |
|---|---|
| Dimension | 8.5 x 11.7 x 5.6 mm |
| Resolution | 80 (h) x 60 (v) pixels |
| Pixel size | 17 µm |
| Field of view | 51° (h) x 37.83° (v) |
| Thermal sensitivity | <50 mK |
| Accuracy | ± 5°C |
| Frame rate | 9 Hz |
| Dynamic range | -10 to 140°C |
| Price | <200$ |

**Table 2.1:** Characteristics of Lepton 2.5

To monitor senior activities, we should first set a device to record these activities. These images should be later used in a dedicated algorithm to detect and classify activities. We chose two thermal cameras set in stereo. This choice is motivated by the fact that thermal cameras have been widely used in surveillance context [170–174]. Thermal cameras with a good resolution are relatively expensive [2]. The price is a high criterion for a hypothetical large-scale production of the final product. However, recently, some manufacturers proposed some very low price thermal cameras, e.g., the FLIR Lepton 2. The characteristic of this camera is shown in Table 2.1.

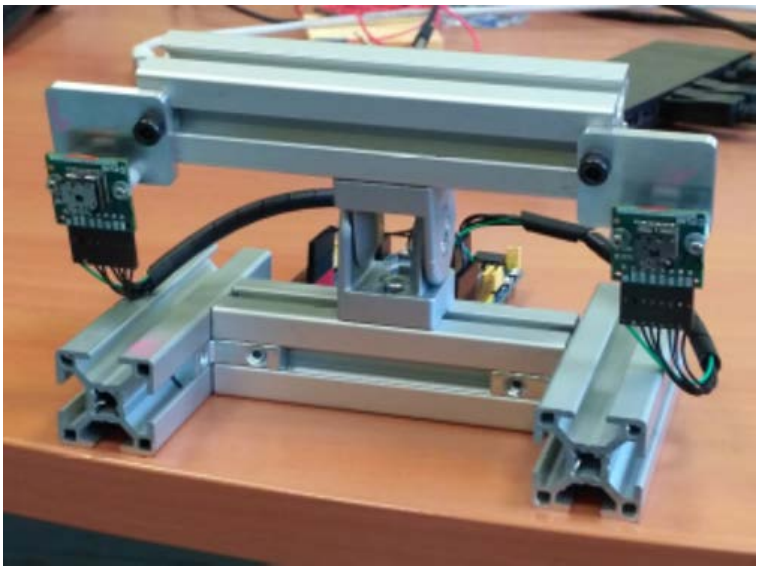


**Figure 2.2:** The stereo system composed of two lepton 2 cameras placed.

The downside is that low-cost cameras have a very low-resolution ($80\times60$ pixels), produce noisy images with value drift over time (the cameras correct this drift from time to time, which has the effect of a sudden temporal jump in image values) and do not have a robust temperature calibration.

The acquisition system is composed of a pair of FLIR lepton two cameras (see Fig 2.2). They are placed in parallel (their optical axes are parallel) to enhance the field of view. The distance between the two cameras (the baseline) has been set to 16 $cm$ to include the two cameras in a not too bulky housing. The device will be placed high up (on the ceiling or a wall just below the ceiling) and directed to monitor an entire room.

Both cameras are controlled using a micro-controller card adapted and programmed by the company NeoTec-Vision. This card allows us to retrieve the images on a PC via USB and, in the future, will interface with an embedded processing board.

## 2.4 Contributions

To perform 3D vision using a stereo system, it is essential to know the cameras characteristics to triangulate. The chapter 3 tackles the problem of determination of these characteristics under our conditions.

### 2.4.1 Stereo-vision and reconstruction

To perform 3D vision with a stereo system, it is important to know the cameras' characteristics to triangulate. The chapter 3 tackles the problem of determination of these characteristics under our conditions. There are two types of characteristics or

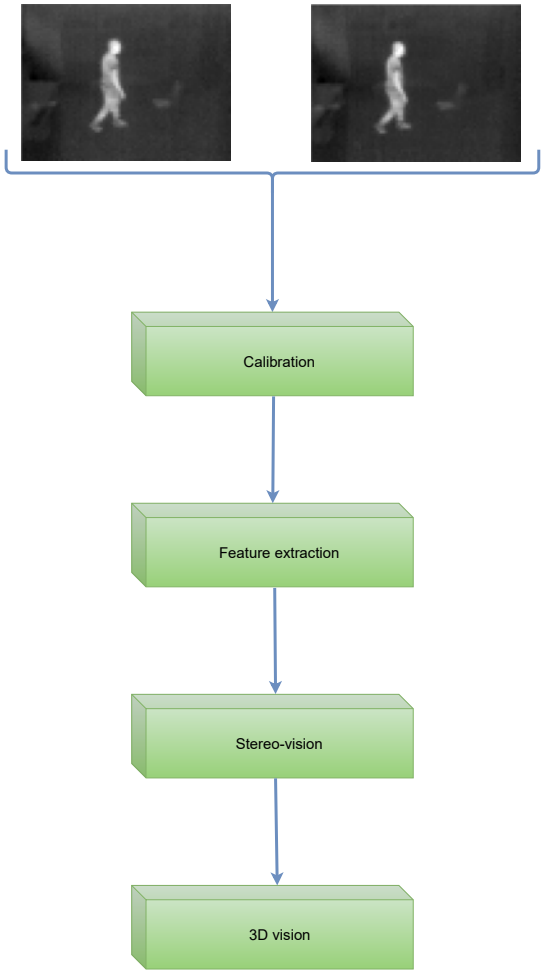



**Figure 2.3:** Our original plan

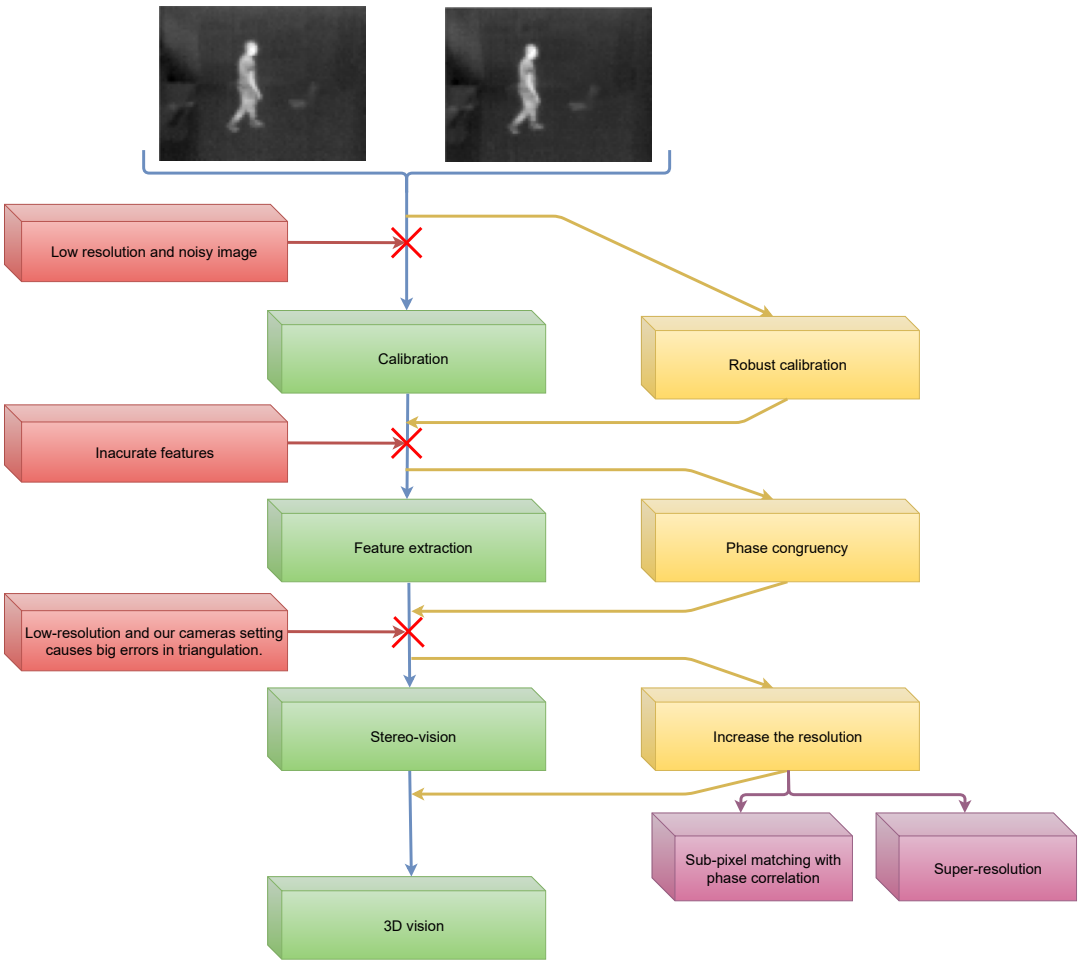



**Figure 2.4:** Our new plan

parameters: intrinsic parameters and extrinsic parameters. Intrinsic parameters are linked to the camera itself, while the extrinsic parameters are linked to the camera's position accordingly to the world. For this purpose, we manufactured a wooden calibration board with some hot bulbs placed to be detected by the thermal cameras. Then we show that it is possible to get accurate calibration results despite the low-resolution aspect of our images. The stereo calibration will give us some information such as the stereo baseline (the distance between the two cameras), the focal length, the fundamental matrix, the intrinsic and extrinsic matrices, etc. As we show in Section 3.3, in state of the art most of the time, the cameras that have been used have higher spatial pixel resolution. To our best knowledge, it was the first a stereo calibration that has been conducted in such a way using very low-resolution thermal cameras and a wooden chessboard. We wanted to bypass our cameras' aspect by performing robust calibration, as shown in Fig 2.4.

Once cameras are robustly calibrated, we might be able to perform stereo vision. But we encounter other problems through the small baseline of our stereo-system and other aspects of our cameras. Indeed, the horizontal field of view is $51°$, and the diagonal field of view is $63.5°$. The maximum frame rate is eight frames per second. The baseline is set to 16 mm. As input images, we had a pair of low-resolution thermal images acquired in stereo condition $I_l$ and $I_r$. The images are rectified, using the output of a robust calibration method [175]. The rectification step simplifies the stereo reconstruction step since that a feature lies now on the same line in $I_l$ and $I_r$. The feature matching (and the estimation of the disparity) is simplified to estimate the translation $\delta_x$ of the feature between the two images along the x-direction. We have: $\delta_x = d_i(x) + d_d(x)$ where $d_i(x)$ and $d_d(x)$ are respectively the integer and the decimal parts of the disparity. Some works have already been performed on thermal stereo-vision based on high-resolution thermal cameras [176, 177], but one of the obstacles to the democratization of thermal cameras is their cost [2]. These cameras' counterpart is their low spatial resolution ($80 \times 60$ pixels for the Lepton 2). Such resolution directly impacts several steps of the traditional stereo-vision framework: stereo calibration, information extraction, information matching between the two views, and triangulation. Matching is a prior step of 3D vision using a stereo system. The matching can be dense or sparse. Dense matching works well for textured images; unfortunately, thermal images are lacking texture. Thus, sparse matching on features extracted from the images must be considered. The final global framework is composed of three main steps (Fig. 4.7): 1) robust features extractions method from the low-resolution thermal images using phase congruency, 2) first rough stereo matching of these features in integer precision (the estimation of $d_i(x)$) and 3) refined sub-pixel matching around the previously matched features (the estimation of $d_d(x)$).

We showed that most of the features extraction method does not work well for thermal images. Then, we presented a method based on log Gabor called phase

congruency. The main advantage of this method is that it is robust to illumination change. Moreover, we compared phase congruency versus traditional features extraction methods in therm of the number of extracted features and robustness to illumination change. We demonstrated that phase congruency is very competitive. While one of the drawbacks of phase congruency is high computation time, we reduced this computation time by implementing it on top of the C++ library Eigen [178] with SIMD virtualization.

Then to deal with our image characteristics, we propose a sub-pixel matching method using phase correlation. Fig 2.4 shows our main strategy with sub-pixel matching. Instead, we apply it to one of the phase congruency outputs. Such an output is less noisy. Using the robust calibration method we defined in 3, we show that when performing stereo matching, an error of 1 pixel can lead to an error of 50 cm in depth computation through triangulation. This is why, we proposed a super-resolution method adapted to thermal images.

### 2.4.2 Super-resolution

One solution to address this induced low accuracy is to increase the size of images, so adding more information. However, such a process, so-called super-resolution, is an ill-posed inverse problem. Indeed, an infinite number of high-resolution images can correspond to the same low-resolution image. When only one image is used during this process, it is called a single image super-resolution (SISR). As sub-pixel matching, super-resolution can also be used to improve the accuracy of the stereo-vision, as shown in Fig 2.4.

Most of the time in state of the art, the authors used bicubic degradation. But in real-world applications, the degradation is more complicated. Moreover, thermal cameras point spread function can be different from visible cameras point spread function. In [179], the authors proposed a network handling multiple models of degradation for visible images. They state that the blind model cannot work well in real applications. The input of their network is the low-resolution image and a degradation map. For known blur kernel and noise, the degradation map is estimated through a dimensionality stretching. Given that real images do not have ground truth, the authors performed a grid search to estimate the degradation settings with good visual quality. Such a scheme will be quite challenging for real-time applications. Moreover, it is difficult for the thermal images to assess the visual quality of the reconstructed images. However, this assumption is not always valid for actual imaging device sensors. Therefore, once generating synthetic low-resolution images for training, we must consider a wide range of noise and blurring artifacts as possible. Unlike previously published super-resolution methods, here we propose a blind model, the Edge Focused Thermal Super-resolution (EFTS), to perform single image super-

resolution for thermal images. Our model is based on residual dense block preceded by an edge extraction module, which focuses the reconstruction on the edge enhancement. Our contributions are threefold:

— First, we investigated to find the best combination of edge operators (Sobel, Kirsch, Laplace, Prewitt) to obtain better results

— Second, we proved that the edge extraction module helps our model to output a reconstructed image with more enhanced edges.

— Third, we showed that in the context of thermal images, very deep networks tend to over-fit given that thermal images contain less pixel variance than their visible counterparts.

### 2.4.3 Detection of fallen person

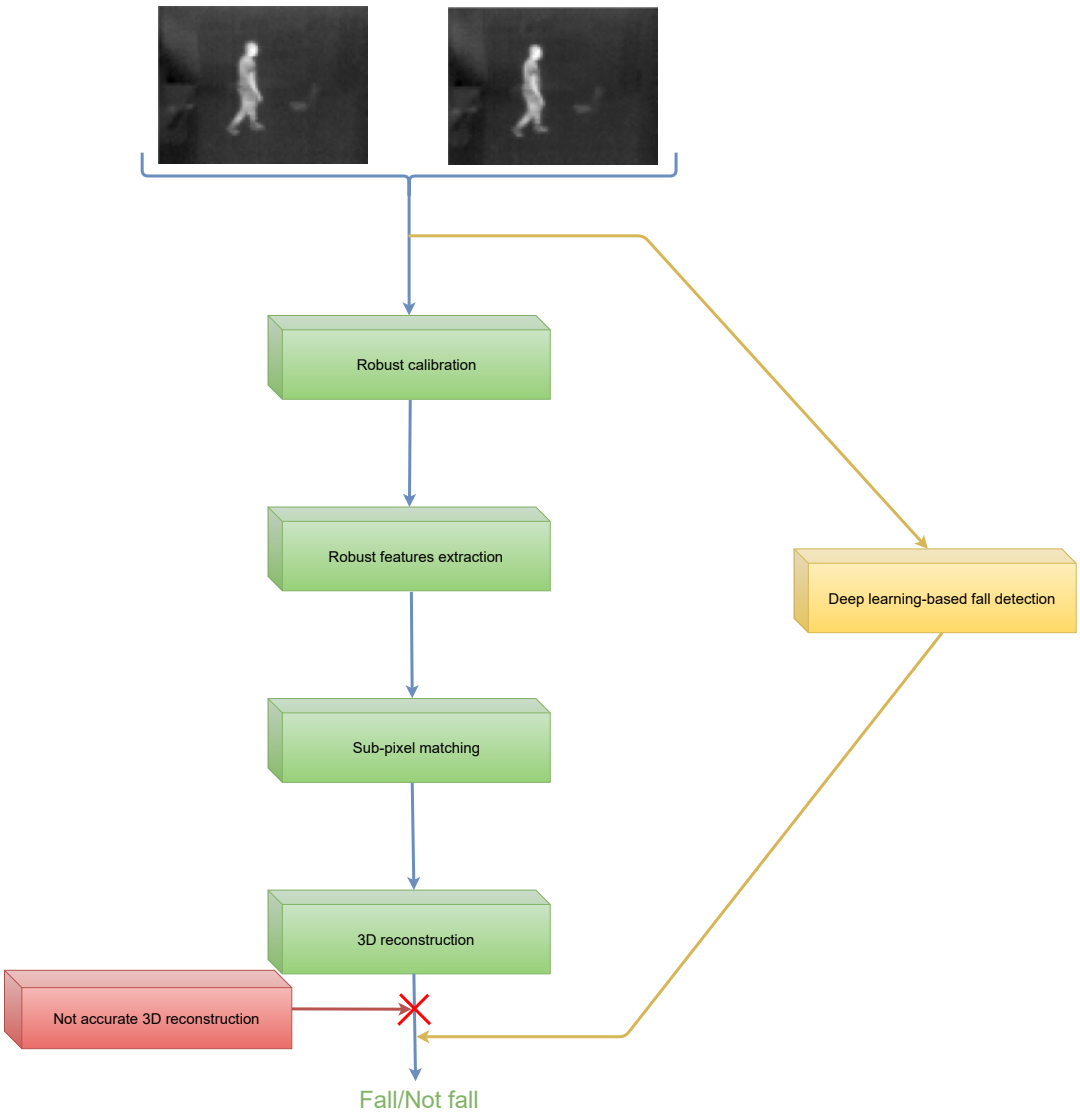



**Figure 2.5:** Fall detection plan

Once we were able to perform super-resolution and accurate stereo-vision, we try to detect falls. In a first attempt we tried to estimate a fall by measuring the distance between the center of gravity of the people and the ground both reconstructed by our stereo-vision setup. Unfortunately, despite the fact that we had developed a robust calibration method, and features extraction and matching at a sub-pixel level of accuracy, our 3D reconstruction framework did not allow us to reconstruct accurately the ground plane, nor to estimate the pose of the person from our low-resolution thermal image pairs.

This is why we proposed a machine learning solution based the detection of the ground plane (Fig 2.5). A fall is then considered if a certain percentage of a person is detected as been on the ground. We trained a machine learning solution (SVM or a deep learning network) to detect if a feature seen in the right and left image lies on the ground or not. For this we had the idea to move a bulb on the ground and not on the ground to train the machine learning solution. The detection or not of a fall is then performed as following: 1) detect robust features on the left and right uncalibrated images pair using phase congruency, 2) match these features in order to got feature stereo pairs, 3) classify each feature stereo pair as on the ground or not and 4) decide if a person is on the ground or not by a simple statistic on the classified feature stereo pairs.

### 2.4.4 Activity recognition

In the state of the art, most of the time, fall can be considered as an activity. In order to detect activities, we chose deep learning-based methods rather than handcrafted-methods because, recently, these types of methods out-perform in many fields. One of the consequences of this choice is that we needed more data. To deal with the lack of thermal datasets, we chose many mechanisms by:

— Train on visible edge and infer on thermal edges

— Train on synthetic thermal image then infer on real thermal images

— Train on visible optical flow then infer on thermal optical flow

— Train a domain generalization model on visible and visible edges, then infer on thermal edges.

The preliminary results show that using Sobel edges map and synthetic thermal images, we got better results. But these results should be improved in a certain margin.

## 2.5 List of publications

**Zoetgnandé Y.**, Cormier G., Fougères A.-J., Dillenseger J.-L., "Thermal Stereo Fall Detection: TSFD", Under review on Journal Computer Vision and Image Understanding

**Zoetgnandé Y.**, Cormier G., Fougères A.-J., Dillenseger J.-L., "Sub-pixel matching method for low resolution thermal stereo images", Infrared Physics and Technology, 105, 2020, pp. 103161, doi: 10.1016/j.infrared.2019.103161.

**Zoetgnandé Y.**, Dillenseger J.-L., Alirezaie J., "Edge focused super-resolution of thermal images", International Joint Conference on Neural Networks (IJCNN), Budapest, 2019, doi: 10.1109/IJCNN.2019.8852320.

**Zoetgnandé Y.**, Fougères A.-J., Cormier G., Dillenseger J.-L., "Robust low resolution thermal stereo camera calibration", 11th International Conference on Machine Vision (ICMV 2018), proc. SPIE 11041, Munich, 2018, pp. 11041-1D, doi: 10.1117/12.2523440.

**Zoetgnandé Y.**, Alirezaie J., Dillenseger J.-L., "Super-résolution d'images infra-rouge basse résolution pour la détection de chutes", RITS 2019, Tours, 2019.

Msaad S., Cormier G., Zoetgnande Y., Prud'homm J., Carrault G., "Frailty detection in older people by monitoring their daily routine: A simulation study", The 20th IEEE International Conference on BioInformatics And BioEngineering

Msaad S., Cormier G., Zoetgnande Y., Prud'homm J., Carrault G., "Détection de la fragilité chez les personnes agées par suivi des activités de routine de la vie", 40e Journées Annuelles de la Société Française de Gériatrie et Gérontologie

## 2.6 Conclusion

In this chapter, we highlighted our main contributions to fall detection and activity monitoring. During our studies, we faced many issues that oblige us to change our track. First, we wanted to calibrate cameras using standard methods. We found that this is not possible using classical calibration. Then, we first tried standard features extractors for stereo-matching and find that we could improve our results using phase congruency. We wanted to detect fall by fitting the ground plane, but we found that our cameras' constraint did not allow us to get accurate stereo vision. This is why we propose a super-resolution method and a fall detection method based on learning the ground. Finally, given that we did not have enough dataset, we investigated many ways to deal with activity monitoring without training the given model on the target dataset.

# 3 Stereo setup

> You limit yourself by reducing the number of viewpoints you see.
>
> Meir Ezra

## 3.1 Introduction

In the first chapter, we explain our choice to use two thermal cameras set in stereo. This chapter will introduce our stereo setup and explicit the method we propose to calibrate these low-resolution cameras. As stated in the previous chapter, our first idea was to use two cameras in stereo-vision and recover the 3D position by triangulation. But before being able to compute such a position, we must dive deeper into image formation. Indeed, the image we have getting in a camera is just *interpretation* of the world (i.e., a projection of the 3D world on a plane).

We will define try to define and exhibit this *interpretation*. One of the classical ways to perform this calibration is to image a geometrically well-defined calibration pattern (classically a chessboard or a grid.). For Monocular vision, the calibration concerns only one camera, but for stereo-vision, most of the time, the two cameras are associated during the process

In this chapter, we will propose a solution for the stereo calibration of low-resolution thermal cameras. For this, we had to solve several sub-problems. 1) Given that thermal cameras are only sensitive to heat, we had first to choose and manufacture a calibration grid adapted to our low-resolution context. 2) The cameras gave only low-resolution and extremely noisy images, so it is challenging to exploit these images in 3D vision, especially for the calibration featured detection in the images.

We choose to overcome this latter problem by performing calibration with a high number of image pairs. The accumulation of image pairs allowed to lower the impact of the noise and gain in precision for the estimation of the parameters. However, in our opinion, it was important to know the optimal number of image pairs to get a robust calibration. That consisted of finding the good trade-off between the ease to perform and the Robustness of the calibration.

This chapter is structured as follows: Section 3.2 recalls some background about vision. Section 3.3 presents some related work about stereo thermal cameras calibration and more specifically about calibration grids and the point detection process in thermal

images. Section 3.4 details our method. Section 3.5 presents some results and our main contributions. Finally, section 3.6 concludes the chapter.

## 3.2 Background

### 3.2.1 Monocular vision

Transforming a scene to an image is complex and can vary a lot. Indeed depending on the type of the optical camera, the acquisition process can change.

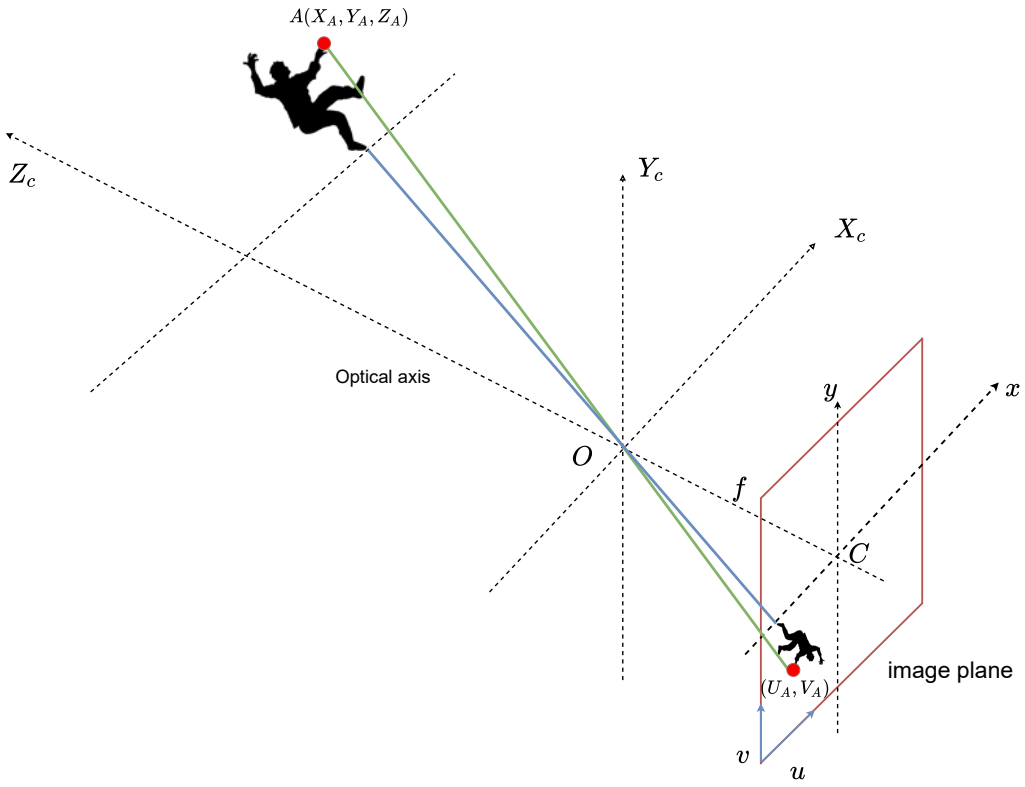



**Figure 3.1:** Geometry of the pinhole camera

One of the simplest camera model is the pinhole camera model. It has been modeled during the 13th century. It is formed by a closed box with a small opening on the front side through which light enters, forming an image on the opposing wall. From Fig 3.1, we have a falling person located at a horizontal distance $Z$. From the Fig 3.1, by projection onto the plane $Cyz$ (or onto the plane $Cxz$), we get the fig 3.2 (in this figure, we set the image plane between O and the scene). So, we have:

$$\begin{aligned} \tan(\beta) &= \frac{Y_A}{Z_A} = \frac{V_A}{f} \\ \tan(\beta) &= \frac{X_A}{Z_A} = \frac{U_A}{f} \end{aligned} \tag{3.1}$$

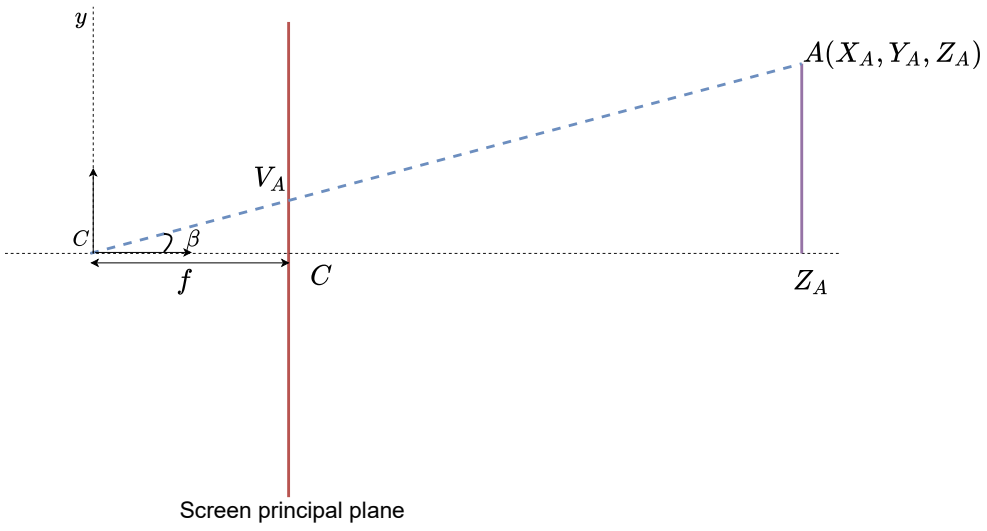



**Figure 3.2:** Projection onto plane $Cyz$

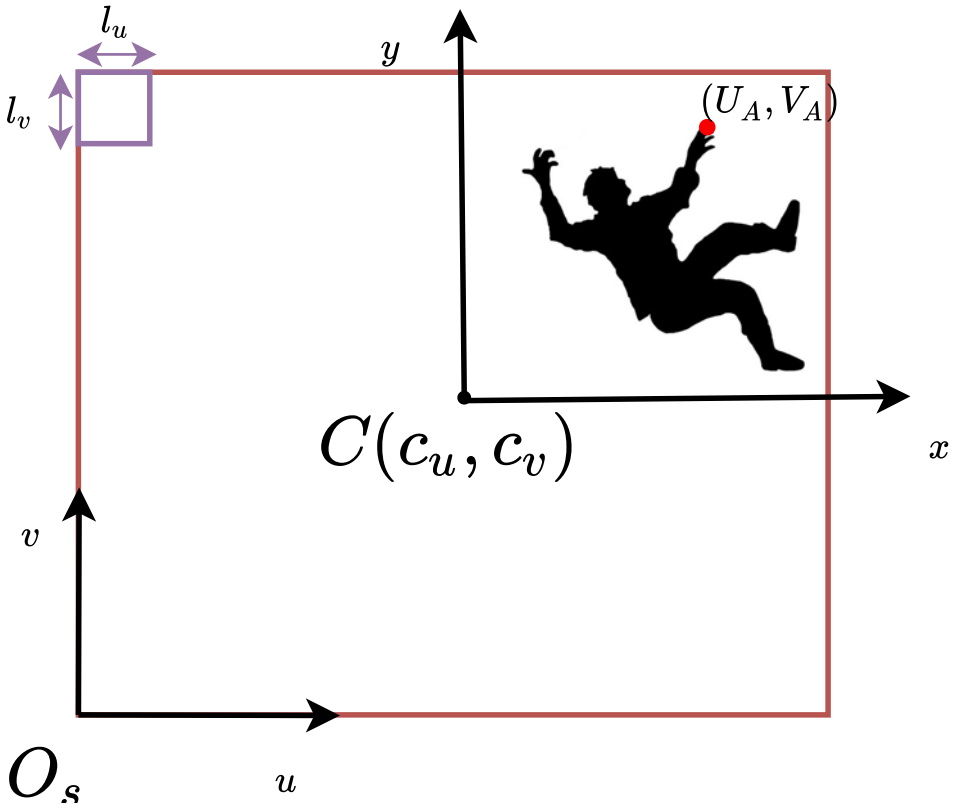



**Figure 3.3:** Perspective projection

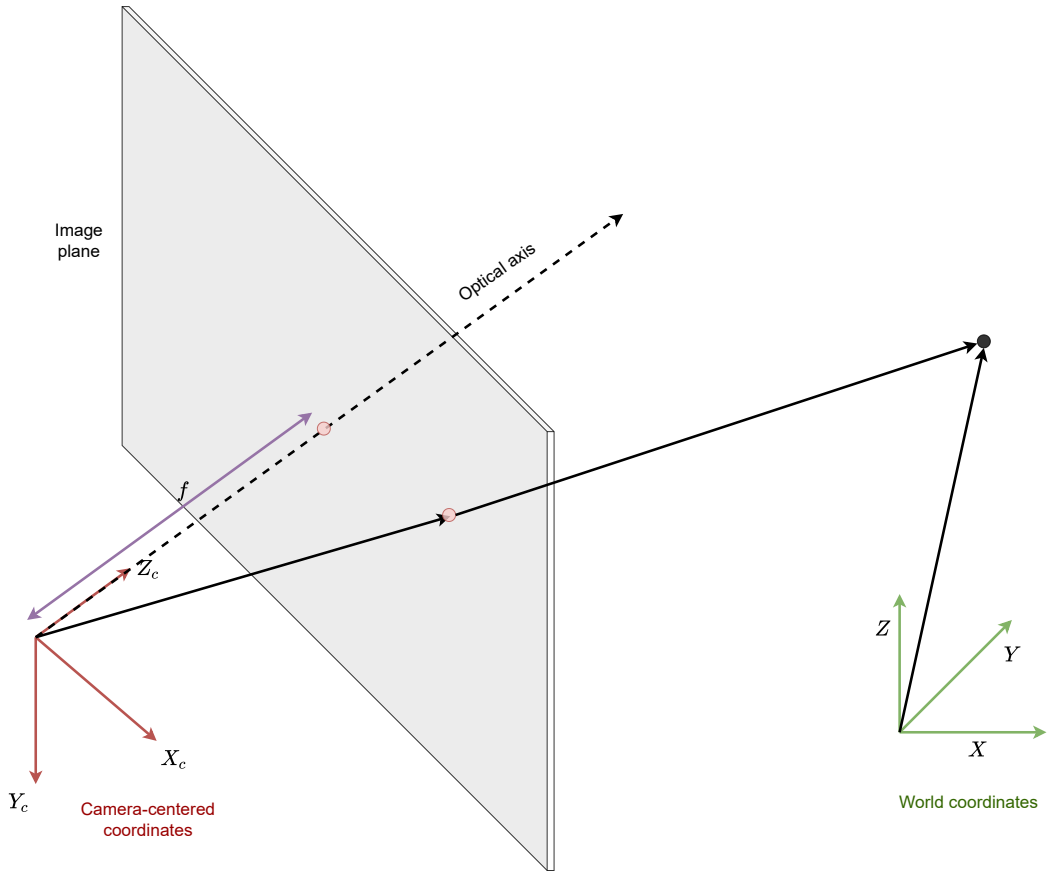



**Figure 3.4:** Camera model (World coordinates and cameras coordinates)

So:

$$U_A = f \times \frac{X_A}{Z_A} \quad \text{and} \quad V_A = f \times \frac{Y_A}{Z_A} \tag{3.2}$$

Equation 3.2 is called the perspective transformation, and it gives the position in mm of the projection of $A$ to the principal image plane. But the final image is a sampling of the principal image plane (Fig 3.3). A pixel has a size $l_u \times l_v$ (in mm2) and the optical center C is at location $(c_u, c_v)$. $c_u$ and $c_v$ are expressed in pixels. Usually, $C$ is close to the center of the image.

The coordinate $(u_A, v_A)$ on the image of the projected point $(U_A, V_A)$ is given by :

$$u_A = c_u + \frac{U_A}{l_u} \quad and \quad v = c_v + \frac{V_A}{l_v} \tag{3.3}$$

where $l_u$ and $l_v$ are the pixel size in $mm$. From Equation 3.3, we have:

$$s \times \begin{bmatrix} u_A \\ v_A \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{f}{l_u} & 0 & c_u \\ 0 & \frac{f}{l_v} & c_v \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_A \\ Y_A \\ Z_A \end{bmatrix} \tag{3.4}$$

Where $s$ is a scaling factor and $K = \begin{bmatrix} \frac{f}{l_u} & 0 & c_u \\ 0 & \frac{f}{l_v} & c_v \\ 0 & 0 & 1 \end{bmatrix}$ is the camera matrix. Besides extrinsic parameters each camera has some parameters called intrinsic. These ones are among others the focal length $f$, the pixel size $l_u$ and $l_v$ and the optical center $c_u$ and $c_u$.

Considering the Fig 3.4, the position of the camera i.e the position of $C, X_c, Y_c, Z_c$ (Cameras coordinates) accordingly to $O_w, X_w, Y_w, Z_w$ (World coordinates) is characterized by a rotation **R** and a translation **T**. These parameters represent the extrinsic parameters of a given a camera. It is worthy to notice that the world coordinates are not fixed and depend on each application.

The relation between $(X_w, Y_w, Z_w)$ and $(X_c, Y_c, Z_c)$ may be expressed as:

$$
\begin{aligned}
(X_c, Y_c, Z_c)^T &= \mathbf{R} \cdot \left[(X_w, Y_w, Z_w)^T + \mathbf{T}\right] \\
(X_c, Y_c, Z_c)^T &= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \cdot \left[(X_w, Y_w, Z_w)^T + \begin{bmatrix} t_1 & t_2 & t_3 \end{bmatrix}\right]
\end{aligned}
\tag{3.5}
$$

and considering homogeneous coordinates we have:

$$
\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}
\tag{3.6}
$$

The Equation 3.4 becomes:

$$
s \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{M} \cdot \begin{bmatrix} X_{w0} \\ Y_{w0} \\ Z_{w0} \\ 1 \end{bmatrix}
\tag{3.7}
$$

where $(X_{w0}, Y_{w0}, Z_{wo})$ is $(X_0, Y_0, Z_0)$ in world system and:

$$
\mathbf{M} = \begin{bmatrix} \frac{f}{l_u} & 0 & c_u \\ 0 & \frac{f}{l_v} & c_v \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\tag{3.8}
$$

And we have:

$$s \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \cdot \begin{bmatrix} X_{w0} \\ Y_{w0} \\ Z_{w0} \\ 1 \end{bmatrix} \tag{3.9}$$

with **M** the perspective transformation matrix. By developing we have:

$$\begin{cases} s \times u = m_{11} \times X_{w0} + m_{12} \times Y_{w0} + m_{13} \times Z_{w0} + m_{14} \\ s \times v = m_{21} \times X_{w0} + m_{22} \times Y_{w0} + m_{23} \times Z_{w0} + m_{24} \\ s = m_{31} \times X_{w0} + m_{32} \times Y_{w0} + m_{33} \times Z_{w0} + m_{34} \end{cases} \tag{3.10}$$

The purpose of the calibration is to estimate the 11 parameters (six intrinsic and five extrinsic) of the model or 12 parameters of matrix M (both are linked together).

For this, we need some known correspondence between 3D points and 2D image points). This correspondence is given by imaging some calibration device that contains some visible features points with the known 3D position.

In perfect condition, using six 3D points, we could determine the missing values. Unfortunately, most of the time, the images are noisy, and it is complicated to localize the $u$ and $v$ accurately. This is why people use more than six 3D points during calibration and estimate the parameters using least-squares methods.

These equations lie on the fact that we used the simple pinhole camera. But in the real case, cameras has lens which can lead to image radial and tangential distortion (Fig 3.5). Radial distortion is the dominant distortion and it is modeled as follows:

$$\begin{pmatrix} u_d \\ v_d \end{pmatrix} = L(\tilde{r}) \cdot \begin{pmatrix} \tilde{u} \\ \tilde{v} \end{pmatrix} \tag{3.11}$$

where:

— $(\tilde{u}, \tilde{v})$ is the position of the given point without distortion

— $(u_d, v_d)$ is the position of the given point after radial distortion

— $\tilde{r}$ the radial distance $\sqrt{\tilde{u}^2 + \tilde{v}^2}$ from the center for radial distortion

— $L$ is a function depending only on $\tilde{r}$

In pixels, we have:

$$\begin{cases} \hat{u} = u_c + L(r) * (u - u_c) \\ \hat{v} = v_c + L(r) * (u - u_c) \end{cases} \tag{3.12}$$

where $(u, v)$ are the measured coordinates, $(\hat{u}, \hat{v})$ the corrected coordinates and $(u_c, v_c)$ the center of radial distortion. By applying Taylor expansion we have:

$$L(r) = 1 + k_1 \times r + k_2 \times r^2 + k_3 \times r^3 + \dots \tag{3.13}$$

So in order to undistorted the image, it is necessary to access the values $(k_1, k_2, k_3, \dots)$. Tangential distortions can be taken into account and associated with radial distortion using the Brown–Conrady model [180]. The distortion parameters can also be integrated into the image formation model and estimated using a least-square procedure [181].

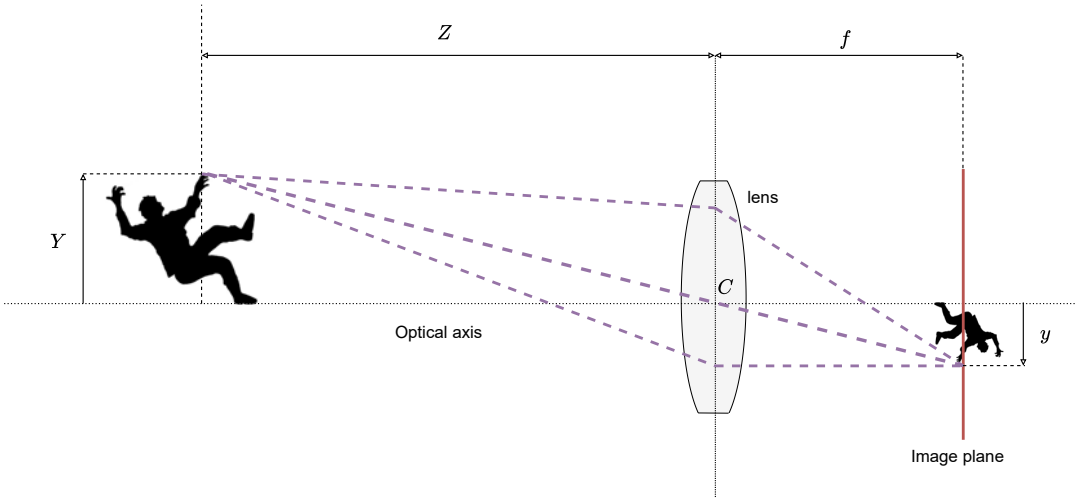



**Figure 3.5:** Lens projection models

### 3.2.2 Stereo vision

A 3D point P expressed in the first camera coordinate system $(O, XC, YC, ZC)$ can be expressed in the second camera coordinate system $(C', X'C, Y'C, Z'C)$ by:

$$P_r = R \times T \tag{3.14}$$

It has been demonstrated that this equation can be rewritten as:

$$P_r \times (T \times RP_l) = 0 \tag{3.15}$$

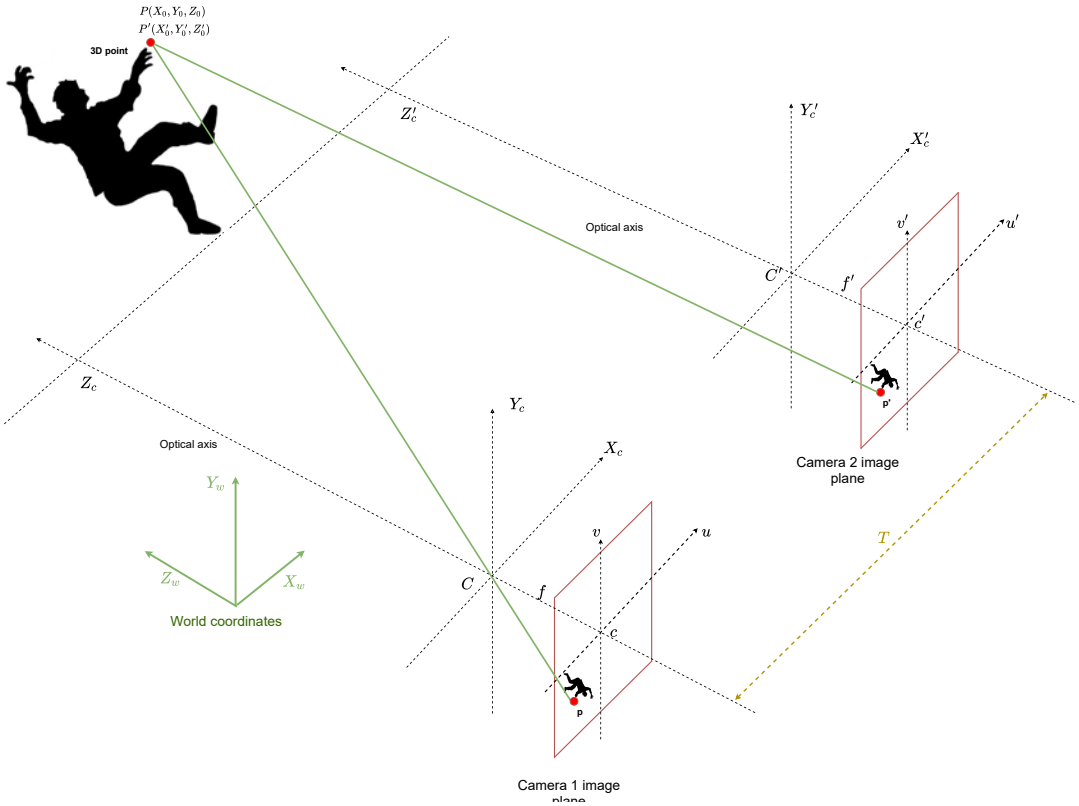



**Figure 3.6:** Stereo system composed of 2 cameras set in parallel

with $\times$ the cross product.

The cross product can be written as a matrix product by:

$$P_l \times ([T] \times RP_r) = 0 \tag{3.16}$$

with [T]x is the cross product matrix build from the vector T.

In summary, we will define a matrix $E$:

$$E = [T] \times R \tag{3.17}$$

which gives the relation between $P_l$ and $P_r$ by:

$$P_l TEP_r = 0 \tag{3.18}$$

E is called the Essential matrix and encodes the information about the extrinsic parameters between the two cameras.

In some cases, the cameras can be parallel (the optical axis $Z_C$ and $Z'_C$ are

parallel). $E$ depends then only on $T$, which is called the baseline (the distance between both cameras).

The point P can be viewed by the left and the right cameras as two points respectively $P_l$ and $P_r$ (see fig 3.6). If we consider that the cameras are described by a pinhole model and that $P_l$ and $P_r$ are expressed in homogeneous coordinate, the relationship between $P_l$ and $P$ can be written as $P_l = K_l P$ (see Eq 3.5) and the relationship between pr and P' can be written as $P_{=} KrP$. So equation 3.18 can be written as:

$$P_r T K_r T\,[T] \times R K_l = 0 \tag{3.19}$$

where F = KrT [T]xR Kl-1 is the Fundamental matrix. It encodes the information of the intrinsic parameters of both camera and the extrinsic parameters.

When we calibrate the cameras, the idea is to use corresponding points (provided by a calibration device viewed by both cameras) to determine F and E and so the extrinsic and intrinsic parameters of the stereo camera model.

When the two cameras are uncalibrated, the matching between 8 points allows getting F and so the other parameters. However, when the two cameras are calibrated (we already have the values of Kl and Kr ), the matching between 2 points is sufficient to determine the essential matrix E. But as in the one camera-case, we have to use more than this minimum number of points to ensure Robustness to noise and low accuracy of points detection. The parameters are then estimated in the least squared procedure.

## 3.3 Related works

### 3.3.1 Calibration grid

They are many works about calibration in stereo vision [182]. Most of these works concern visible images. A few works have been done for thermal stereo calibration. The main difference is that the points we want to detect must be hot. Generally, there are two ways to make the calibration grid visible by thermal cameras: passive and active heating.

Passive heating consists of heating some black/white calibration grid. Because of the color, the black and white parts will not get the same temperature. This allows getting a gradient of temperature that is sufficient to distinguish features [183, 184]. Once the grid is heated, it can be used for imaging. Another solution is to use sunlight to heat an asymmetric calibration pattern painted on a Dibond board[185]. In the best cases, they were able to get an RMS re-projection error of 0.348 on a $640 \times 512$ pixels

camera. Shibata et al.[186] use the same calibration grid to make a joint geometric calibration of visible and far-infrared cameras. Each of these cameras is prior calibrated separately, and then they estimate the extrinsic parameters. They get good results compared to state of the art. The resolution of their visible and thermal images was respectively $640\times480$ and $160\times128$. St Laurent et al.[187] try to find the best calibration pattern to calibrate stereo thermal infrared cameras. They present a passive calibration grid that can produce optimal image contrast to get a robust calibration. They are able to get an RMS re-projection error of around $0.1$. They use thermal cameras with a resolution of $640 \times 480$ pixels.

Active heating consists of using a self-heating grid. The self-heating features can be small light bulbs[188–190] or resistors[191]. Several grids have been proposed: 81 light bulbs arranged in a $9\times9$ matrix[189], black/white chess with 25 bulbs arranged in a $5 \times 5$ matrix to be used to calibrate simultaneously thermal and visible cameras[190].

It is important to notice the fact that the lowest resolution used in these works is $160 \times 128$[186]. But they used a hybrid stereo calibration with a good resolution visible camera $(640 \times 128)$ and a low-resolution thermal camera. None of these works doesn't concern low-resolution cameras such as ours.

### 3.3.2 Point detection

The problem is twice, first the features must be segmented, and then they must be located on the images. Generally, a simple threshold is sufficient to segment the regions representing features (e.g., the bulbs). However, depending on the resolution of the thermal camera, it is more or less important to locate the features in sub-pixel precision.

Yang et al.[190] propose a semi-automatic calibration method. They select first the four extremities of the matrix containing the bulbs. Then the features are estimated in sub-pixel accuracy using a quadratic approximation. The resolution of their thermal and visible cameras were respectively $240 \times 320$ and $768 \times 576$.

Ellmauthaler et al. [189] propose an algorithm to get a robust calibration. The algorithm has three steps. First, for each calibration image, each point in the grid is estimated in sub-pixel accuracy by computing the center of mass of each point after grey-scale thresholding. In the second step, they estimate the camera parameters. They then used the estimated homography matrix $H$ to refine the center of mass. This process is iteratively performed until convergence. It is important to note that this calibration, done in monocular vision, can be applied to stereo vision. But, the calibration is performed on a camera with a resolution of $320 \times 244$.

## 3.4 Our method

Our cameras are more challenging, given their resolution and noise. So, it is necessary to adopt a different way to calibrate and evaluate the results. As expressed in the introduction, we will use multiple pairs of images to reduce the impact of noise and low resolution. The calibration grid must be relatively easy to handle, and moreover, the thermal property of the features must be constant over time.

### 3.4.1 Calibration grid

First, we discarded passive calibration grids even if they are easier to handle, but the external thermal condition influences the quality of the image[187]. Given that we wanted to collect many images, the grid can get cool more and less quickly, changing the measured information in the images over time.

For the design of our active grid, we had to consider the low resolution and the reduced field of view of our cameras. We have manufactured a calibration grid (Fig. 3.7), which is composed of automobile bulbs placed on a wooden board. The temperature of the bulbs can higher than 37 degrees Celsius, which corresponds to human body temperature. The wooden board allows us to make sure that the bulbs are well isolated and so they will be distinguished by the cameras. We placed 36 bulbs as a $6 \times 6$ matrix. Each bulb was separate from its row-wise and col-wise nearest neighbors by 160 mm. The size of the grid is 1 meter x 1 meter. For the calibration purpose, we should make sure that all lights have almost the same brightness.

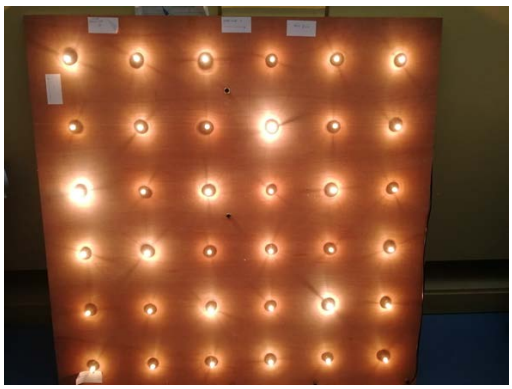


**Figure 3.7:** Calibration grid

### 3.4.2 Stereo calibration

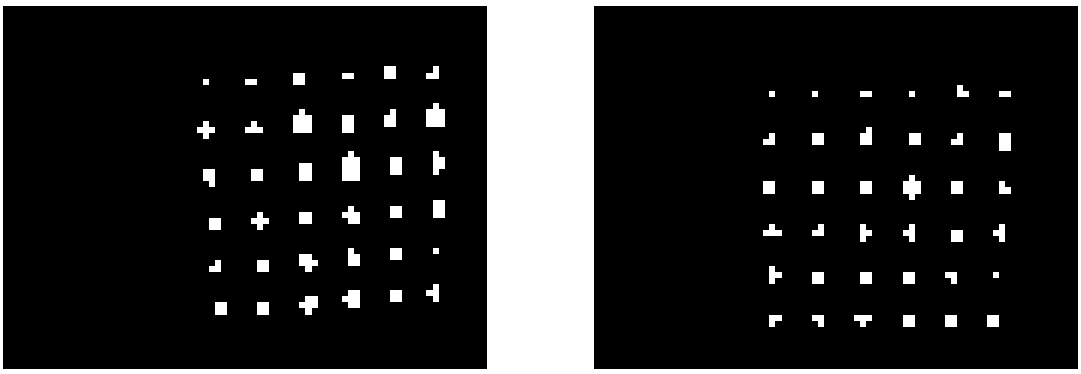
We used two fixed cameras separated by a distance of 140 mm.

To perform the stereo calibration, it is important to detect the points accurately. In the OpenCV calibration module[192], points are detected in pixel mode accuracy. This accuracy is sufficient for most of the cameras which have an acceptable resolution

but not in our case of low resolution and noisy images. Inspired by the work of Ellmauthaler et al. [189], we developed our own sub-pixel point detection tools. Because we manufactured a wooden support for the calibration grid, which is less susceptible to drive the heat, a simple grey-scale threshold is sufficient to separate the bulbs and the background.

#### 3.4.2.1 Calibration features localization

Given $N$ views, the goal is to extract and locate the points through sub-pixel accuracy. The images furnished by the FLIR sensor are in 16 bits, so first, we normalized the image pixels in the range [0,255]. The image is then binarized using a threshold estimated by the Otsu method[193] (Fig. 3.8). We extracted then all the connected components, estimated their centroid, and give them a label. We get the list of the points located in pixel accuracy, but, as shown in Fig. 3.8, the points are not always well located. The sub-pixel accuracy was obtained by computing the center of mass on the gray-level information around each centroid. This process is performed on all the original thermal images.



(a) Left image (b) Right image

**Figure 3.8:** Stereo pairs threshold image

#### 3.4.2.2 Parameters estimation

The estimation of the parameters was performed once we have detected all the 36 points in all the image pairs. We used these points as inputs for calibration methods available in the OpenCV calibration module[192]. As outputs we got intrinsic parameters such as cameras matrices ($M_l$ and $M_r$) and distortion vectors, and extrinsic parameters such as fundamental matrix $F$, essential matrix $E$, rotation matrix $R$ and translation matrix $T$. Our calibration method took into account radial and tangential distortion.

### 3.4.3 Robustness of the calibration

The main particularity of our calibration is the low resolution and the noise of our images. It can be easily shown that if the number of input image pairs used for calibration is too low, the noise can have a direct impact on the estimation of the parameters, i.e., if we perform the calibration on two different sets of input images, it is highly probable that to get different values of stereo calibration outputs. This is mainly due to the difficulty of detecting the positions of the points in the images accurately. If we increase the number of input image pairs, the calibration becomes more robust to noise in the sense that if we perform the calibration on two different sets of input images, we will get similar estimated parameters. But in this case, we also increase the difficulty and the handling time of the process. A trade-off must be found between accuracy or Robustness and handling facility.

To estimate the optimal number of images to get a robust calibration, we used the following bootstrap method.

— First we create a set of 100 image pairs of the calibration grid. We verified that in these images, all the 36 points were visible.

— The Robustness of the calibration using $x$ pairs of input images was estimated by:

1. We randomly selected $x$ pairs of images from the 100 and performed the calibration. The calibration outputs are stored.
2. We repeated this process 1000 times in order to get 1000 calibration outputs.
3. The dispersion of the 1000 calibration outputs was a good indicator of the Robustness of the calibration.

— This process was repeated for several $x$. $x \in \{5, 10, 15, ..., 40\}$

From all the different calibration outputs, we chose to observe more particularly four specific outputs, which are:

— the estimated focal length

— the RMS re-projection error

— the average epipolar error

— the estimated baseline: this value can be easily compared to the ground truth, 140 mm.

## 3.5 Results

Through the bootstrap method, we can determine how the values converge to a certain value. We use box plots to observe the dispersion and to detect outliers.

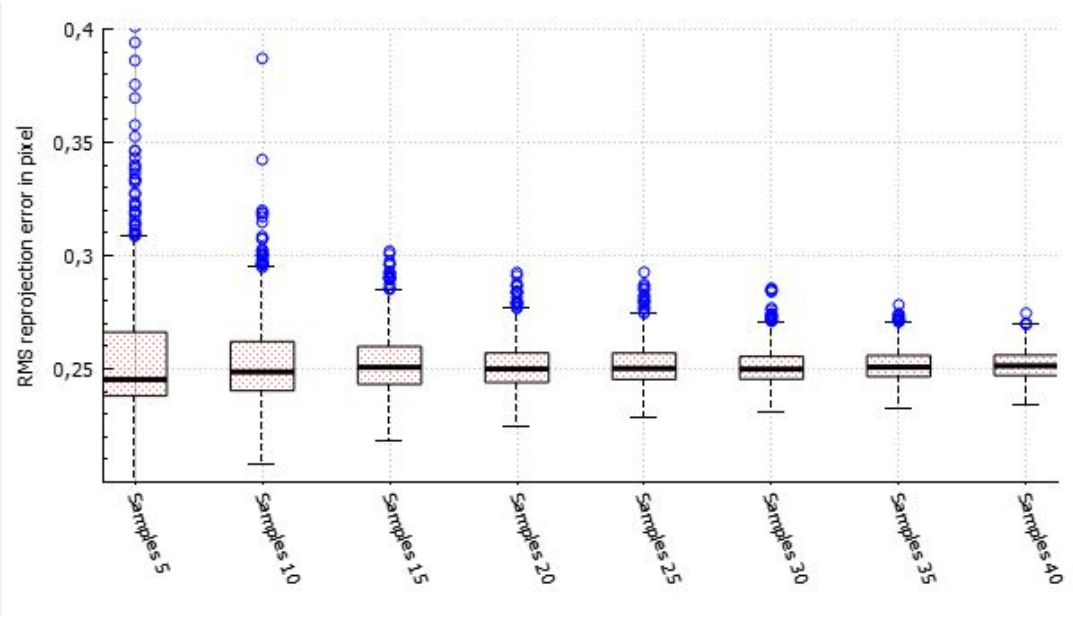



Figure 3.9: RMS

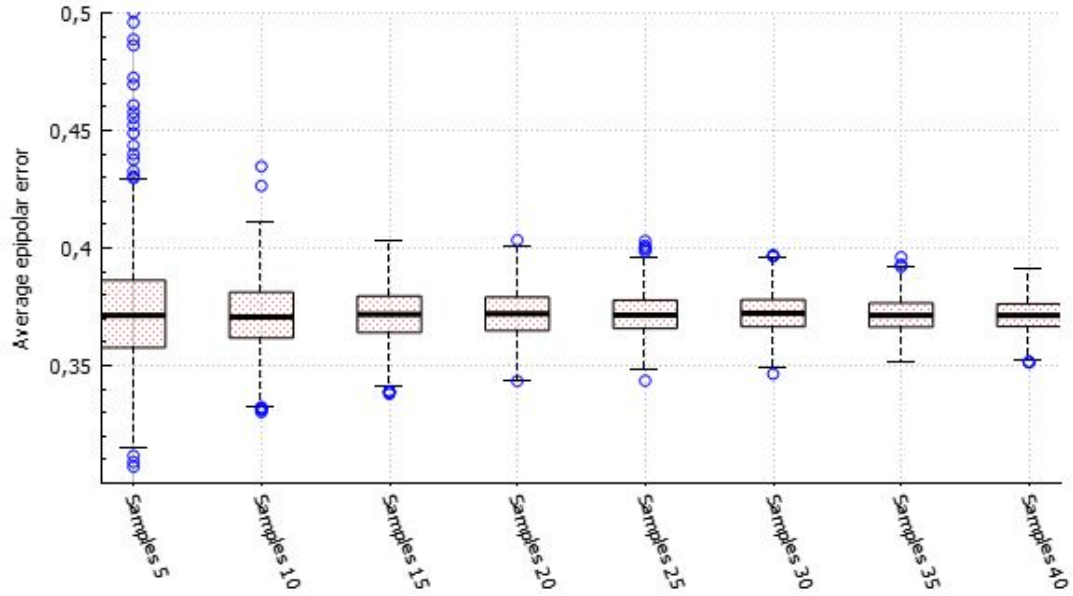



Figure 3.10: Average

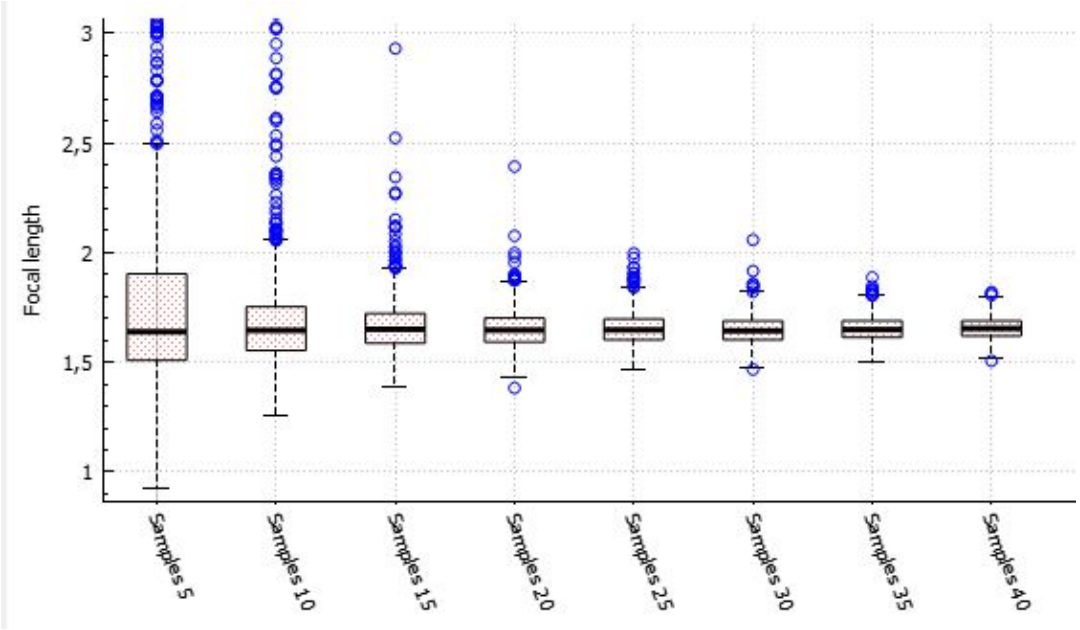



Figure 3.11: Focal length

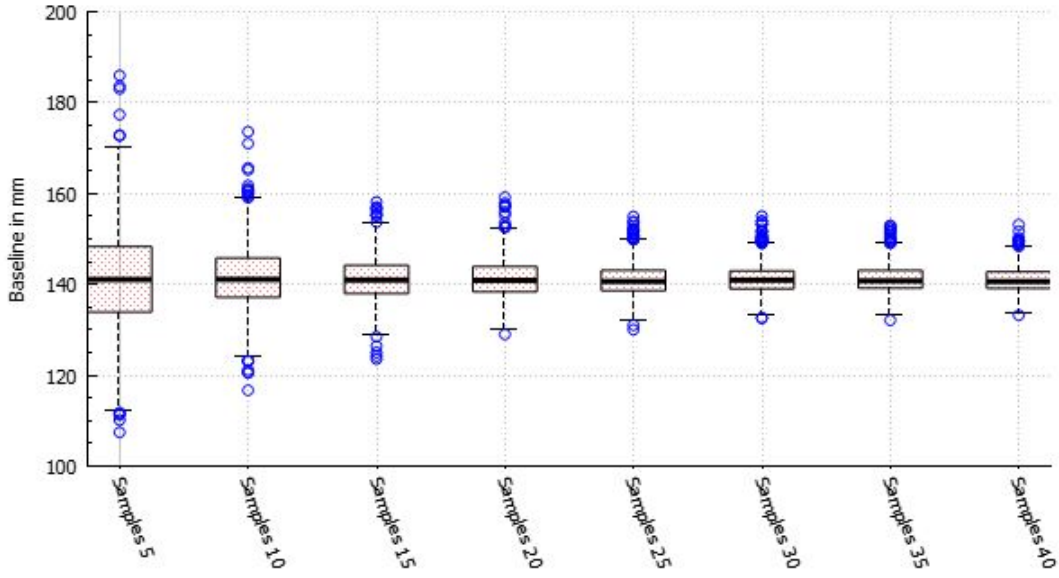



Figure 3.12: Baseline

In Figs. 3.9, we can see that the median distribution of the calibration results is relatively similar, regardless $x$ and close of the ground truth for the baseline (140 mm). However, we can see that if $x \leq 30$ image pairs, the dispersion of the calibration results is very high with many outliers. From 35 image pairs, we can see that the dispersion is stabilized and relatively small. We can thereby consider that from 35 image pairs, the calibration will almost estimate the same parameters regardless of the input images, and so our calibration can be considered robust.

Concerning the accuracy, with 35 image pairs, the focal length and the baseline were estimated relatively precisely. The RMS seems to be stabilized at 0.25 pixels, which correspond to an RMS of 1 pixel in $320 \times 240$ resolution. This could be considered relatively important, but in our case, we face very noisy images, which is not the case in high-resolution images. Moreover, if we would apply our method directly to high-resolution images, we would probably have better accuracy than the classical methods because of our noise robustness and sub-pixel feature localization.

We have computed the mean re-projected error to compare our method with previous ones because the previous methods use this metric for comparison. The results are shown in Table 3.1. It can be noticed that although we have a very low-resolution, we are able to compete with other methods that have higher resolutions.

| Method | Camera | Resolution | MRE |
|---|---|---|---|
| Gschwandtner et al. | PathfindIR | 360 x 288 | 0.4918 |
| Yang et al. | GUIDE IR112 | 320 x 240 | 1.2214 |
| Vidas et al. | Miricle 307K | 640 x 480 | 0.3031 |
| Proposed method (35 images pair) | FLIR Lepton 2 | 80 x 60 | 0.20 |

**Table 3.1:** Comparison of our method with others

## 3.6 Conclusion

In many applications, thermal images are used to characterize hot objects. In such a situation, it is important to locate an object in 3D space. This can be done under a stereoscopic vision. However, this later needs an accurate stereo camera calibration. This calibration is made more difficult when the images come from low-resolution cameras ($80 \times 60$ pixels in our case).

To overcome this difficulty, we first proposed a calibration grid composed of a matrix of 36 bulbs placed on a wooden board. We proposed a method to locate features on the thermal images with sub-pixel accuracy in a second step. Finally, we estimate the optimal number of image pairs, which has to be used for a robust calibration.

We have shown that in low 80x60 pixels resolution context, we can get a robust calibration. We can retrieve the calibration parameters and some physical values such as the cameras' focal length or the baseline between the cameras.

Our other main contribution concerns the estimation of the number of images needed to get a robust calibration. There was no study in the state of the art to estimate the optimal number of images that give almost always a robust calibration. Calibration is generally performed only one time without any warranty of the relevance of the obtained calibration parameters. We have shown that in our context, with 35 images it is always possible to get a robust calibration.

Generally, once the camera is calibrated, it is possible to perform the 3D reconstruction. After a matching process, this is done to associate each point in the left image to a point in the right image. Due to our images' low resolution, some techniques applied to high resolution visible and thermal images may not work well. In the next chapter, we will investigate how to perform stereo matching using our low-resolution thermal images.

# 4 Stereo vision

> One accurate measurement is worth a thousand expert opinions.
>
> Grace Hopper

## 4.1 Introduction

Detection of people is crucial in computer vision for security or safety applications (intrusion detection, pedestrian collision detection, fall detection). In this chapter, we used the stereo cameras we set in the previous chapter.

To perform 3D vision, there are prior steps: features extraction, stereo matching, and then stereo triangulation. This chapter describes each of these steps in the context of very low-resolution thermal images. Our method called $ST$ (for Stereo Thermal) is based on Phase congruency and Phase correlation.

The chapter is structured as follows: Section 4.4 presents our framework ST. It details the features extraction method for thermal images based on phase congruency, the stereo matching method, and the sub-pixel matching method based on phase correlation. In Section 4.5, we discuss, analyze, and explain these results. Finally, section 4.6 concludes the chapter and gives perspectives.

## 4.2 Backgrounds

As we have seen in the previous chapter, there are in the case of stereo-vision some relationships between left camera to right camera and both to the world. The relationship between a 3D point and the respective two projection onto the left and right image plane can be expressed through epipolar geometry. This part is the continuation of the Section 3.2.2.

The standard epipolar geometry can be seen in Fig 4.1. The two cameras centers are $O_l$ and $O_r$. When the two cameras are parallel the epipolar points $e_l$ and $e_r$ and the projected points $P_l$ and $P_r$ are in the same lines as shown in Fig 4.2. If the points $P_l$ and $P_r$ represent the projections on the 3D point $P$, they should be in the same epipolar line.

Most the time we do not have the position of $P$. In order to determine its position we must first determine the position of $P_l$ and $P_l$. But how to choose $P_l$ and

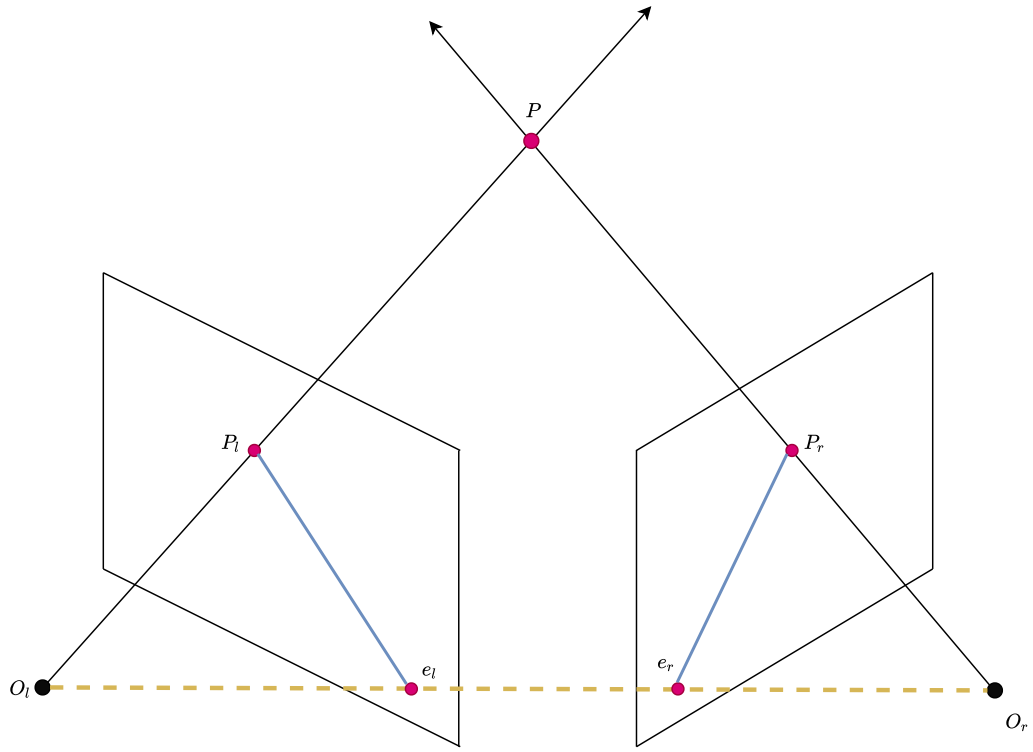



**Figure 4.1:** Given a 3D point $P$ and its projections $P_l$ and $P_r$ onto respectively left and right image. The dotted orange line is the baseline (the distance between the two cameras) while the blue lines are the epipolar lines.

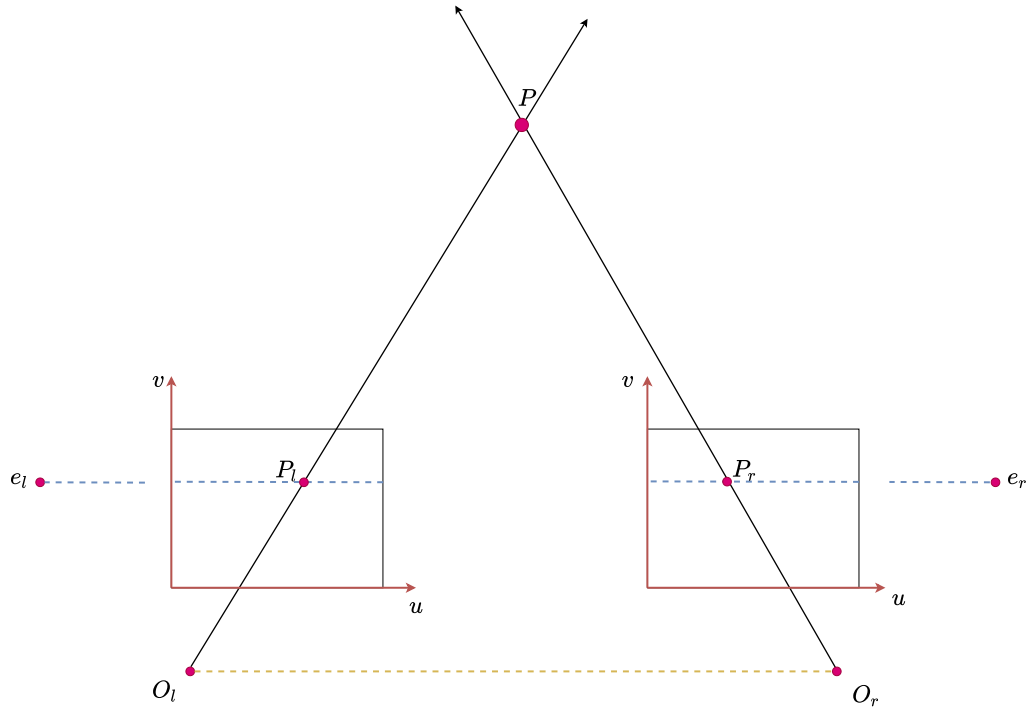



**Figure 4.2:** Epipolar geometry with parallel cameras

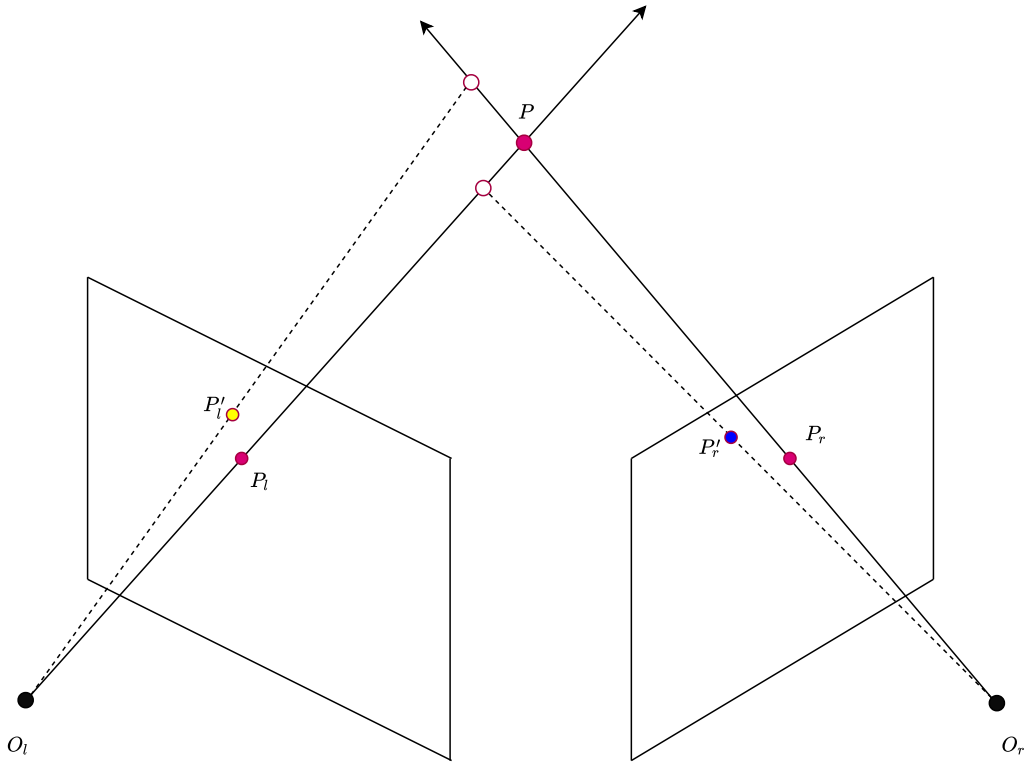



**Figure 4.3:** Epipolar geometry with parallel cameras

$P_r$? As shown in Fig 4.3, choosing wrong points to correspond (mismatch) can lead to determining wrong $P$. Matching $P'_l$ with $P_r$ or $P_l$ with $P_r$ will lead to a wrong $P$ position. Given two image $I_l$ and $I_r$ it is important to choose the right values. The matching is based on *how similar* is $P_l$ with $P_r$. But *how similar* depends on the application, the type of image and how fast the matching should be done.

A prior strong constraint for matching can be epipolar constraint described in equation 3.19.

## 4.3 State of the art

### 4.3.1 Features

As mentioned before, the goal of the stereo reconstruction is to reconstruct the 3D aspect of a scene by the triangulation of corresponding points seen by both cameras. The feature extraction aims to provide some robust, informative, and non-redundant items (usually 2D points, but also lines, contours, regions) from the images for future matching and triangulation. A feature can be either global or local. Depending on the application, one can consider local features or global features.

Generally, global features are used in image retrieval, object detection, and classification. They are based on the fact that humans can easily recognize objects with

a single glance [194]. The most common global features descriptors are Histogram of Oriented Gradient [195, 196], Invariant moments [197, 198] and Co-occurrence Histograms of Oriented Gradients [199, 200]. The main drawback of these features is that their extraction is difficult and computationally costly. Since they are global, they are also difficult to use directly in stereo-vision. Usually, they serve to detect and isolate some objects of interest in the images. Then some local features are extracted in the region around these objects.

Unlike global features, local features are easier to extract. In this section, we will combine descriptors and keypoints. While a keypoint represents a 2D position and other information such as scale and orientation, as for the descriptor, it represents the visual description of the feature. Most of the time, an image feature is built by combining a descriptor with a keypoint.

In the literature, the most common local features methods are Harris corner detector [6], KLT [7], phase congruency [10, 11], FAST [8] and BRIEF [9]. Fig 4.4 displays some advantages and drawbacks of some of these local features. These methods can be used as-is for the extraction of features in thermal images even if they seem less suitable because these images are noisier and contain less information. So in the specific context of thermal images, the literature proposes some simpler features extraction methods such as thresholds and template matching [201], using an improved version of the Harris corner detector [177] or using the Canny edge detector [202]. However, it should be interesting to study the behavior of the classical methods applied to thermal images. In [12], the authors compare in this context some features detectors such as Harris, Canny, Difference-of-Gaussian, and phase congruency. They prove that, regarding thermal images, phase congruency can extract more features than the methods as mentioned earlier. This is why we **explored phase congruency as a feature extraction algorithm**.

Before to exhibit phase congruency in Section 4.4.1, we will explain a bit deeper some of the features extractors we cited earlier.

#### 4.3.1.1 Harris corner detector

In [203], Harris et al. propose a new feature detection based on corners and edges. The new feature is based on the Moravec corner detector. The main idea behind the corner is that it should be recognized at intensity values within a small window. Moreover, shifting the window in any direction should yield a large change in appearance. So in *flat* regions, there is no change in all directions. When the region is an *edge*, there is no change along the edge direction. When the region is a *corner*, there is a significant change in all directions.

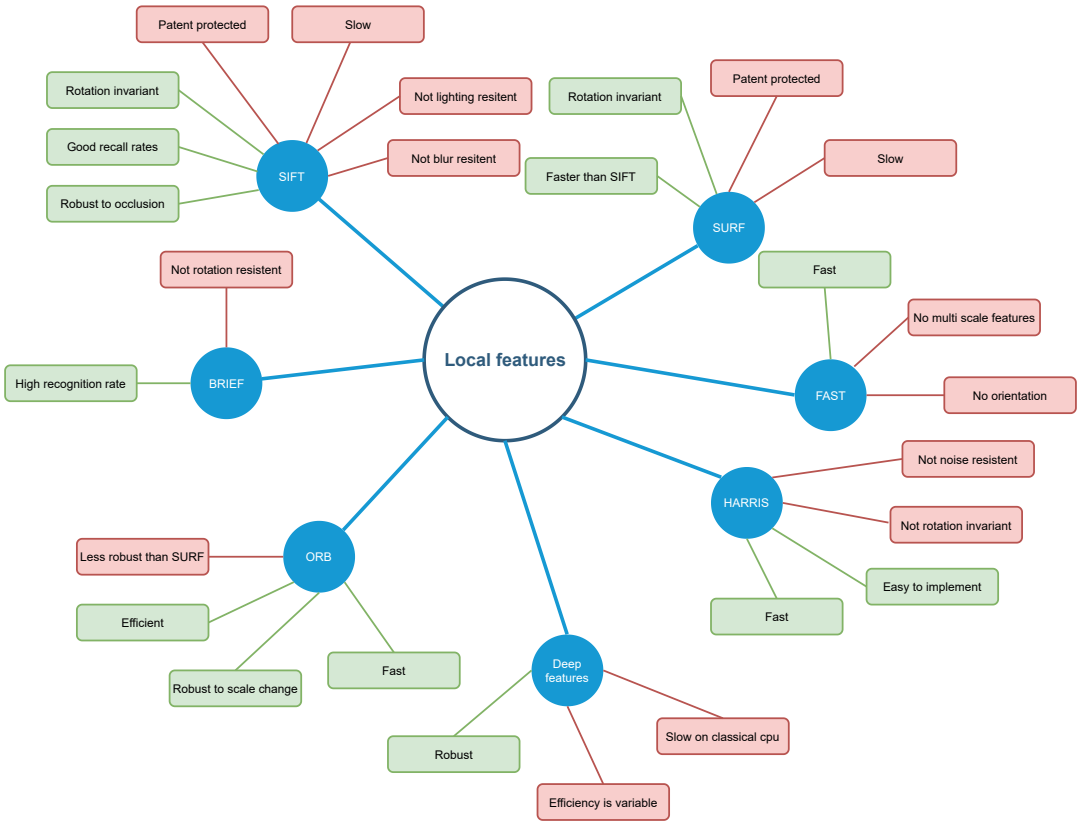



**Figure 4.4:** Advantages and drawbacks of some classical features extractors (not exhaustive)

Given an image $I$, the sum squared difference (SSD) is defined as follows:

$$E\left(\nu_x, \nu_y\right) = \sum_{x,y} w(x,y)\left[I(x+\nu_x, y+\nu_y) - I(x,y)\right]^2 \tag{4.1}$$

where $\nu_x$, $\nu_y$ are the $x$, $y$ coordinates of every pixel in our 3 x 3 window $w$. From Equation 4.1 by applying Taylor Expansion we have :

$$\begin{aligned} E\left(\nu_x, \nu_y\right) &\approx \begin{bmatrix} \nu_x & \nu_y \end{bmatrix} \left( \sum \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} \nu_x \\ \nu_y \end{bmatrix} \\ &\approx \begin{bmatrix} \nu_x & \nu_y \end{bmatrix} \Pi \begin{bmatrix} \nu_x \\ \nu_y \end{bmatrix} \end{aligned} \tag{4.2}$$

with $\Pi = \sum \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$.

Given that the detection of corners is linked to how large if $E$, it is possible to get the directions for both largest and smallest $E$. This can be done by computing the eigen values of $\Pi$ as follows:

$$\begin{aligned} \det \Pi &= \lambda_1 \lambda_2 \\ trace\ \Pi &= \lambda_1 + \lambda_2 \\ R &= \det \Pi - k(trace\ \Pi)^2 \end{aligned} \tag{4.3}$$

Where $\lambda_1$ and $\lambda_2$ are the eigenvalues of $\Pi$, and R is the score calculated for each window.

A given region is:

— *flat* if $|R|$ is small (that is to say $\lambda_1$ and $\lambda_2$ are small)

— a *edge* if $R < 0$ (that is to say $\lambda_1 >> \lambda_2$)

— a *corner* if $R$ is large

So usually, regions with a $R$ higher than a threshold are considered as corner, which are good tracking points. J. Shi and C. Tomasi propose a variant in which the score is computed by [204]: $R = min(\lambda_1, \lambda_2)$.

In their paper, they demonstrated experimentally that this score criterion was much better than Harris' score. The main advantage is that the implementation is easy and fast. Unfortunately, it is not robust to noise.

#### 4.3.1.2 Kanade–Lucas–Tomasi (KLT) feature tracker

In [205], Lucas et al. propose a feature extraction method based on spatial intensity information. This method is based on Harris Corner Detector and the value of eigenvalues $\lambda_1$ and $\lambda_2$. Given two image $I_1$ and $I_2$ two gray-scale images. Let $p = [p_x, p_y]^T$ be a pint in $I_1$. The main idea is to find $q$ a point in $I_2$ in such way that $q = p + d$. $d$ is the velocity or the optical flow of the image at $p$. The residual function is defined as follows:

$$\begin{aligned} \epsilon(d) &= \epsilon(d_x, d_y) \\ &= \sum_{x=p_x-w_x}^{p_x+w_x} \sum_{y=p_y-w_y}^{p_y+w_y} \left[I_1(x,y) - I_2(x+dx, y+dy)\right]^2 \end{aligned} \tag{4.4}$$

where $(2w_x + 1) \times (2w_y + 1)$ is the window size.

Most of the time, KLT is implement using pyramidal feature tracking.

#### 4.3.1.3 Scale-invariant feature transform (SIFT)

In [206], Lowe propose a robust feature detector from multi-scale oriented patches. Their objective is a feature detector invariant to uniform scaling, orientation, illumination changes. It is based on pyramids to be scale invariant. Given an image $I$, the author successive apply the processes (Fig 4.5) as follow:

— Construction of Scale space:

$$L(x, y, \sigma) = G(x, y, \sigma) \star I(x, y) \tag{4.5}$$

where $G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp^{-(x^2+y^2/2\sigma^2)}$

— Computation of the Difference of Gaussian (DoG) by approximating the Laplacian of Gaussians:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k_i\sigma) - G(x, y, k_j\sigma)) \star I(x, y) \\ &= L(x, y, k_i\sigma) - L(s, y, k_j\sigma) \end{aligned} \tag{4.6}$$

where $k_i\sigma$ and $k_j\sigma$ are different scales of Gaussian blurring.

— Locate the Difference of Gaussian Extrema by scanning each DoG image.

— Sub-pixel localization of the potential features points by using 3D curve fitting on $D$. This is done using Taylor Series Expansion. For example if $\zeta = (x, y, \sigma)$ then:

$$D(\zeta) = D + \frac{\delta D^T}{\delta \zeta}\zeta + \frac{1}{2}\zeta^T \frac{\delta^2 D}{\delta \zeta^2}\zeta \tag{4.7}$$

By differentiating and setting to 0, we have $\hat{\zeta} = -\frac{\delta^2 D^{-1}}{\delta \zeta^2}\frac{\delta D}{\delta \zeta}$ to localize $(x, y, \sigma)$.

— Discarding candidates with poorly determined locations but have high edge responses. This is done through the analysis of the principal curvature. They compute the trace and the determinant of the Hessian $H$ such as follows:

$$\begin{aligned} Trace(H) &= D_{xx} + D_{yy} = \alpha + \beta \\ \det(H) &= D_{xx}D_{yy} - D_{xy}^2 = \alpha\beta \\ R &= \frac{Trace(H)^2}{\det(H)} < \frac{(r+1)^2}{r} \end{aligned} \tag{4.8}$$

with $r = \frac{\alpha}{\beta}$. If $R$ for a candidate keypoint is larger than a threshold, that keypoint is poorly localized and hence rejected.

— Assignment of Keypoints orientations

— Keypoints building to output features

The main advantages of SIFT are locality, distinctiveness, quantity, efficiency, and extensibility.

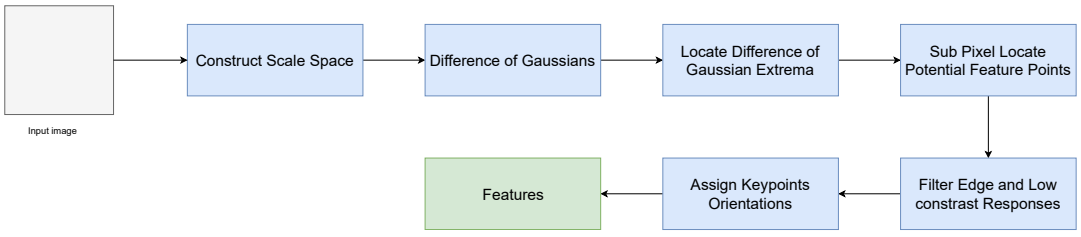



**Figure 4.5:** SIFT features computation overview

#### 4.3.1.4 Speeded up robust features (SURF)

In [207], Bay et al. propose a new fast feature detector, inspired by SIFT and based on fast computation of the Hessian matrix using integral images. The pipeline of SURF is similar to SIFT, with some differences. For example, while in SIFT scale-invariant features are detected with cascaded filters and Difference of Gaussians, SURF uses some filters to approximate Gaussian smoothing. SURF also uses an integer approximation of the determinant of Hessian. Using such techniques, SURF is faster than SIFT.

#### 4.3.1.5 Features from Accelerated Segment Test (FAST):

If SIFT or SURF are robust the time computation is very expensive. This is why, in [8], Rosten et al. propose a new feature detector. This feature detector is based on the brightness of a given point accordingly to its neighbors. They consider a circle of 16 pixels around a corner candidate $p$. $p$ is considered as corner if there is some neighbors $n$ of $p$ that are all brighter or darker than $p$. A neighbor $v$ will be labeled:

$$label(v) = \begin{cases} d & \text{if } I(v) \leq I(p) - t \text{ (darker)} \\ s & \text{if } I(p) - t < I(v) < I(p) + t \text{ (similar)} \\ b & \text{if } I(v) + t \geq I(p) + t \text{ (brigther)} \end{cases} \tag{4.9}$$

The rest of the algorithm use non-maximal suppression and enhance repeatability.

FAST is faster than Harris Corner detector (115%) and SIFT (195%). The main drawback of FAST is that it should be first trained to construct a decision tree.

#### 4.3.1.6 Binary Robust Independent Elementary Features (BRIEF)

In [208], Calonder et al. propose a feature descriptor which is a bit string description of an image patch constructed from a set of binary intensity tests. The binary intensity test is based on some a small number of pair-wise intensity comparisons. Given an image $I$ and a smoothed patch $\mathbf{p}$ the binary test is given by:

$$\tau(\mathbf{p}; \mathbf{x}, \mathbf{y}) = \begin{cases} 1 & \text{if } \mathbf{p}(\mathbf{x}) < \mathbf{p}(\mathbf{y}) \\ 0 & \text{otherwise.} \end{cases} \tag{4.10}$$

The BRIEF descriptor if an $n_d$-dimensional bitstring as follows:

$$f_n(\mathbf{p}) = \sum_{i=1}^{n_d} 2^{i-1} \tau(\mathbf{p}; \mathbf{x}_i, \mathbf{y}_i) \tag{4.11}$$

In some cases, BRIEF is faster and detect more features than SIFT and SURF.

#### 4.3.1.7 Oriented FAST and rotated BRIEF (ORB)

In [209], Rublee et al. propose a new feature descriptor which is free to use while SIFT and SURF have been patented. ORB is based on FAST and BRIEF feature descriptor and is rotation invariant and noise resistant. FAST features do not have orientation, and they need to be augmented with a pyramid scale to be more robust.

They perform multi-scale image pyramid and affect an orientation for each located keypoint.

In [210], Rosin define the moments of a given image $I$ and a centroid as follows:

$$m_{pd} = \sum_{x,y} x^p y^q I(x,y) \tag{4.12}$$

where $x$, $y \in [-r, r]$ , r the radius of the patch and the centroid $C$ is:

$$C = \begin{pmatrix} \frac{m_{10}}{m_{00}} & \frac{m_{01}}{m_{00}} \end{pmatrix} \tag{4.13}$$

and the orientation of the patch is given by $\theta$:

$$\theta = atan2(m_{01}, m_{10}) \tag{4.14}$$

So they modify FAST by computing the moments, the centroid and the orientation of the patch. Besides, ORB is not only a modified FAST but also a modified BRIEF. Indeed, they modify BRIEF to make it rotation-aware. For that, for any feature set of n binary tests at location $x_i$,$y_i$) (Equation 4.11), they define a $2 \times n$ matrix as follows:

$$\mathbf{S} = \begin{pmatrix} \mathbf{x}_1, & \cdots, & \mathbf{x}_n \\ \mathbf{y}_1, & \cdots, & \mathbf{y}_n \end{pmatrix} \tag{4.15}$$

By combining $\mathbf{S}$ with $\theta$ they obtain $\mathbf{S}_\theta = \mathbf{R}_\theta \mathbf{S}$ with $\mathbf{R}_\theta$ the rotation matrix. The BRIEF operator becomes:

$$g_n(\mathbf{p}, \theta) = f_n(\mathbf{p}) | (\mathbf{x}_i, \mathbf{y}_i) \tag{4.16}$$

#### 4.3.1.8 KAZE

In [211], Alcantarilla et al. propose KAZE a new feature detection. It is based on non-linear scale spaces allowing local adaptive blurring. In [212], they propose AKAZE by improving KAZE. They use Fast Explicit Diffusion in a pyramidal fashion to reduce KAZE computation load. Besides, the new descriptor is based on LDB [213].

#### 4.3.1.9 AGAST

In [214], Alcantarilla et al. propose AGAST, a feature detector based on FAST. They improve FAST by finding the optimal decision tree. While FAST is based on three classes *darker*, *similar* and *brighter*, they add three other comparisons *not darker*,

*similar* and *not brighter*. Compared to FAST, the main advantage of AGAST is that it does not need to be trained on the test images to find the optimal decision tree.

#### 4.3.1.10 Deep learning features

Recently, there is a lot of work about deep learning-based descriptors and features. We will talk deeply about deep learning in the Chapter 5 with super-resolution and Chapter 7 with activity recognition. The main idea is to train the networks to output high-level features.

Most of the time, the features are extracted from the one-but-last layer. These features can be used for K-means clustering [215], SVM [216] or as input of a sigmoid/softmax layer.

Unlike handcrafted methods that are generic, most of the time, the nature of the extract features depends on the type of the application. For example, the learned features from an image where a person is hiking can be different according to whether one wants to detect a person or identify the activity.

### 4.3.2 Stereo vision: features matching in the context of thermal images

The idea behind the stereo-vision is to exploit the difference between the two views, knowing more or less the position of each camera relative to the other one. Several method families are proposed in the literature for stereo computation: global methods such as dynamic programming [217], intrinsic curves [218] or local ones including block matching [219], gradient-based optimization [220] and features matching [221].

Among these methods, features matching seems to be the most adapted to a sparse features situation as in our specific low-resolution case. If most of the proposed methods have been developed for visible images, other authors propose an adaptation to thermal images.

In [177], after having extracted Harris corners, the authors perform features matching by computing correlation within a search window, discarding outliers, and regularizing the matched features spatially. In [176, 202], among several variants, the authors propose a relatively similar framework for pedestrian detection: after having extracted features with the Canny edge detector, they perform matching using cross-correlation and specialized methods adapted to a human silhouette. Features matching techniques are also used for some thermal image analysis in medical applications [222], [223]. So in such a condition, the extracted features must be very robust. In [224], the authors propose a framework to correct motion artifact due to body movement when recovering breast images. While the amplitude of the Fourier transforms gives the pixel intensity, the phase components represent the spatial information. Given

three consecutive frames $I_1$, $I_3$ and $I_3$, they combine their phase and amplitude to get a final matched image $I_3'$. The matching is performed using the method described in [225]. In [12], the authors extract the features from thermal images using Log-Gabor filters bank and then perform 1D matching through epipolar lines using the Lades similarity [14] as a matching criterion. So, processing thermal images using Fourier-based methods is a common way. But all of these works concern thermal images with a reasonable pixel resolution. **Is it possible to adapt or to develop a matching method that works on small resolution image in which the details and so the features are sparse and very blurry?**

All of these works concern thermal images with a reasonable pixel resolution (over $80 \times 60$). In [226], the authors state that the computation time of the phase congruency can be reduced by down-scaling the original from $320 \times 240$ to $160 \times 120$ and $80 \times 60$, but they have not explored this solution given the induced loss of efficiency. In addition to given images that present little information, the low resolution of the images has also a direct impact on the 3D reconstruction. Indeed being wrong of 1 pixel in a $4 \times 4$ image is not the same as being wrong of 1 pixel in a $1000 \times 1000$ image.

The information matching between left and right views of the stereo pair is a critical phase, since small errors at this step may yield significant errors in the 3D localization especially in the z direction as demonstrated below. Fig 4.6 shows the principle of stereopsis. Given two ground points $M$ and $N$ with respectively $M_1$, $N_1$ and $M_2$, $N_2$ their projections on the first and the second image, $M_1$ and $N_1$ are the same points in the first image while $M_2$ and $N_2$ are not the same in the second image. This difeerence is proportional to the disparity which is proportional to the depth difference. The distance between a 3D point and the stereo system can be determined as follows [227]:

$$z = \frac{\epsilon}{b/h} \tag{4.17}$$

where $b$ is the baseline of the stereo system (distance between the two cameras), $\epsilon$ the disparity function and $h$ the distance between the scene and the camera system. Deriving 4.17, we have:

$$dz = \frac{d\epsilon}{b/h} \tag{4.18}$$

The error $d\epsilon$ on the disparity has a direct impact on the precision in z, especially when $b/h$ is small. The influence of the baseline on the depth precision has been well studied in the literature. In [228], the authors empirically show that to determine the distance of a point accurately, the latter must not be further away more than $(10 - 15)$ $\times$ baseline. Moreover, in [229], the authors stated that using their stereo setup, a

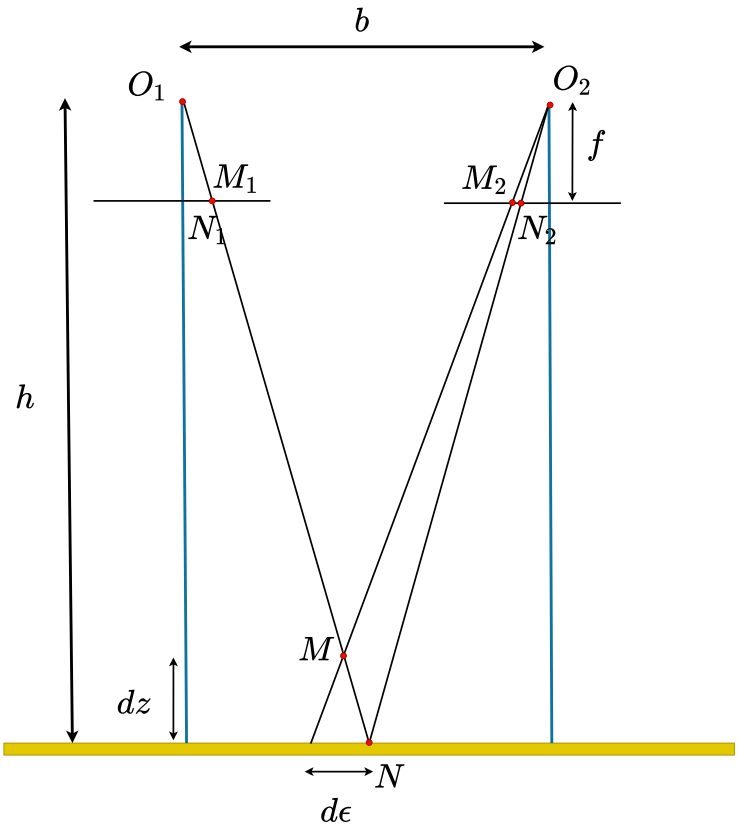



**Figure 4.6:** Stereopsis principle from [227].

disparity error larger than 0.1 pixels will result in a relative distance error of 2.5% for an object located at 60 meters. This is why, most of the time, the stereo-vision system uses a large baseline [230]. In this application, we chose to use a small baseline given indoor constraints. Nevertheless, some applications (as our indoor system) are constrained to use a small baseline. **Is it possible to gain in 3D accuracy even in case of small baseline and low image resolution?**

Many solutions exist to improve the accuracy of stereo-vision. Among them we have super-resolution [5] and sub-pixel matching [231]. Considering sub-pixel matching, we must first define the disparity map, which *"refers to the apparent pixel difference or motion between a pair of a stereo image"* [232]. In the classical methods, the difference is expressed in integer precision. Sub-pixel matching proposes to estimate e in floating precision. There are three main methods of sub-pixel matching: interpolation, fitting, and phase correlation. In interpolation methods, the sub-pixel disparity is estimated by searching the extreme of interpolated cost volume. These methods have two main drawbacks. First, the interpolation of the cost volume is computationally costly. Second, artifacts can be introduced by the interpolation even if most of the time, the accuracy is satisfactory [233, 234]. The fitting methods use disparity plane or cost volume to estimate the sub-pixel disparity. The methods using disparity plane fitting involve segmentation constraints [235]. The cost volume fitting methods search the extreme of a parabola representing this cost volume. These methods are fast but not accurate. Unlike these two methods, phase correlation offers both high efficiency and accuracy. It is performed using fast Fourier transforms and other supplementary approaches [236]. The phase correlation is the normalized cross-power spectrum, so the matching corresponds to a peak, which has to be estimated [237] in sub-pixel accuracy. In the state of the art, all of the methods are performed for visible images. We proposed to compute the sub-pixel disparity for thermal images using phase correlation. To our knowledge, it is the first time a sub-pixel matching is performed in such a way in thermal images.

## 4.4 Materials and method

### 4.4.1 Features extraction

Features extraction was one of the critical points of our framework. Thermal images are characterized by a lack of texture, noisy, and since the resolution of our images is low, a robust method is necessary to extract features from such images. Besides the low-resolution aspect, other characteristics of our images had to be taken into account. Our cameras are uncooled, and thus, they are influenced by ambient temperature. The temperature drift is sometimes compensated in the camera by some corrections which introduce sudden brightness changes. More, this is camera dependent and is not synchronized, which leads to a time-varying difference in

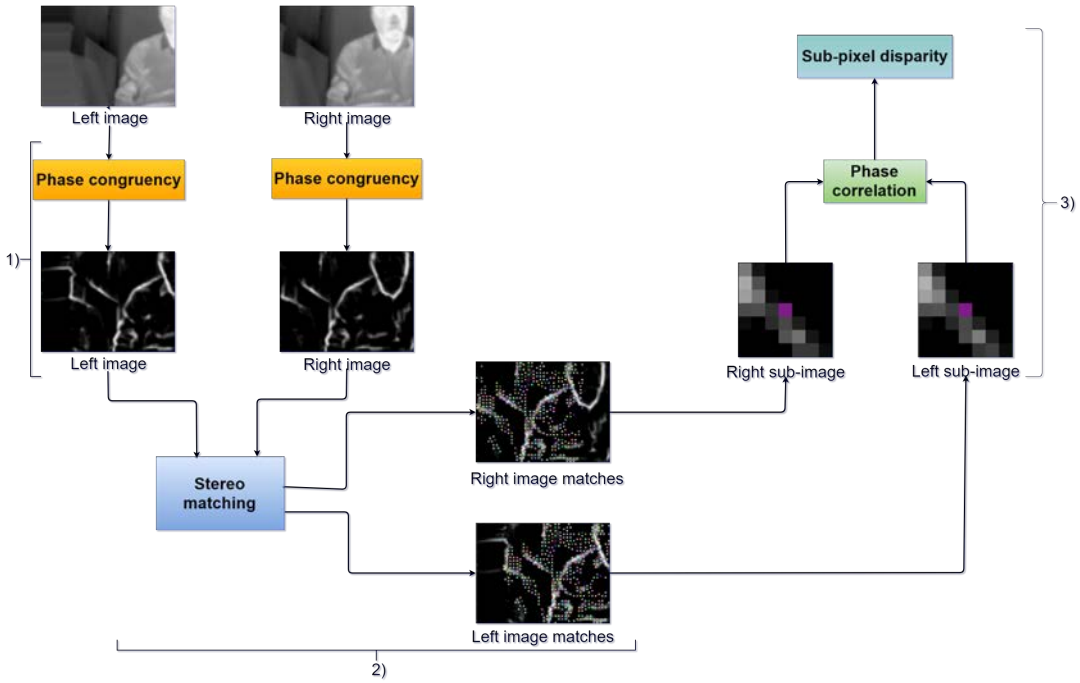



**Figure 4.7:** Sub-pixel matching framework: 1) robust features extraction; 2) rough features matching; 3) refined matching in sub-pixel accuracy.

brightness between the images. For all these reasons, we chose a method based on phase congruency, which takes into account primitives that are only linked to the difference of viewpoint between the two cameras.

To extract features, we adapted the phase congruency estimation method proposed by [238] and its variant proposed by [13] to our low-resolution thermal images. The 1D phase congruency is the ratio [10]:

$$PhaseCong(x) = \max_{\bar{\phi}} \frac{\sum_n A_n \cos(\phi_n(x) - \bar{\phi})}{\sum_n A_n} \tag{4.19}$$

Where $A_n$ represents the amplitude or energy of the $n^{th}$ Fourier component and $\phi_n(x)$ the local phase, which can be calculated using the Hilbert transform. Unfortunately, this version of the phase congruency was noise sensitive and yielded inaccurate features localization as proved by [239]. In [240], the authors modified a bit the equation to circumvent the previous version drawbacks:

$$PhaseCong_2(x) = W(x) \frac{\lfloor \sum_n A_n \left[ \Upsilon_n - \Psi_n \right] - T \rfloor}{\sum_n A_n(x) - \epsilon} \tag{4.20}$$

where $\Upsilon_n = \left| \cos(\phi_n(x) - \bar{\phi(x)}) \right|$; $\Psi_n = \sin(\phi_n(x) - \bar{\phi(x)})$, $W(x)$ is the frequency spread weighting, $T$ is a noise threshold and $\epsilon$ is a small value (eg. 1-e3) to avoid

division by 0.

The equation (4.20) can be extended to the two-dimensional image domain by applying it on several orientations $\theta$ after filtering the image by a bank of Log-Gabor filters. To reduce the computation cost due to the number of orientations and scales of the Log-Gabor filters bank, we implemented the variant proposed by [13] using a monogenic filter instead of Log-Gabor. The monogenic signal is a Riesz transform concatenated with a 2D signal. This was possible by constructing a monogenic filter in the frequency domain [13].

As results of the 2D extension, we get a set of $PC(\theta)$, the phase congruency at orientation $\theta$. The features are then estimated by combining all the $PC(\theta)$. This is done by computing the following values [240]:

$$a = \sum (PC(\theta)cos(\theta))^2 \tag{4.21}$$

$$b = 2\sum (PC(\theta)cos(\theta))(PC(\theta)sin(\theta)) \tag{4.22}$$

$$c = \sum (PC(\theta)sin(\theta))^2 \tag{4.23}$$

Combining $a$, $b$ and $c$ gives a hint of the strength of the feature. More precisely, the maximum moment $M$ and the minimum moment $m$ can be estimated by:

$$M = \frac{1}{2}(c + a + \sqrt[2]{b^2 + (a-c)^2}) \tag{4.24}$$

$$m = \frac{1}{2}(c + a - \sqrt[2]{b^2 + (a-c)^2}) \tag{4.25}$$

These moments are used to characterize the features. Higher is the maximum moment more significant will be the feature, and higher is the minimum moment more probably this feature point will be a corner. Because of the lack of information in images, we only took $M$ into accounts. So a feature was considered significant if $M$ was higher than a threshold $\gamma$.

As a result of this step, we produced two images $I_{fl}$ and $I_{fr}$, which are the images of the moment $M$ after applying the phase congruency on respectively the input images $I_l$ and $I_r$ (Fig. 4.7).

### 4.4.2 Stereo matching

In rectified image condition, the matching of a feature $F^l$ of $I_{fl}$ to its corresponding one $F^r$ in $I_{fr}$ is simplified to find the most similar feature along the corresponding

epipolar line. As suggested by [12], we used the Lades similarity [14] performed a $5 \times 5$ window as matching metric:

$$S(F^l, F^r) = \frac{\sum_j^{nb_{feat}} f_j^l f_j^r}{\sqrt[2]{\sum_j^{nb_{feat}} f{l}_j^2 \sum_j^{nb_{feat}} f{r}_j^2}} \tag{4.26}$$

where $nb_{feat}$ is the total number of features in the selected window.

However, compared to their work, we took into account more matching constraints than only the similarity, epipolar constraints and left-rigth consistency:

— Uniqueness: a feature in the left image was matched with only one feature in the right image.

— Continuity: the disparity must vary smoothly.

— Ordering: for a couple of matched features $f_{1l} \leftrightarrow f_{1r}$ , $f_{2l} \leftrightarrow f_{2r}$ (the symbol $\leftrightarrow$ represents the matching relationship), if $f_{1l}$ is at the left of $f_{2l}$ we ensured that $f_{1r}$ was at the is of $f_{2r}$.

We also took into account the type of features and their orientation in the matching process.

We tried to reduce the computation time using a disparity range of $d$. This range can be obtained if there is prior knowledge about the scene or it can be deduced by the disparities values of the previous frame in a frame by frame framework.

### 4.4.3 Sub-pixel matching

Phase correlation is a well-known method allowing to get the translation of a signal. The translation between two images in the spatial domain is expressed in the frequency domain using the Fourier transform. Let $I_1$ and $I_2$ be two $M \times N$ images, $M$ and $N$ odd for mathematical simplicity. These images can be expressed like $I_1 = I(x, y)$ and $I_2 = I(x + \delta x, y + \delta y)$ with the translation $(\delta x, \delta y)$. Let $m$ and $n$ defined as $M = 2m + 1$ and $N = 2n + 1$. By computing their respective Fourier transform $F_1(u, v)$ and $F_2(u, v)$, the normalized cross-power spectrum is given by:

$$R(u, v) = \frac{F_1(u, v)F_2^\star(u, v)}{|F_1(u, v)F_2^\star(u, v)|} \tag{4.27}$$

where $u = -m, ..., m$, $v = -n, ..., n$ and $F_2^\star$ is the complex conjugate of $F_2$.

Given that the most relevant components in the phase correlation matrix are the low-frequency ones, certain authors propose to filter $R$ by a rectangular low-pass function of size $U$ [237]. They also proved that the ratio of $\frac{U}{M} = 0.5$ is the one that gives the best accuracy.

The Phase-only correlation is the inverse discrete Fourier transform of $R$ and is defined as follows:

$$r(x,y) = \frac{1}{N \cdot M} \sum_{u=-m}^{u=m} \sum_{v=-n}^{v=n} R(u,v) \exp(-2\pi iux/M) \exp(-2\pi ivy/N) \tag{4.28}$$

where $u = -m, ..., m$ and $v = -n, ..., n$.

The peak position, in $r$, corresponds to the translation along $x$ and $y$ directions, but this position is in integer precision.

Some authors propose to recover the displacement in sub-pixel precision from the information contained in $r$. There are globally three ways to estimate the sub-pixel displacement accurately: the detection of the peak by local least square fitting on selected spectral components in $R$ [236], by fitting a closed-form analytic model to the correlation peak [237], or directly from the data of $r$ using a peak detection formula assuming a peak model [241, 242]. We implemented this last principle.

The correlation peak was modeled as a cardinal sine affected by the low pass filter:

$$PeakModel_{xy} = r(x,y) \approx \alpha \frac{\sin(\frac{V}{M}\pi(x+\delta_x))}{\pi(x+\delta_x)} \times \frac{\sin(\frac{V}{N}\pi(y+\delta_y))}{\pi(y+\delta_y)} \tag{4.29}$$

where $\delta_x$,$\delta_y$ are respectively the displacements along $x$ and $y$ axis, $\alpha$ is the peak value ( $0 \leq \alpha \leq 1$) and $V = 2U + 1$.

Given that we used stereo cameras, by rectifying the images, we were able to neglect $\delta_y$. So the phase correlation function became:

$$PeakModel_x = r(x) \simeq \alpha \frac{\sin(\frac{V}{M}\pi(x+\delta_x))}{\pi(x+\delta_x)} \tag{4.30}$$

The main idea was then to recover $\delta_x$, the max peak in sub-pixel accuracy, from the sample values in $r$. A robust solution has been described in [242]. Let consider the case in Fig. 4.8, the peak model max is close to the sample $x = p$ which has the highest values $\alpha$. We also considered two other measurements located at $\pm d$ pixels away from $p$ at $x = p + d$ and $x = p - d$.

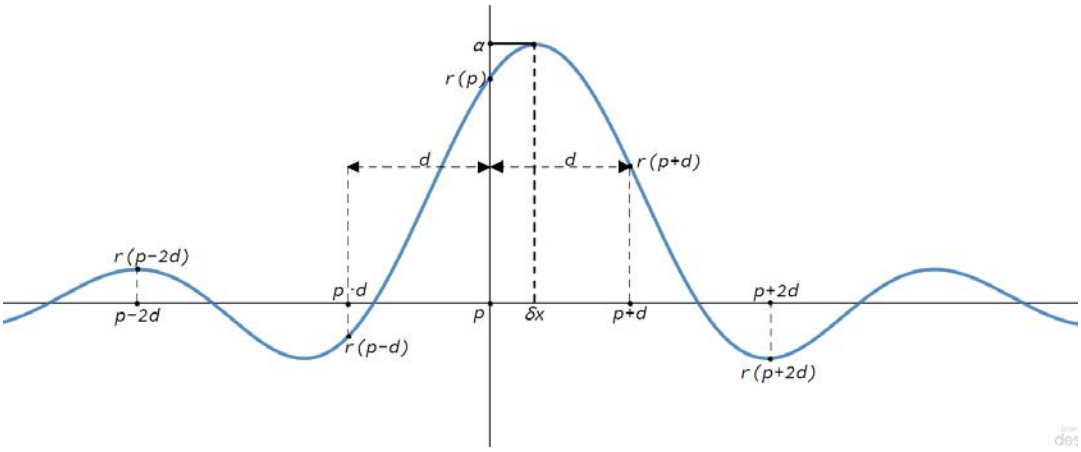



**Figure 4.8:** Peak model. In the vertical axis, the value of the phase correlation value at the point in the horizontal axis.

Using the model, the three points can be rewritten as:

$$\begin{cases} PeakModel_x(p-d) = r(p-d) \simeq \alpha \frac{\sin(\frac{V}{M}\pi(p-d+\delta_x))}{\pi(p-d+\delta_x)} \\ PeakModel_x(p) = r(p) \simeq \alpha \frac{\sin(\frac{V}{M}\pi(p+\delta_x))}{\pi(p+\delta_x)} \\ PeakModel_x(p+d) = r(p+d) \simeq \alpha \frac{\sin(\frac{V}{M}\pi(p+d+\delta_x))}{\pi(p+d+\delta_x)} \end{cases} \tag{4.31}$$

By combining these equations, it can be proved that:

$$\begin{aligned} (p-d+\delta_x)\, r\,(p-d) + (p+d+\delta_x)\, r\,(p+d) \\ -2\,(p+\delta_x)\cos\left(\frac{V}{N}\pi d\right) r\,(p) = 0 \end{aligned}$$

which can be rewritten as

$$v(p,d) = \delta_x u(p,d) \tag{4.32}$$

with

$$u(p,d) = r(p-d) + r(p+d) - 2\cos\left(\frac{V}{N}\pi d\right) r\,(p)$$

and

$$v(p,d) = 2p\cos\left(\frac{V}{N}\pi d\right) r(p)$$
$$-(p-d)\,r(p-d) - (p+d)\,r(p+d)$$

This allows us to estimate $\delta_x = u(p,d)^{-1}\, v(p,d)$.

However, in order to minimize the impact of noise, several observations with different values of $p$ and $d$ around the highest peak in r can be used. So it is possible to select $\chi$ values of $p_i$ and $d_i$ and then get $\chi$ equations $v(p_i,d_i) = \delta_x u(p_i,d_i)$ where $i \in \{1,2,...,\chi\}$. Resolving these equations is equivalent to minimize $\delta_x$ in:

$$\delta_x = \sum_{i=1}^{\chi} |v(p_i,d_i) - u(p_i,d_i)|^2 \tag{4.33}$$

This equation can be solved using Singular Value Decomposition (SVD) [243].

The existing phase correlation-based sub-pixel matching methods are applied on high resolution visible images (around $3000 \times 3000$ pixels [244]) using large sub-windows ($41 \times 41$ pixels) for the phase correlation estimation. We had to adapt this class of methods to our low-resolution problem. Let consider two matched points $F_l$ with coordinate $(x_l, y_l)$ and $F_r$ with coordinate $(x_r, y_r)$ respectively in $I_{fl}$ and in the shifted image $I_{fr}$. We knew that $y_l = y_r$ and $d_i(x) = x_r - x_l$. Then we defined two sub-images $Is_l(x_l, y_l)$ and $Is_r(x_r, y_r)$ with the same windows size $W$ around $F_l$ and $F_r$ (Fig. 4.7) and performed the refined phase correlation-based matching in these sub-images.

The main parameters of this method are the sub-images windows size $W$ and the number $\chi$ of observations used for the least square estimation of the sub-pixel displacement (4.33). Because of our low resolution we limited the number of observations $\chi$ to 6: $p_i = p \pm 1$ and $d_i \in \{1,2\}$.

## 4.5 Results and discussion

### 4.5.1 Datasets

In this chapter, we used two datasets: a thermal infrared video benchmark for visual analysis with high-resolution infrared images [245] and an own dataset (called Tvvlgo in the rest of the chapter) with images acquired by our system. Tvvlgo was created by placing the stereo system in the ceil of a room, and we collected 1000, 80x60 pixels images pair of a person moving in a room. We also used another third image where a person is sitting in from of the cameras. The dataset we used is available

online [246].

### 4.5.2 Feature extraction

In [11], the author proposes to use a threshold on the phase congruency moments of $\gamma = 0.3$. Because of our low-resolution images context, we feared that this threshold could not extract enough features. We tried other smaller thresholds at $\gamma = 0.1$ and $\gamma = 0.01$.

#### 4.5.2.1 Evaluation of the number of extracted features

To validate our choice of the features extraction method, we compared the implemented phase congruency to other standard features extraction methods using a low-resolution image given by a FLIR lepton 2 (Fig. 4.9-a). The authors in [226] have already made such comparison, but they only compared the phase congruency method to Harris corner detector, SIFT, Canny, and KLT. We wanted to go further. So we have compared our approach with the three thresholds (PhaseCong(0.01), PhaseCong(0.1) and PhaseCong(0.3)) to the OpenCV [247] implementations of ORB, BRISK, FAST, Shi Tomasi, SURF, AGAST, GFTT, and KAZE. Our framework was implemented using C++ compiled with GCC 7. We compared the methods according to 2 criteria: the number of extracted features and the execution time. For this later measurement, all the codes were executed on an Intel Core i7-3687U CPU. To get the execution time we used the C++ API **std::chrono::high_resolution_clock::now**. For each feature detector, the feature extraction process was completed 1000 times, and we computed the mean and the standard deviation of all computation times.

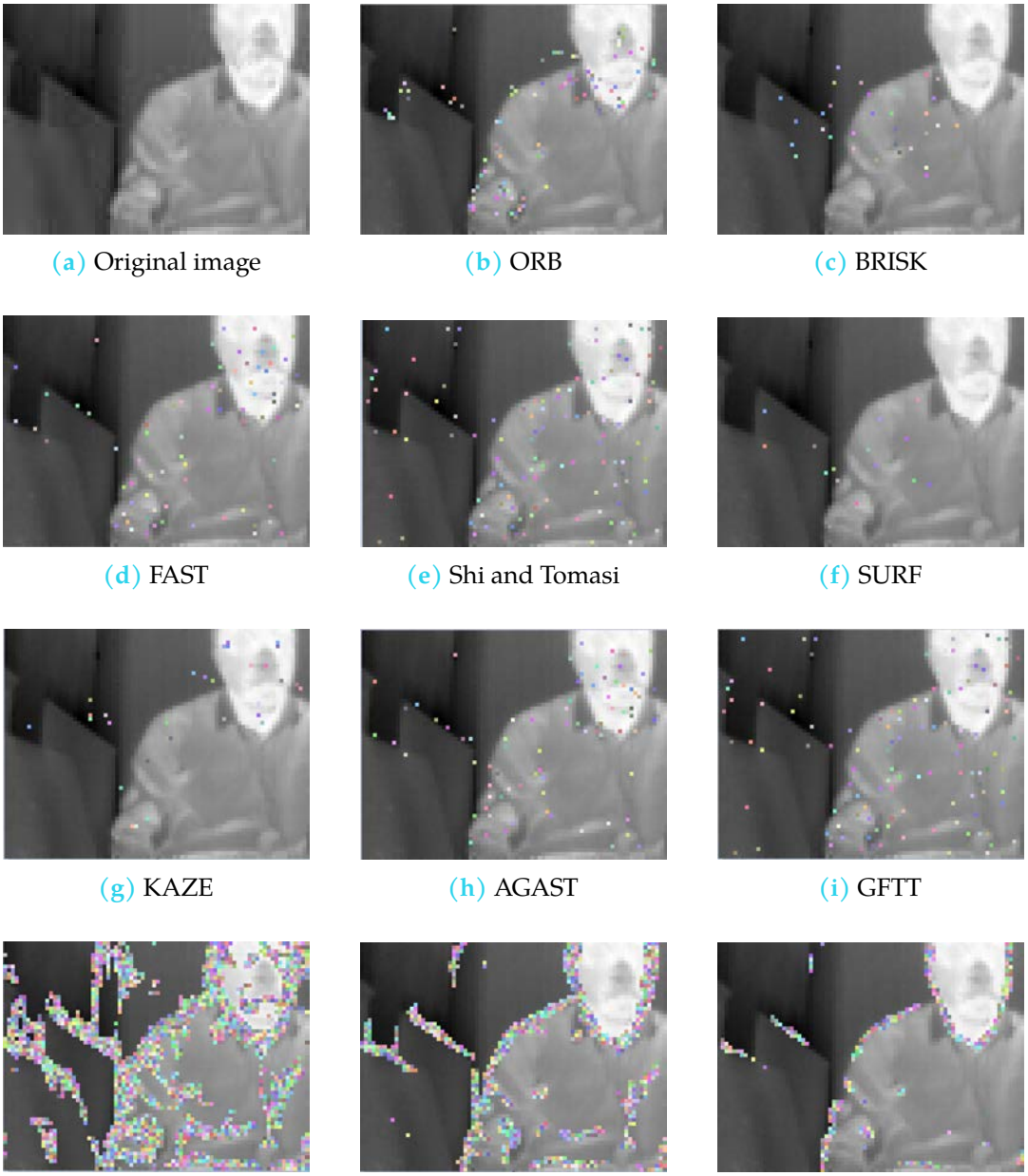
The extracted features can be seen in Fig. 4.9 and measurements are sampled in Table 4.1.

As illustrated by Table 4.1 Phase congruency can extract more features than other methods. Even using a high threshold, we could extract two times more features than Shi-Tomasi and GFTT. The counterpart of this advantage is a relatively high computation time compared to techniques such as FAST. Fortunately, the feature extraction for a pair of images took approximately 10 ms with Phase congruency, which is compatible with the 8 frames per second rate of our cameras.

These results are visually confirmed by Figure 4.9, where we extracted features from a low-resolution image using different feature extractors. It is noticeable that while most of the feature extractors try to extract robust features, phase congruency does the same by focusing on edges. Using the threshold $\gamma$, we could control the number of features returned by phase congruency. $\gamma = 0.1$ seemed to give a good trade-off between the sparsity and redundancy of features.

To conclude, even if Phase congruency is slower to compute than the other

classical method, it gave a higher number of features and so seemed to be unavoidable in processing our low resolution and texture-less images.



(a) Original image (b) ORB (c) BRISK

(d) FAST (e) Shi and Tomasi (f) SURF

(g) KAZE (h) AGAST (i) GFTT

(j) PhaseCong(0.01) (k) PhaseCong(0.1) (l) PhaseCong(0.3)

**Figure 4.9:** Features (colored dots) extracted by several feature extractor methods. The color of feature is generated randomly.

#### 4.5.2.2 Robustness to illumination change

One particularity of our cameras is that we sometimes noticed a sudden change in the brightness. It is probably a re-calibration of the sensor. The feature extraction method should so be robust to brightness changes. We evaluated this robustness using

| Feature extractions methods | Number of features | Execution time (us) |
|---|---|---|
| ORB (Oriented fast and Rotated Brief) | 117 | $245 \pm 20$ |
| BRISK (Binary Robust Invariant Scalable Keypoints) | 34 | $373474 \pm 350$ |
| FAST | 77 | $90 \pm 7$ |
| Shi Tomasi | 120 | $189 \pm 15$ |
| SURF (min Hessian = 300) (Speeded Up Robust Features) | 14 | $1436 \pm 100$ |
| AGAST (Adaptive and generic corner detection based on the accelerated segment test) | 84 | $202 \pm 29$ |
| GFTT (Good Features to track) | 120 | $350 \pm 32$ |
| KAZE | 33 | $13217 \pm 123$ |
| PhaseCong(0.01) | 1734 | $5224 \pm 480$ |
| PhaseCong(0.1) | 777 | $5224 \pm 480$ |
| PhaseCong(0.3) | 275 | $5224 \pm 480$ |

**Table 4.1:** Comparison between feature extractor methods ORB, BRISK, FAST, Shi Tomasi, SURF, AGAST, GFTT, KAZE and Phase congruency

a simulated brightness change as defined by Szelinski [248, Chapter 3]:

$$g(x) = \alpha f(x) + \beta \tag{4.34}$$

where f(x) and g(x) are the pixel values, and $\alpha$, $\beta$ control respectively the contrast and the brightness.

So we took 1000 images (80x60) acquired using lepton 2, then we simulated brightness changes by varying the parameter $\beta$ from 0 to 100. For each image, we computed the number of features detected for each $\beta$. Let $\Omega_\beta$ be the number of matches for a value of brightness control $\beta$. The Figure 4.10 represents the variation of the $\Omega_\beta$ according to $\beta$.

We also needed to evaluate the features extraction method in terms of re-detection.

For this, first, we computed the features at $\beta$=0. Then by increasing $\beta$, we estimated this robustness by counting the percentage of features still detected at the same location as for $\beta$=0. We called this percentage features a re-detection rate. Figure 6 shows the Features re-detection rate vs. brightness change for one image of the Tvvlgo dataset [246].

We compared PhaseCong(0.1) with other feature extractors such as ORB, FAST, SURF, AGAST, and GFTT. Our results also confirmed those of Hajebi et al [249] assessing that the features detected by phase congruency from thermal images present the advantage to be more stable than the others to illumination changes (Fig. 4.10 and 4.11). Figure 4.10 shows that given different values of illumination, phase congruency is the feature extraction method that can extract the highest number of features. Figure 4.11 shows that phase congruency is the method with the best re-detection rate. While detecting more features, phase congruency is also more robust to brightness changes than the other classical methods.

### 4.5.3 Stereo matching

To evaluate our stereo matching method, we have performed a stereo matching on 15 low-resolution $80 \times 60$ stereo pairs provided by our cameras (Tvvglo [246]). The stereo matching method explained in the section 4.4 was applied. To verify the accuracy of the method, we manually counted the number of mismatches given the number of matches. For each feature extracted in $I_l$, we visually verified if the matched feature in $I_r$ was inside a $5 \times 5$ window centered on the estimated disparity. The results showed that we had a percentage of mismatches of less than $1\%$ through all of the 15 image pairs.

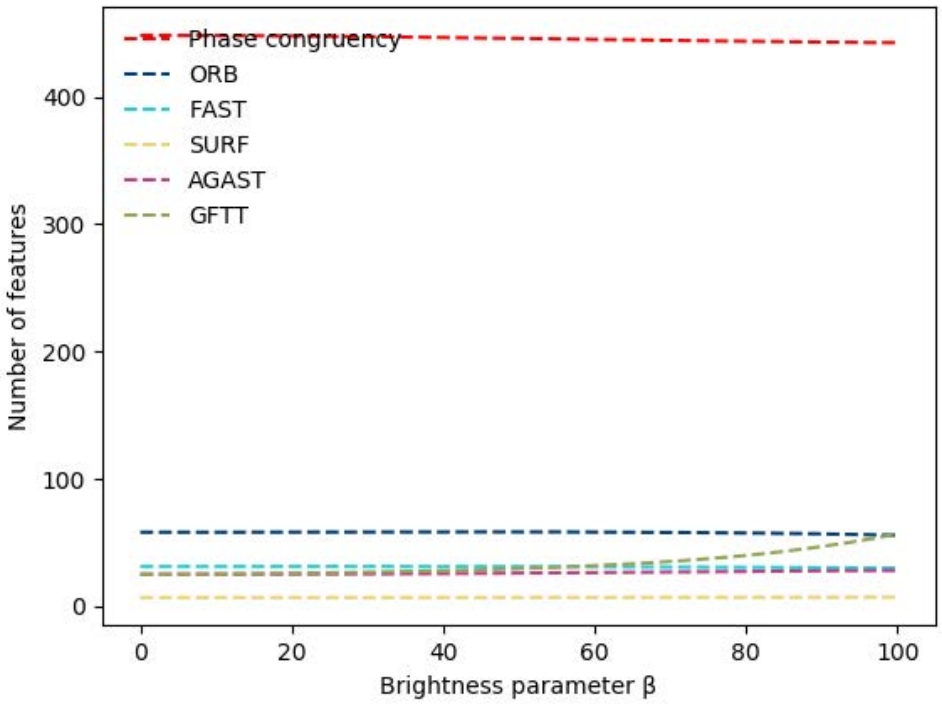



**Figure 4.10:** Average number of features detected for each image by some features extractors

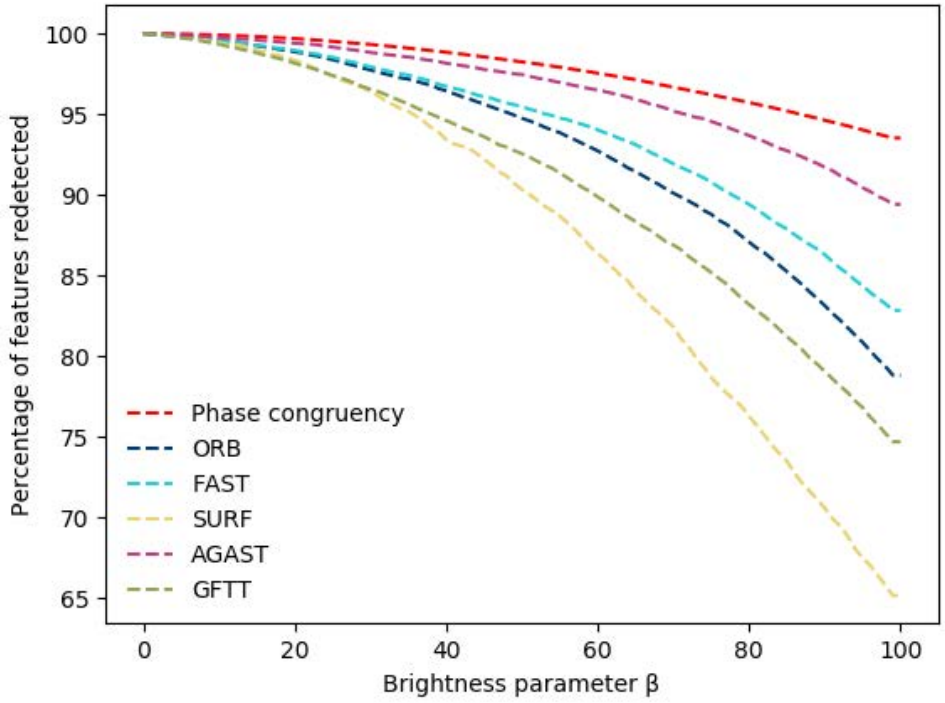



Figure 4.11: Features re-detection rate

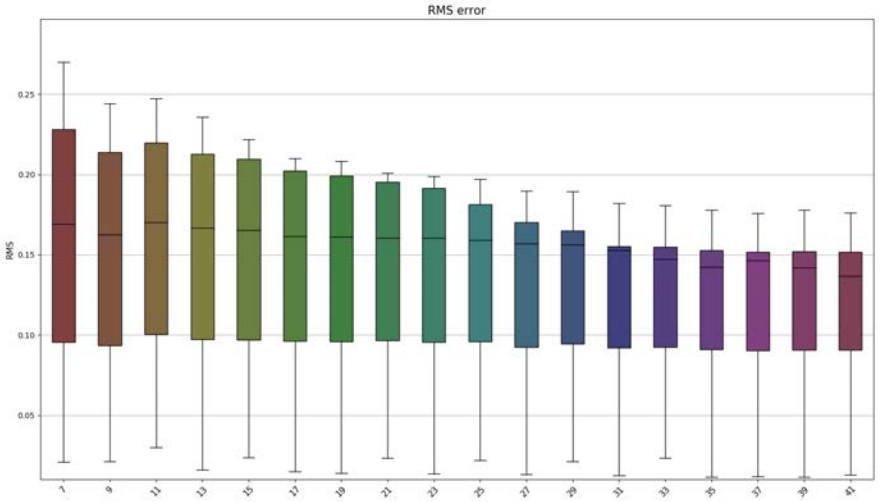
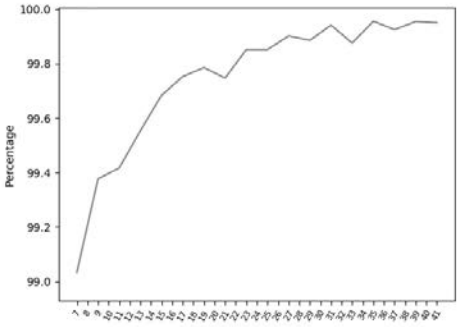
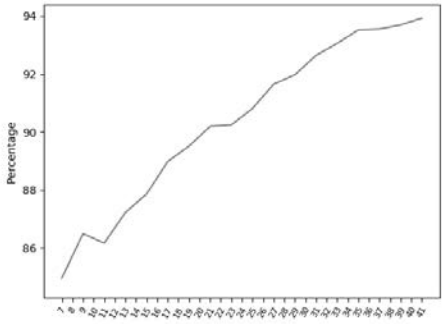


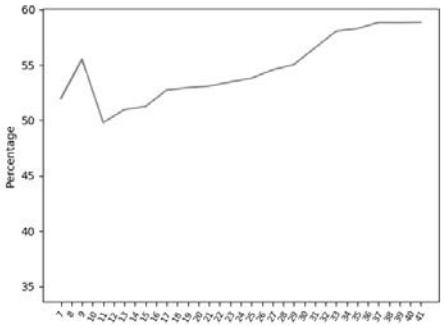

Figure 4.12: Box plots of the root-mean-square deviation (RMSD) of the matching error (in pixel) vs. sub-images window size



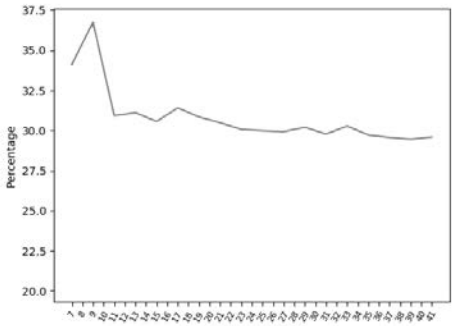

(a) Percentage of matching with an error less than 0.5 pixels




(b) Percentage of matching with an error less than 0.25 pixels

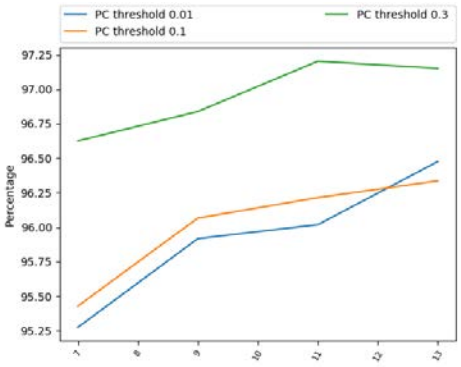



(c) Percentage of matching with an error less than 0.1 pixels

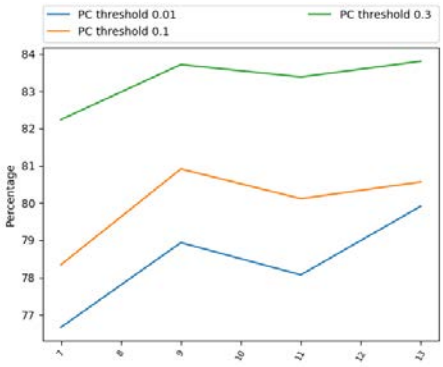



(d) Percentage of matching with an error less than 0.05 pixels

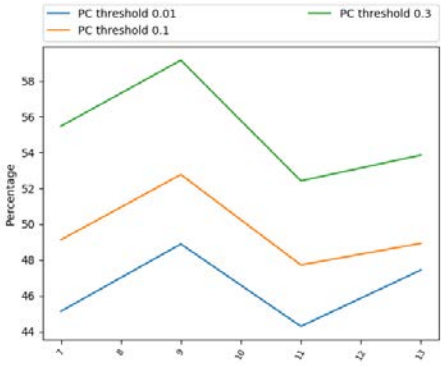
**Figure 4.13:** Impact of the sub-images window size (abscissa) on the rate of good matching (ordinate in %) respects to a specific level of precision



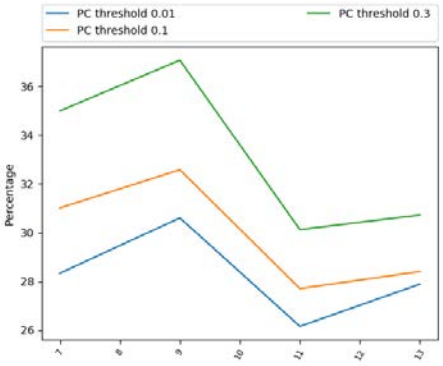

(a) Percentage of matching with an error less than 0.5 pixels




(b) Percentage of matching with an error less than 0.25 pixels




(c) Percentage of matching with an error less than 0.1 pixels




(d) Percentage of matching with an error less than 0.05 pixels

**Figure 4.14:** Percentage of correct matches (ordinate in %) with a given level precision $\tau$ according to the sub-images window size (abscissa) and the phase congruency moments threshold (PC threshold)

### 4.5.4 Sub-pixel matching

As mentioned in section 4.4.3, one of the key parameters is the size $W$ of the sub-images windows in which the phase correlation is computed. In the state of the art, the sizes of sub-images vary a lot. For images with good resolution such as $4000 \times 4000$, a $W$ of $41 \times 41$ can be chosen [244]. But these window sizes must not only be set regarding the size of the images but also depends on the type of the scene and how the disparity varies through the whole image. Nevertheless, it would be unreasonable to use a $W$ of $41 \times 41$ when the image size is $80 \times 60$.

In fact, we set 2 experiments, one on good resolution images in order to estimate the sole impact of $W$ on the accuracy of sub-pixel matching and one on our low-resolution images to prove the validity of our method according to the end application and to estimate the combined impact of the phase congruency threshold $\gamma$ and $W$ on the sub-pixel matching accuracy. We designed the following experiment to estimate the sub-pixel matching accuracy: for a specific thermal image $I_1$, we created a second image $I_2$ by shifting the information of $I_1$ by $\Delta$ pixels along the $x$-axis. The accuracy can be determined according to the distance between an estimated sub-pixel disparity to $\Delta$ (the true disparity). We called it *matching error*. We also created a *matching precision rate* measurement. For this, we set a precision threshold of $\tau$. Given an estimated disparity $\delta$ if $|\delta - \Delta| \leq \tau$ the match is considered as correct else as a mismatch. The *matching precision rate* is the ratio between the number of correct matches (relative to a level of precision $\tau$) to the total number of features.

For the good resolution thermal images, we used a $512 \times 512$ image of people coming from [245]. We shifted these images by $\Delta \in \{20, 20.125, 20.250, ..., 25\}$ pixels. The matching errors were computed on all the features for all the set of $\Delta$. Fig. 4.12 shows the box plot of the root-mean-square deviation (RMSD) of the matching errors as a function of the window size $W$, with $W \in \{7, 9, \ldots, 41\}$. The impact of the window size on the rate of good matching respects to a specific level of precision $\tau$ can be seen in Fig. 4.13. We drew this curves for 4 levels of precision: $\tau = \{0.05, 0.01, 0.025, 0.5\}$ pixels.

The low resolution ($80 \times 60$ pixels) images were acquired with our FLIR Lepton 2 camera system ([246]). For each image we shifted them by $\Delta \in \{0, 0.125, .., 30\}$pixels with a stride of $0.125$ pixels. The size of our sub-images was in the range $W \in \{7 \times 7; 9 \times 9; 11 \times 11\}$. The Figure 4.14 shows the impact of the window size $W$ and the phase congruency moments threshold $\gamma$ on the rate of good matching respects to a specific level of precision $\tau$, $\tau \in \{0.5, 0.25, 0.1, 0.05\}$ pixels in our case. The value of the phase congruency moments threshold $\gamma$ is directly related to the number of features extracted on the images (Table 4.1). However, we had to find a good tradeoff between the number and significance of the extracted features.

We studied the impact of the sub-images window size in which the phase correlation is estimated on the accuracy of the sub-pixel localization. As shown in Fig. 4.12, using a wide sub-images window allowed us to get better precision. From $7 \times 7$ to $29 \times 29$, the precision was increased progressively, but from $31 \times 31$, the gain in precision was no more very noticeable. The phase congruency magnitude represents the filtered information of the image, so using a wider window does not help necessary to get better precision. This result is also confirmed in Fig. 4.13 and 4.14 on the cases were low precision was sufficient. These Figures showed the percentage of correct matches at a certain precision using different sizes of window size. For low precision (errors below $\tau = 0.5$ or $\tau = 0.25$ pixels), the matching rate was increased with higher sub-images window size. This behavior was shared on both high and low-resolution images. The only difference between these 2 cases is that, for the matching of low-resolution images, we had to limit the sub-images window size to $13 \times 13$, which is already large compared to the $80 \times 60$ image size.

More surprisingly, by analyzing the three Figures accurately, it can be noticed that a window $9 \times 9$ offered better precision than an $11 \times 11$ one. This is even more visible on the high precision matching rates (errors below $\tau = 0.1$ or $\tau = 0.05$ pixels) curves (Fig. 4.13-c and d and 4.14-c and d). One explanation could be that the precision of the match did not only depend on the size of the window size but overall by the ratio between noise and relevant information in this window. In our framework, they are 2 processes that handle the image noise: 1) the phase congruency magnitude represents filtered information of the image and 2) a low pass filter the phase correlation. In this latter case, the filter bandwidth was directly set proportional to the sub-images window size ($U = 0.5W$ in (4.27)). It seemed like that for $W = 9 \times 9$, the ratio between the useful information (on low spatial frequency in the cross-power spectrum (4.27)) and noise (on the high spatial frequency) reached a local maxima. If this hypothesis is confirmed, more noise is included in the phase correlation for higher $W$, degrading so its shape.

The expected precision also had a direct impact on the matching rate. We can see in Fig. 4.13 that for the high-resolution image, almost all the features (more than 99%) are matched with a precision lower than 0.5 pixels regardless of the window size. This matching rate decreased to around $90\%$, $55\%$, and $33\%$ for respectively a precision lower than 0.25, 0.1, and 0.05 pixels. We can notice the same behavior for the low-resolution image with matching rates (in the case of the best phase congruency threshold) around $97\%$, $83\%$, $55\%$ and $34\%$ for respectively a precision lower than 0.5, 0.25, 0.1 and 0.05 pixels. Moreover, we noticed the same matching rate ($\approx 33\%$), whatever the resolution of the images when a high matching precision is expected.

The choice of the phase congruency moments threshold also had an impact on the matching rates. Because of the low-resolution of our images, we wanted to increase

the number of features using the lower phase congruency moments threshold. This was the case, as shown in Table 4.1. The number of extracted features increased from 275 to 1793 when we decreased the threshold of $\gamma$ from 0.3 (as recommended in [240]) to 0.01. Unfortunately, the matching rates for a specific precision, also decreased when we decreased the threshold, as we can see in Fig. 4.14. Decreasing the threshold brought less stable features to match. Depending on the application, a good tradeoff has to find between the number of matched features and their reliability.

To summarize, in our specific low resolution stereo thermal camera case, a sub-images window size window of $9 \times 9$ can be a good tradeoff between the accuracy and the phase correlation computation time. With this configuration, we were able to match around $97\%$ of the features with a precision of at least 0.5 pixels. If a higher sub-pixel matching accuracy is needed, this rate falls to $55\%$ or $34\%$ for a respectively a precision less than 0.1 or 0.05 pixels. In this case, an external outlier rejection process based on the image or the 3D scene content should be added during or after the matching [250].

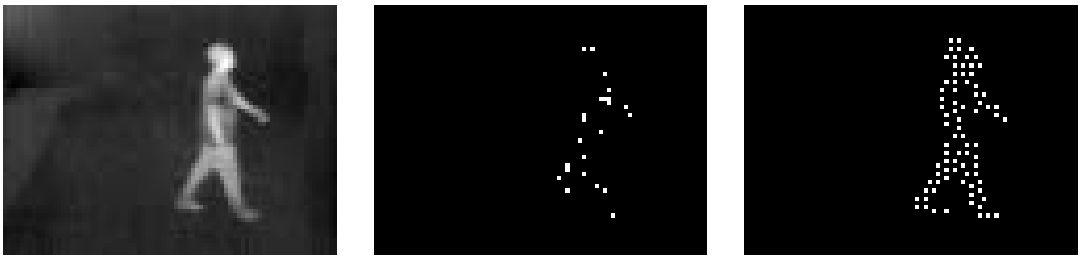
### 4.5.5 3D reconstruction

#### 4.5.5.1 Importance of sub-pixel matching

After matching, an error on the estimation of the disparity can have high consequences for the stereo reconstruction, especially to estimate the distance of a 3D point to the cameras (Z direction) [229] accurately. In this section, we wanted to estimate the range of error along the Z direction when we estimated the disparity with an error of one pixel (in pixel resolution) and also the evolution of this range using sub-pixel disparity estimation (error of 0.5 or 0.1 pixels). For this, we chose a pair of matched points, which gave a distance Z=3350 mm after stereo reconstruction for a disparity d. If we have an error of 1 pixel (a disparity of $d_{new} = d \pm 1$ pixel), the estimation of Z was spread from Z = 3108 mm to 3590 mm (a range of 482 mm). Being wrong in a range of 482 mm in a human body reconstruction can be severe for many surveillance applications. For an error of $d_{new} = d \pm 0.5$ the range was reduced to 270 mm and for $d_{new} = d \pm 0.1$ pixel to 52 mm. Such results explain why sub-pixel matching is essential in our low-resolution context.

#### 4.5.5.2 Evaluation of the whole framework

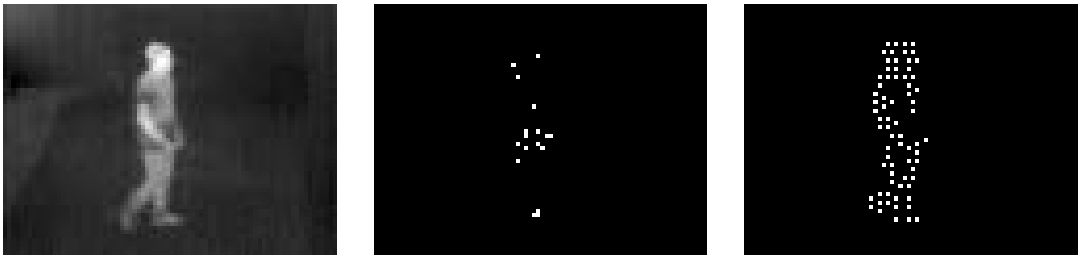
In this section, we tried to compare the whole framework of our method ST versus a method available in classical computer vision libraries. As a feature extractor, we chose ORB because this feature extractor has already been proved to be more robust than other [15]. For the matching, we used KNN because of the sparsity of the features and also because it estimates the disparity in sub-pixel precision. We implemented this framework (we called **ORB + KNN matching**) using the functions



(a) Left image of the 507 th pair of Tvvlgo

(b) Matches re-projected using ORB (#24)

(c) Matches re-projected using ST (#80)



(d) Left image of the 547 th pair of Tvvlgo

(e) Matches re-projected using ORB (#18)

(f) Matches re-projected using ST (#71)

**Figure 4.15:** 3D points projected in the images space with Z as the value of the pixel.

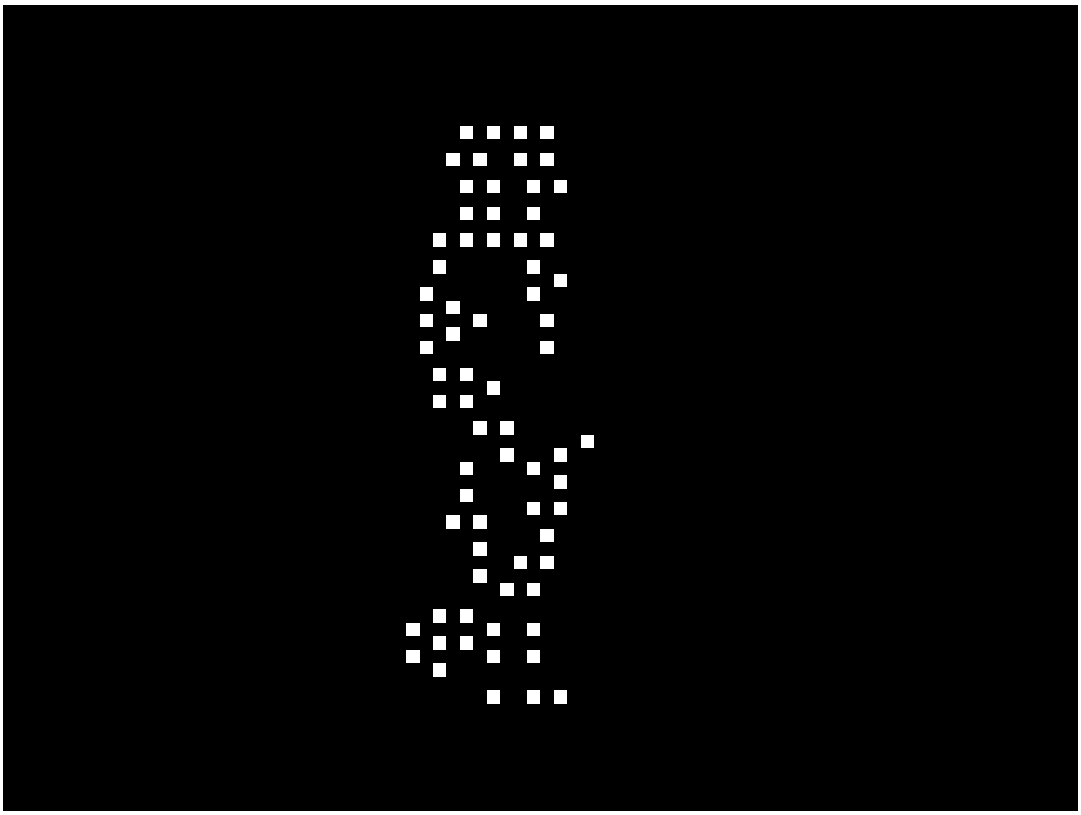


**Figure 4.16:** Box plot in the value of Z for our method ST and ORB + OpenCV KNN.

| | ORB + OpenCV KNN (Baseline) | ST (Our method) |
|---|---|---|
| 507 th pair | 1523.94 $\pm$ 1612.76 | 3648.10 $\pm$ 256.43 |
| 547 th pair | 1800.85 $\pm$ 1538.92 | 3746.56 $\pm$ 391.44 |

**Table 4.2:** Mean value and standard deviation of Z after triangulation. Values are in millimeters.

available in OpenCV. For ST, we set $\gamma = 0.1$ and for phase correlation we used a sub-image size of $9 \times 9$.

We applied these methods on two image pairs (Fig. 4.15a and Fig. 4.15d) we selected from the Tvvlgo dataset. In these images, the person was approximately at a distance of 3-4 meters. We compared the results of these methods according to two criteria: the numbers of matched features and the consistency of the 3D reconstruction, especially along the Z direction.

The Figure 4.15 represents the matches projected on the left images when using ORB + KNN (4.15b and 4.15e) and ST (4.15c and 4.15f). First of all, we can see that using ST, we have 4 $times$ more matches than ORB + KNN: 80 vs. 24 and 71 vs. 18 respectively for the 507 th pair and the 547 th pair. It can also be noticed that the matches are better distributed over the whole human shape than using ST than using ORB + KNN. This is due to the fact that phase congruency can extract more features from thermal images than ORB, as reported in Table 4.1.

For all the matches, we performed triangulation to estimate the 3D position of the point. We made statistics on the estimated depth Z. Figure 4.16 shows the boxplot of the distributions of Z for both methods. The mean and the standard deviation of these distributions are given in Table 4.2. The results obtained using ORB + KNN seems to be inconsistent (very wide distribution of Z in a range of about 5000 mm). On the contrary, for ST, the distribution is more compact. The median and mean values are around 3500 mm, which is consistent with the experimental conditions, and the standard deviations (256 mm and 391 mm) are more reliable concerning human body proportion. The results output by ORB + KNN can be explained by the fact that the matches are not correct, leading to very bad triangulation outputs.

## 4.6 Conclusion

This chapter proposed our method ST, a sub-pixel stereo matching method, adapted to thermal images. Thermal images have the disadvantages of being less textured compared to gray-scale or color images. Besides this lack of texture, low-cost

thermal cameras can have a very low resolution ($80 \times 60$ pixels in our case), which is detrimental to an accurate stereo reconstruction. To overcome these limitations, we proposed a framework composed of a robust feature extraction method based on phase congruency, a robust rough matching process based on Lades distance between the extracted features, and a refined sub-pixel matching process based on phase correlation. When applied to low-resolution thermal images, our feature extraction method was able to extract more features than state of the art methods. As well, our sub-pixel matching method was able to match around 97% of extracted features with an (average) error under 0.5 pixels. For about 55% of the features, the matching error was even below 0.1 pixels. Such a level of accuracy is necessary for the stereo reconstruction of a 3D scene or the 3D localization of objects. With our stereo setup, a precision bellow 0.1 pixels corresponds to a maximal error of $\approx 51$ mm in the depth direction. With our framework, such a precision level seems now achievable even for very low-resolution thermal stereo cameras. We also compared ST versus a classical method in state of the art (ORB + KNN) for 3D reconstruction. Our method showed more consistent results than ORB + KNN.

Once the sub-pixel matching is performed, it could be essential to compare our method in 3D localization using ground truth values. Such evaluation will show us how sub-pixel matching is improving 3D vision.

# 5 Super-resolution

> The difference between something good and something great is attention to detail.
>
> Charles R Swindoll

## 5.1 Introduction

Since the images we are using are characterized by low resolution and noise, we need to find a solution to enhance our images' quality. As we have concluded in the Chapter 4, pixel size (and so the image resolution) has a direct impact on the stereo-vision accuracy, mainly due to matching uncertainty. The main idea is to reduce the 3D uncertainty zone to be more accurate when performing stereo-vision (Fig 5.1). For this, we need images with a high resolution because when the resolution is high, the cost of a mismatch of 1 pixel around the correct match is less critical than for a lower resolution image. Besides, there is more information in high-resolution images, so extracted features should be more robust, and matching should be more accurate.

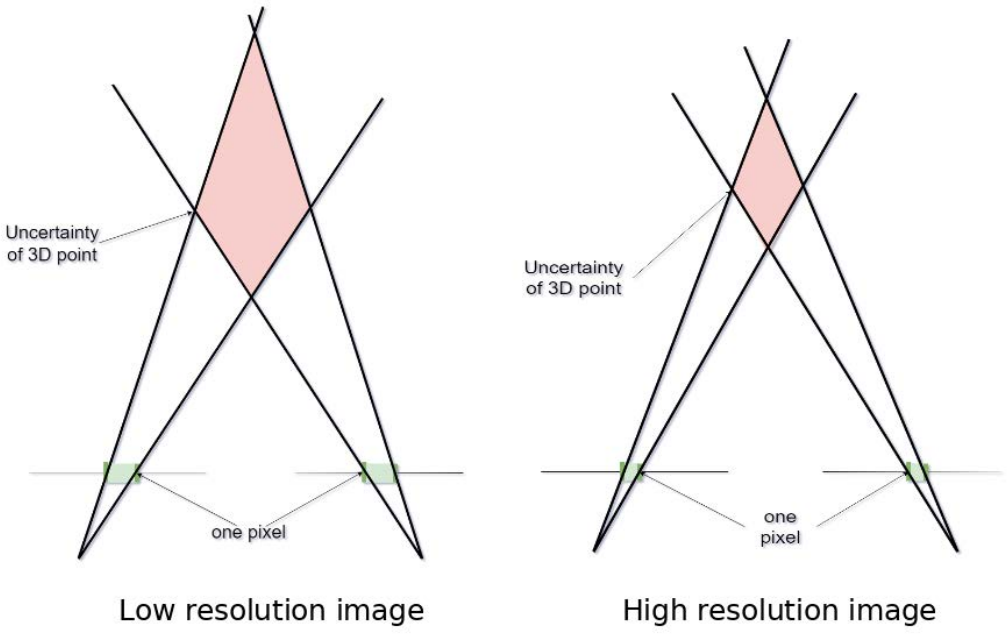



**Figure 5.1:** Importance of super-resolution for stereo-vision

This chapter's main idea is to find a way to estimate and obtain high-resolution images $I_{hr}$ from our low-resolution infrared camera images $I_{lr}$ in order to gain

3D location accuracy in our stereo-vision framework. This process is called super-resolution (SR). There are two main methods used in super-resolution: single-image SR (SISR) and multi-image SR (MISR). The SISR process uses a single image to reconstruct a high-resolution version of the image. However, a single image is quite limited in the amount of information it provides. In contrast, MISR uses several low-resolution images of the same scene, acquired from the same or different sensors, to construct an HR image. The advantage of MISR compared to SISR is that it allows additional information to be derived from the same scene observations. It, therefore, enables the construction of an image with high spatial resolution. However, MISR is confronted with many other problems, such as the multiplications of sensors, the registration between images, and the change of intensities over time. In our case of an infrared camera, we believe that only a SISR method can be used. We will focus on this type of method.

The chapter is structured as follows. The Section 5.2 discuss about the methods in the literature to determine $I_{hr}$ from $I_{lr}$. All these methods assume some degradation models. The Section 5.3 details our approach to perform super-resolution for thermal images. Section 5.4 presents qualitative and quantitative results and Section 5.5 concludes the chapter and presents some perspectives.

## 5.2 State of the art

Generally, there are three ways to perform super-resolution: interpolation based methods [251], model-based optimization methods [252, 253] and learning-based methods [254, 255].

### 5.2.1 Relation between high and low-resolution images: degradation model

The difference between an image of the same scene acquired with a low and a high resolution camera comes directly from the image formation process. Some authors have tried to model this image formation process [256]. Thus, starting from an irradiance $E_i(\cdot)$, the image formation process can be seen as the convolution of $E_i(\cdot)$ with the point spread function (PSF) of the camera. Then, the result of this convolution is sampled at discrete pixel locations (spatial decimation) (Fig 5.2). This process can be summarized as follows:

$$I_{lr}(m,n) = E_i \star \omega_i \star a_i(m,n) = E_i \star PSF_i(m,n) \tag{5.1}$$

Where $\omega_i$ represents the optical effects (lens and finite aperture), and $a_i$ represents the spatial integration performed by the CCD sensor.

Each camera has its own $PSF$. Let $PSF_{hr}$ and $PSF_{lr}$, the $PSF$s of respectively

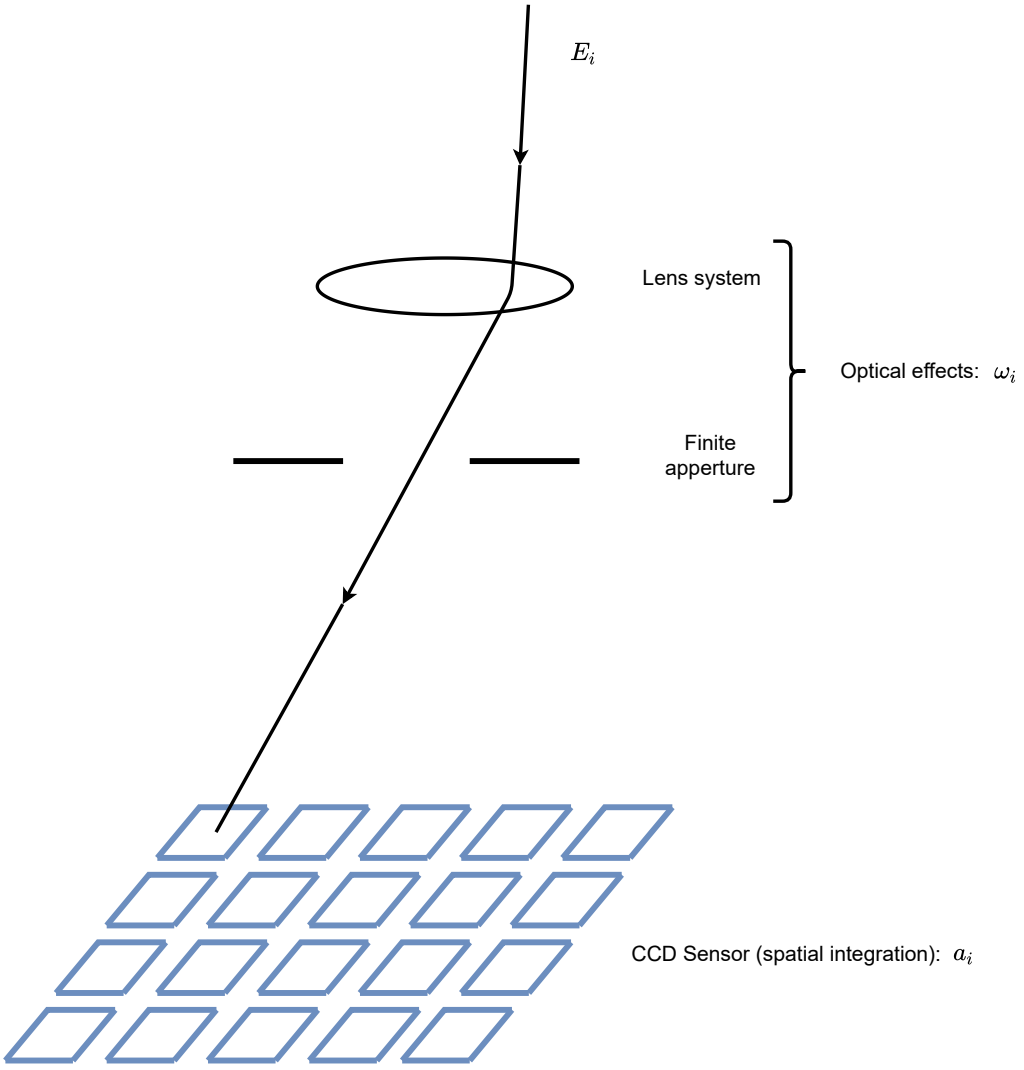



**Figure 5.2:** Image formation model from [256]

given high-resolution and low-resolution cameras. Rather than determining the relationship between $PSF_{hr}$ and $PSF_{lr}$, it is easier to determine the relationship between a high resolution image $I_{hr}$ and a low resolution image $I_{lr}$.

The correspondence between $I_{hr}$ and $I_{lr}$ can be expressed in many ways. One of the simplest model is done by averaging of the pixels of $I_{hr}$ within the sampling space of $I_{lr}$ [257]:

$$I_{lr}(m,n) = \frac{1}{q^2} \sum_{x=qm}^{(q+1)m-1} \sum_{y=qn}^{(q+1)n-1} I_{hr}(x,y) \tag{5.2}$$

where $q$ is the decimation factor. But most of the time, there are other factors such as blurring, warping and noise. In [258], Irani Michal and Peleg Shmuel propose a more complete model as follows:

$$\begin{aligned} I_{lr}(m,n) &= OI_{hr} + \sigma\,(m,n) \\ &= d\,(h\,(w\,(I_{hr}\,(x,y)))) + \sigma\,(m,n) \end{aligned} \tag{5.3}$$

where $O$ the observation matrix composed by the combination of a warping function $w$, a blurring function $h$ and a down-sampling operator $d$, and $\sigma$ is additive noise. This equation (degradation model) has been modified or simplified in many situations. There are multiple degradation models reported in [259], however, the basic model is Equation 5.3.

### 5.2.2 Interpolation-based methods

Most of the time, the interpolation-based methods have been used for image coding, image resizing, and image manipulation. Interpolation based methods can be divided into two categories: polynomial-based [260–266] and edge directed.

Let $f$ be a sampled function and we want to interpolate if to $g$ with $g(x_k) = f(x_k)$ with $x_k$ an interpolation node. If the data are equally spaced, the interpolation function can be written as:

$$g(x_k) = \sum_k c_k u\left(\frac{x - x_k}{h}\right) \tag{5.4}$$

Where $h$ is the sampling increment, $x_k$ the interpolation nodes, $u$ the interpolation kernel, and $c_k$ parameters depending on the sampled data.

#### 5.2.2.1 Polynomial-based

Polynomial filters have wide use in medical imaging (image rotations, isotropic volume recovering, etc.). The main idea behind the polynomial based method is to approximate the pixels values with a polynomial function. They can be divided into two categories: fixed and adaptive polynomial.

Concerning the fixed polynomials, we can classify the methods by increasing the degree of polynomials [261], we can find bilinear, cubic (Cubic Interpolation [262], Cubic B-Splines [260]), and higher-order polynomials [261]. They show that high degree B-spline gives better quality at the expense of the computation time. In [267], Unser et al. propose a fast version of the B-spline method by computing filters recursively. One of the limits of their approach is that the contrast of the interpolated image is not good. Besides, B-spline interpolation outputs sometimes block or mosaic effects.

The authors propose some adaptive polynomials. In [263], De Natale et al. propose an adaptive bilinear interpolation. They apply their method to image compression. In [264], Seong WonLee and Joon Ki Paik propose a new image interpolation method based on B-splines to exceed these limits. Their strategy is based on an adaptive zero-order interpolation and an adaptive four directional moving average.

The main advantage of polynomial-based interpolation is that they are fast. But they struggle to work well when there are some discontinuities in the image, such as edges. Moreover, the results are most of the time characterized by blurring or artifacts.

#### 5.2.2.2 Edge-directed

The main idea behind these methods is to enhance edges in the output image. Some ways try to estimate the edge orientation during the interpolation [268–271]. There are many types: explicit interpolation, the fusion of edge orientations, new edge-directed interpolation, soft-decision interpolation, and robust soft-decision interpolation using weighted least squares.

### 5.2.3 Model-based optimization methods

Most of the time, these methods are based on a minimization of a least-square cost function as follows:

$$\hat{I}_{hr} = \arg\min_{I_{hr}} \phi(I_{hr}, I_{lr}) = \arg\min_{I_{hr}} \|I_{lr} - OI_{hr}\|^2 \tag{5.5}$$

where $\phi(I_{hr}, I_{lr})$ is the fidelity term and $O$ the observation matrix (Equation 5.3). This least square problem can be solved using the generalized inverse (or pseudo-

inverse) as follows:

$$\hat{I}_{hr} = (O^t O)^{-1} O^t I_{lr} \tag{5.6}$$

In [252], Irani and Peleg propose an iterative back-projection (IBP) method by computing a residual between a simulated low resolution image with the real low resolution image. They algorithm is based on the image acquisition where the resulting low resolution is the combination of many high resolution images. From Equation 5.3, the relationship between $(x, y)$ and $(m, n)$ using k frames is as follows:

$$\begin{aligned} x &= x_k^0 + s_x m \cos\theta_k - s_y n \sin\theta_k \\ y &= y_k^0 + s_x m \sin\theta_k + s_y n \cos\theta_k \end{aligned} \tag{5.7}$$

where $\theta$ is the rotation of the $k$th frame, $s_x$ and $s_y$ are the sampling rates in the $x$ and $y$ directions. By using Taylor expansion on two low resolution image, they apply an iterative refinement. Given a sequence of low resolution images:

1. First, they initially assume that there is no motion between frames as follows:

$$\hat{I}_{hr}^0 = upscale(I_{lr}) \tag{5.8}$$

2. Then, they approximate the motion by solving a the relationship between two consecutive frames. At $t$th iteration we have:

$$\hat{I}_{hr}^{t+1} = \hat{I}_{hr}^t + upscale(O\hat{I}_{hr}^t - I_{lr}) \tag{5.9}$$

3. Finally, they wrap $I'_{lr}$ toward $I_{lr}$ using the currently estimated motion parameters and go to step two.

While their method is better than some interpolation methods for recovering edges, it is sensitive to noise.

There are many improvements of the Equation 5.5 by adding a regularization term $\mathcal{R}$ [272–274] as follows:

$$\hat{I}_{hr} = \arg\min_{I_{hr}} \phi(I_{hr}, I_{lr}) = \arg\min_{I_{hr}} \|I_{lr} - OI_{hr}\|^2 + \lambda\mathcal{R}(I_{hr}) \tag{5.10}$$

Where $\lambda$ is a trade-off parameter.

Choosing $\mathcal{R}$ depends on a given prior and some assumption on the image data distribution. So different regularizer will provide different high-resolution images.

### 5.2.4 Learning-based methods

Learning-based methods are a vast set of techniques: features pyramid, projection, neural networks, manifold, and compressive sensing.

#### 5.2.4.1 Features pyramid

The methods have been widely used for face images super-resolution. Given a high-resolution image $I_{hr}$, it is first down-sampled and blurred, producing the Gaussian resolution pyramid. Gaussian pyramids are then used to generate Laplacian pyramids and finally to features pyramids. The system is then trained to select given a low-resolution patch, a high-resolution patch which is the most similar. This similarity can be defined in terms of nearest neighbors [275] or using a tree search [276].

#### 5.2.4.2 Projection

Projection-based methods are divided into three categories: Principal Components Analysis (PCA), Independent Components Analysis (ICA), and Morphological Components Analysis (MCA). The main idea behind these methods is to reduce the dimension of $I$. $I$ is converted to $\vec{I}$, the lexicographically ordered vector of all pixels of $I$.

In [277], Capel David and Zisserman Andrew propose a super-resolution method using multiple views. They apply their method (based on PCA) to text and face images. A given image $I$ can be modeled using PCA components as follows:

$$f = Vy + \mu \tag{5.11}$$

Where $V$ represents the set of principal components basics vectors and $\mu$ the average of the training image, the dimension of $y$ is supposed to smaller than the $I$ dimension. Instead of applying PCA on the whole image, they use it to small regions such as the eyes, nose, mouth, and cheek areas. The main drawback of the PCA-based super-resolution method is that they are sensitive to occlusions.

In [278], Liu et al. propose an independent components analysis-based method for face super-resolution. First, they use PCA to approximate high-resolution images of ICA components. This is done offline. Then, they estimate ICA components coefficients of a given high-resolution image given a low-resolution using the MAP algorithm. Finally, they compute an intermediate result, which will be used to compute a structure tensor representing the relationship between the low-resolution image and the high-resolution image.

In [279], Liang et al. address image super-resolution using morphological component analysis. The method is composed of three steps. First, the given low resolution $I_{lr}$ is up-sampled to $I_{zoom}$ using bilinear interpolation. Secondly, they apply

MCA to $I_{zoom}$ to output $I_{hr}^{g}$ an intermediate high-resolution image and $I_{s}^{g}$ unsharp masking. Then, They apply residue compensation. Another intermediate image $I_{hr}^{l}$ is constructed by downsampling $I_{hr}^{g}$, combining the result to $I_{lr}$ and performing neighbor reconstruction to output $I_{hr}^{l}$, which has the same dimension as $I_{hr}^{g}$. To get the final estimated high resolution image add $I_{hr}^{g}$ to $I_{hr}^{l}$.

#### 5.2.4.3 Manifold

Like PCA or ICA, these set of methods performs super-resolution through dimension reduction. They assume that the manifold of $I_{hr}$ and $I_{lr}$ are similar geometries. They are three manifold learning methods that have been used in the literature: Local Linear Embedding (LLE), Local Preserving Projection (LPP), and Orthogonal Locality Preserving Projections (OLPP).

In [280], Hui Zhuo and Lam Kin-Man propose a super-resolution method based on Local Linear Embedding (LLE). Their approach is composed of two stages algorithm: reconstruction using Eigen-transformation and patch-based local structure refinement. The main drawback of LLE is that they are down-performing for low-resolution images present in the training data.

In [281], Park Sung Won and Savvides Marios propose a super-resolution method based on local preserving projection. LPP is supposed to preserve local structure and image data space structure. LPP is non-linear, while LLE is non-linear. The LPP algorithm is composed of three steps:

— First, they construct an adjacency graph where each training patch is a node, and each node has K nearest neighbors. In their work, they evaluate the influence of the patch size on the reconstructed image quality. They find that small patches produce blurry images, while big patches produce images with some artifacts. They set the size of patches to $24 \times 24$.

— Secondly, they estimate the weights linking a node to its neighbors. They compute weight between two neighbors $x_i$ and $x_j$ by computing the Gaussian kernel of the Eucliedian distance. If $x_i$ and $x_j$ are set as connected, their weight $w_{ij} = 1$ else $w_{ij} = 0$.

— Finally, they compute eigenmaps. The eigenvectors and eigenvalues are computed as follows:

$$XLX^{t}a = \lambda XDX^{t}a \tag{5.12}$$

where $D$ is a diagonal matrix with $D_{ii} = \sum_{j} w_{ij}$, $L = D - W$, and the projection matrix A with eigenvectors $a_i$ as column vectors.

In [282], Kumar BG Vijay, and Aravind Rangarajan propose a super-resolution

method based on Orthogonal Locality Preserving Projections. The OLPP algorithm is divided into three steps.

— First, like in LPP, they construct the adjacency graph where each node $i$ represents an image $x_i$. The distance between nodes is define in terms of K nearest neighbors. Like in [281], the patches size is $24 \times 24$.

— The next step is the weights computation. If $x_i$ and $x_j$ are connected then $w_{ij} = \exp^{-\left(\|x_i - x_j\|^2/t\right)}$ where $t \in \mathbb{R}$ else $w_{ij} = 0$.

— Finally, they compute the orthogonal basis functions. Like for LPP, $D$ is defined as $D_{ii} = \sum_j w_{ij}$ and $L = D - W$. They compute the first orthogonal basis vector $a_1$ as the eigenvector of $(XDX^t)^{-1}XLX^t$ and the $k$th eigenvector $a_k$ of:

$$M_k = \left\{I - (XDX^t)^{-1}A_{k-1}B_{k-1}^{-1}A_{k-1}^t\right\}(XDX^t)^{-1}(XLX^t) \tag{5.13}$$

where $A_{k-1} = [a_1, \cdots, a_{k-1}]$ and $B_{k-1} = A_{k-1}^t(XDX^t)^{-1}A_{k-1}$.

#### 5.2.4.4 Compressive sensing

Compressive sensing is a technique that has been introduced in 2006. It is based on the sparsity of the signal to reconstruct and incoherence applied through the isometric property. Most of the time, for super-resolution two over-complete dictionaries ( $\mathbf{D}_{hr}$ for high resolution images and $\mathbf{D}_{lr}$ for low resolution image) are built. A given dictionary contains $K$ prototype image patches and a high resolution image $I_{hr}$ is the linear combination of these patches: $I_{hr} = \mathbf{D}_{hr}\alpha$ with $\alpha << K$. The coefficients $\alpha$ are determined by minimizing the term:

$$\lambda \|\alpha\|_1 + \frac{1}{2}\left\| \begin{bmatrix} F\mathbf{D}_{lr} \\ \beta P\mathbf{D}_{hr} \end{bmatrix} \alpha - \begin{bmatrix} FI_{lr} \\ \beta w \end{bmatrix} \right\|_2^2 \tag{5.14}$$

where :

— F extracts features from $\mathbf{D}_{lr}$ patches

— P extracts the overlapping area between two successive reconstructed images

— $\beta$ is a trade-off measure

Most of the time, compressive sensing is used with another method MAP, IBP, support vector regression, and Wavelet-based.

#### 5.2.4.5 Neural networks

As in many other fields, deep learning-based methods (a sub-set of learning-based methods) have achieved better results than other methods. The main idea is to learn a set of weights by computing the error between $I_{lr}$ and $I_{hr}$ and back-propagating the error to update weights. They can be divided into nine categories based on each method's main contribution and the network architecture:

**Linear networks:** These networks are basically designed without skip connections or multiple-branches. There are two philosophies: early up-sampling and late up-sampling. For early up-sampling, the low-resolution image is resized to the high dimension size, and then the details are recovered during learning. For late up-sampling, the learned features maps are resized, most of the time, in the last layer of the neural network.

Regarding early up-sampling, in [283], Dong et al. are the first to propose a convolutional neural network to perform super-resolution. Their network, SRCNN, is composed of 3 modules: *features extraction and representation*, *non-linear mapping* and *reconstruction*. The authors first up-sample low-resolution images using bi-cubic interpolation. In [284], Kim et al. propose VDSR, which is a very deep network inspired by VGGNet [285]. The difficulty of training is by-passed by global residual learning and gradients clipping. Handling multiple scales helps their model to generalize better and gives better results than a single scale model. In [286], Zhang et al. propose a super-resolution network called DnCNN based on SRCNN. They also apply batch normalization and ReLU layer. In [287], Zhang et al. propose IRCNN, a neural network performing denoising, deblurring, and super-resolution. The structure is composed of 7 dilated convolution layers with batch normalization.

Other networks perform up-sampling in the late stage of the neural network. In [288], Dong et al. propose a method FSRCNN, based on SRCNN, which is faster. FSRCNN is composed of four layers, namely features extraction, shrinking, non-linear mapping, and expansion. FSRCNN features extraction layer use different input size and filter size compared to SRCNN. The second layer reduces the dimension of the feature dimension to reduce the computational load. The non-linear mapping layer is the same in SRCNN and FSRCNN, while the fourth layer is a deconvolution layer. In [289], Shi et al. propose a deeper model called ESPCN. They use a sub-pixel convolution layer to up-sample feature maps.

**Residual networks:** While VDSR and ESPCN can face vanishing gradients, some authors propose residual learning, making deeper models vanishing gradient-free. In [290], Lim et al. propose a super-resolution method, called EDSR, based on ResNet architecture [291]. They remove batch normalization and ReLU activation outside of

each residual block. To enhance their network, they also perform multi-scale learning for the same network. In [292], Ahn et al. also apply ResNet bloc for super-resolution. Their neural architecture is composed of cascaded modules merged using $1 \times 1$ convolutional layers.

In [293], Jiao et al. propose a model on top of DnCNN [286]. FormResNet is composed of two sub-networks, similar to DnCNN. The first DnCNN is called *formatting layer*, which incorporates Euclidean and perceptual loss. The second DnCNN is called *DiffResNet* which input is the output of the *formatting layer*. In [294], Fan et al. also propose a two-stage architecture. Their network is composed of 2 residual sub-networks. The first sub-network involves a low-resolution stage with six residual blocks, while the second sub-network involves the high-resolution stage with four residual blocks. The features maps output by the low-resolution sub-network is up-sampled before being fed to the high-resolution sub-network. In [295], Mao et al. propose a super-resolution method based on UNet [296]. Their network is an auto-encoder composed of symmetric convolutional and deconvolutional layers with ReLU after each layer. The low-resolution images are first up-sampled using bicubic interpolation. Thus the input and the output of the neural network have the same dimension.

**Recursive networks:** The main idea behind recursive networks is to apply many times the same operations on the features maps. Such a strategy reduces computational while breaking down the super-resolution problem to simpler problems. In [297], Kim et al. propose a recursive network applying the same convolution multiple times. The network is deep, while the number of parameters does not increase. Their network is composed of two sub-networks: *embedding network* and *inference network*. The *embedding network* extract feature maps from low-resolution input image while the *inference network* recursively apply the bloc (convolution + Relu) layer to up-scale the features maps. In [298], Tai et al. propose a deep recursive residual network (DRRN) composed of 52 convolutional neural networks. Even if their network is deeper than VDSR, REDNet, and DRCN, their network complexity is lower. They combine residual learning with local identity connections between small blocks. To avoid vanishing and exploding gradient, they apply gradient clipping. In [299], Tail et al. propose a deeper model with 80 layers. Their model is composed of three parts, each one similar to SRCNN. The first module extracts features from a bicubic up-sampled low-resolution image. The second part is composed of a series of memory blocks (six) stacked together. Each memory block is composed of a recursive unit and a gate unit. The last module reconstructs the high resolution residual by aggregating features. In [300], Li et al. propose a network SRFBN composed of feedback blocks containing projection groups. For training, Li et al. use the curriculum learning approach. The training begins with simple images, and the complexity of the high-resolution images

increases more and more during training.

**Progressive reconstruction designs:** Rather than applying recursion, other authors propose to perform super-resolution hierarchically. Most of the time, the network is composed of many sub-networks that successively up-scale the low-resolution image by 2.

For example, in [301], Wang et al. propose a progressive reconstruction neural network based on sparse coding. They use the first convolutional layer to extract features. Then, these features are fed into a Learned Iterative Shrinkage and Thresholding Algorithm (LISTA) network.

In [302], Lai et al. propose LapSRN, a super-resolution neural network that gradually up-scales the image by a factor of 2. LapSRN is composed of three sub-networks. The first one up-scales the low-resolution image by two and so on. Each sub-network has a Charbonnier loss, and each layer is formed of a convolution/deconvolution followed by a leaky ReLU. They also extend LapSRN to a multi-scale version.

**Densely connected networks:** This class of methods is based on DenseNet [20]. DenseNet has been successfully used for image classification.

In [303], Tong et al. propose a network (SRDenseNet), which is heavily based on DenseNet. A given layer receives the output of all previous layers concatenated. Tong et al. propose three variants of SRDenseNet, which have both a first convolutional layer that extracts low-level features. Each variant is indeed composed of four sub-modules: *features extraction layer* extracting low level features, a *dense block series* (8 height dense blocks) which extract high level features, *two deconvolution layers* and finally a *reconstruction layer*. In the first variant, the model is composed as the succession of the four sub-modules. In this version, residual learning only concerns the layers contained in the dense blocks, and only high-level features are used for the reconstruction. The second variant adds a skip connection by concatenating the output of the *features extraction layer* with the output of the *dense block series*. In the third variant, the low-level features are combined with each dense block input. The last variant gives the best performance.

Influenced by this model, in [18], Zhang et al. propose a Residual Dense Network with residual connections at a high and local level. At the local level, residual dense block input is the combination of the previous residual block outputs. In each residual block, there is a local feature fusion. At a global level, the residual dense blocks outputs are fused, and global residual learning is performed.

In [304], Haris et al. propose a neural network based on iterative up and down-sampling. Their model is composed of three modules: *features extraction*, *back-projection stages* and *reconstruction*. The features extraction phase is composed of a $3 \times$

3 convolution followed by a $1 \times 1$. Then, the *back-projection stages* is composed of a first up-projection sub-module followed by pairs of (down-projection + up-projection) modules.

**Multi-branch designs:** This class of methods uses multi-path signal flow to enhance the performance. In [305], Ren et al. propose CNF, a multi-path neural network composed of multiple blocks of SRCNN with a different number of layers. The SRCNNs are set in parallel, and their outputs are fused using sum-pooling. In [306], Hu et al. propose CMSC a cascaded multi-scale cross-network. Their model is composed of three modules: *features extraction*, *cascaded sub-networks* and a *reconstruction* module. Each sub-network is composed of two branches of convolutions with some combinations.

In [307], Hui et al. propose Information Distillation Network (IDN) composed of three modules: *feature extraction*, *information distillation* and *reconstruction*. *Feature extraction* module contains 2 ($3 \times 3$ convolution + ReLU). The *information distillation* module is compose of a succession of distillation blocks. Each block is composed of *enhancement* and *compression*. The enhancement unit is composed of $3 \times 3$ convolutions. The input of the enhancement unit is concatenated with a slice of the output of the third convolution and add to the output of the last convolution. The compression unit is composed of a $1 \times 1$ convolution. Finally, the reconstruction module is composed of a $17 \times 17$ deconvolution layer.

In [308], Qiu et al. propose Embedded Block Residual Network (EBRN) which is composed of three modules: *feature extraction*, *embedded block residual learning* and *reconstruction*. The *feature extraction* module is compose of three convolutions layers. The *embedded block residual learning* is composed of Blocks Residual Module (BRM).

**Attention-based networks:** While previous models consider equally all features channels; some authors integrate attention mechanisms in their super-resolution models.

In [309], Choi Jae-Seok and Kim Munchurl propose SelNet composed of 22 convolutional layers containing each a selection unit. Their model uses residual learning and a modified version of gradient clipping.

In [310], Zhang et al. propose Residual Channel Attention Network (RCAN), a very deep learning model with 400 layers. The network is composed of Residual blocks in Residual blocks (RIR) with long skip connections (LSC). There are a series of residual blocks in each residual group similar to those in [18] with short skip connection (SSC). In each residual block, there is a channel attention mechanism.

In [311], Anwar Saeed and Barnes Nick propose densely residual Laplacian attention Network(DRLN) a neural network composed of four modules: *feature extraction*, *cascading over residual on the residual*, *up-sampling* and *reconstruction*. Each

residual block contains three residual units. Cascading connections are either long skip, medium skip or local skip.

**Multiple degradation handling networks:** Most of the time, the authors assume a bicubic degradation through a simplification of the equation 5.3 as follows:

$$I_{lr}(m,n) = d\left(I_{hr}\left(x,y\right)\right) \tag{5.15}$$

Unfortunately, such simplification does no represent real-world situations. In [179], Zhang et al. propose to incorporate the information about the blur kernel and the noise with the low-resolution image before feeding the network. They first vectorize a Gaussian kernel, then they apply PCA on this vector, and finally, they stretch with the noise level to produce degradation maps. To use their model, one must first know the blur kernel and the noise level. Their network is composed of 12 convolutional layers with batch normalization and ReLU.

**Generative adversarial networks models:** As in the synthetic image, Generative Adversarial Networks (GAN) have also been used for super-resolution. We will go more in-depth about GANs in Chapter 7. Here, we will depict the most valuable contributions of GANs in super-resolution. GAns are composed of two networks, a generator, and a discriminator, that are playing against each other. The generator's role is to create fake images that should be as possible as similar to real images, while the discriminator's role is to distinguish real images from fake images.

In [312], Ledig et al. propose SRGAN, a GAN-based super-resolution with perceptual loss. The generator is composed of a *feature extractor block*, a *series of residual blocks* and a *reconstruction block*. Each residual block comprises a convolution, batch normalization, PReLU, convolution, batch normalization, and element-wise sum. There are skip connections between residual blocks. The discriminator is mainly based on the VGG network. The perceptual loss, they use, is composed of content loss and adversarial loss. In [313], Sajjadi et al. propose EnhanceNet by combining three loss functions: adversarial loss, perceptual loss, and texture matching loss. In [314], Wang et al. propose a neural network based on SRGAN. They remove batch normalization and incorporate dense blocks with the residual connection. They also add a global residual connection and train the GAN using Relativistic loss [315].

**Special cases of thermal images:** While many published works involve visible (RGB) images, the thermal images have received less attention. In [16], Choi et al. propose TEN, a thermal enhancement network inspired by SRCNN with three convolutional

layers. In [316], Han et al. use two inputs for their networks: a low-resolution near-infrared and a high-resolution visible image. In [17], Bhattacharya et al. proposed a model CNN with skipped connections. Inspired by VDSR, in [317], He et al. proposed a two convolutional neural network using 20 + 10 layers. The low-resolution image is gradually upscale from 1 to 2 and then to 8.

## 5.3 Approach

Our image characteristics have to be taken into account when dealing with low-cost thermal cameras: 1) they suffer for low signal-to-noise ratio (SNR) so even high-resolution images are more affected by noise than their visible counterparts; 2) thermal images are highly texture-less so it remains difficult to extract and propagate some information from low to high resolution; 3) they undergo some image acquisition effects such as infrared halo effects and history effects. So two thermal images taken by the same sensor can be very different.

Another serious difficulty is that high-resolution thermal imaging cameras are expensive and, therefore, not widely available. It is, thus, difficult to create learning sets from low/high-resolution real image pairs.

For all these reasons, we choose first of all to use and **adapt a CNN network based on residual dense block** (RDN [18] which proved to give high-quality super-resolution images in the RGB case).

We also focused on solutions to handle specific information relevant to our stereo-vision process. We noticed in our images that edges were often less sensitive to temperature variation or to measure errors. One of the challenges of super-resolution is to reconstruct salient edges. So, we choose to integrate a mean **to focus on the edges in the CNN**.

Another problem is the learning dataset. Because it was not possible to access some real low/high-resolution image pairs, we decided to create or own dataset by creating low-resolution images by applying degradation models to real high-resolution images. To be more generic, we choose to learn **a blind model for thermal images**. For each image we have generated synthetic low resolution images by randomly selecting a degradation model values to have both enough training images and enough generalization. These values are selected from $\{\eta_{min}, \eta_{max}\} \times \{\sigma^2_{min}, \sigma^2_{max}\}$.

Therefore, we propose a blind model, the Edge Focused Thermal Super-resolution (EFTS), to perform single image super-resolution for thermal images. In this section, we will first discuss the choice of a blind model, then we will present the network we used, with some emphasis on the edge extraction module, which will bring some edge saliency to the network.

### 5.3.1 Blind model

In most of the works done on super-resolution, authors use some degradation models in their process. Mostly, they used bicubic degradation models. In our case, the learning set will be built by applying a degradation model to high-resolution images to learn the relationship between $I_{hr}$ and $I_{lr}$. So, the model must be more sophisticated than a simple bicubic one.

The basis model described in Equation 5.3 seems to be generic enough to handle or mimic the real degradation provided by a low-resolution thermal camera.

The blur kernel $\eta$ has a significant impact on the degradation or reconstruction of the image. It is one of the effects of the PSF of the camera. The most common blur kernel is isotropic Gaussian with kernel size and a standard deviation [318]. In certain conditions, it is also possible to consider an anisotropic blur kernel where $x$ and $y$ standard deviations are different [319]. When there is long exposure time, in [320], the authors use more complex blur models such as motion blurring. The estimation of the blur kernel is essential for the reconstruction. If the estimated blur kernel is smaller than the ground truth blur kernel, reconstructed images are smooth, and if it is bigger than the ground truth blur kernel, there are some artifacts in the reconstructed images.

The noise term, $\sigma(m, n)$, is also crucial since low-resolution thermal cameras are more sensitive to noise than higher resolution ones. Infrared image drawbacks such as halo and noise are highlighted. Traditionally, noise is assumed to be Gaussian. To deal with noise, there are many solutions. One way is first to perform denoising and then implement super-resolution. But the denoising process causes the loss of some image information. It is also possible to perform super-resolution before denoising, but this process becomes computationally costly given that it is completed with a high-resolution image. To deal with these drawbacks, denoising and super-resolution can be performed jointly. But here also, the estimation of the noise model is essential for the reconstruction.

The question is then how to determine the model's parameters to be as close as possible to our real situation. In [179], to determine the blur kernel in real-world applications, authors used a visual quality assessment. For real-time applications, this is quite complicated. Moreover, if human eyes can compare two visible reconstructed images' quality, such a task will be more complicated for thermal images. Even high-resolution thermal images contain noise, blurring, and artifacts. One could anticipate that the degradation model can be learned only one time for a given thermal camera. But depending on the luminosity, the heat, the distance between the cameras and objects, the image's quality can be up- or downgraded.

It is difficult to determine the degradation model for all these reasons given that

such degradation model is not fixed. This is why we are using a blind model for our network. Rather than performing SISR for a specific type of degradation model values (one blur kernel $\eta$ and one noise $\sigma$), we want our network to be as general as possible. This is why for each image, we have generated synthetic low-resolution images by randomly selecting a degradation model values to have both enough training images and enough generalization.

### 5.3.2 Proposed network

As stated before, our solution's main idea is to start from an architecture that has proven to be efficient for RGB images super-resolution, to adapt it to the specificity of thermal images, and add a module that will promote the process of edges within the network. So, we have integrated an edge extraction module at the beginning of the network.

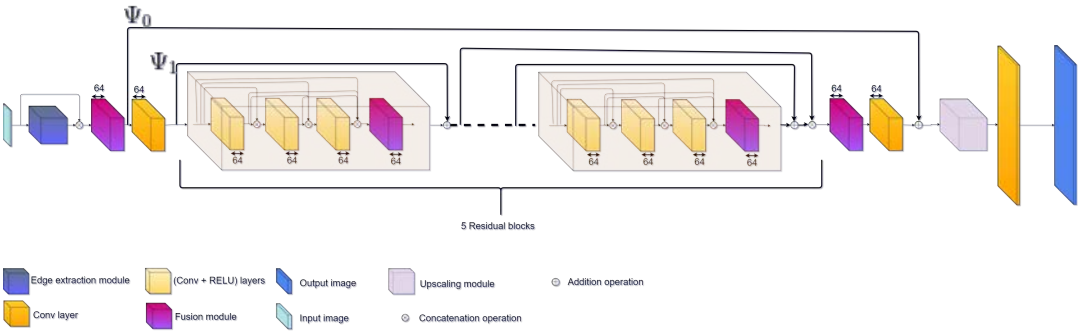



**Figure 5.3:** Edge Focused Thermal Super-resolution (EFTS)

Our approach is based on two principles: enhance the image while keeping edge saliency.

The proposed EFTS model (Fig. 5.3) is composed of four modules: 1) edge extraction module (EEM) (Fig. 5.4), 2) shallow feature extractor (SFE), 3) non-linear mapping module (NMM) and finally 4) an upscaling module (UM).

#### 5.3.2.1 Edge extraction module

Image edge detection is one of the basic fields in image processing. To detect edges in an image $I$, a kernel is generally convoluted with this image. There are three types of edge operators: classical, Zero crossing (Laplacian of Gaussian), Gaussian, and colored edge detectors.

In [321], the authors highlight the advantages and disadvantages of each type of edge detector. The Laplacian of Gaussian (LoG) is a well-known operator that can find correct edges, but it does not work well at corners and edges. The main advantage of LoG is that it is noise-tolerant, given that a Gaussian kernel first blurs the image. There are classical operators such as Sobel [322], Prewitt [323] and Kirsch [324]. If

these methods are simple and can detect edges in many orientations, they would be sensitive to noise. Zero crossing operator such as Laplacian [325] can detect edges and their orientations using fixed characteristics in all directions. On the other hand, Gaussian and colored edge detectors are complex and time-consuming.

The edge extraction module's primary objective is not to denoise the input image but rather to represent the noise and blurring effect. So, the network would receive additional information as input. Edges have a crucial factor in a thermal image, and these edges have an essential impact on segmentation [226].

As mentioned earlier, all the operators respond differently to noise and blurring and can represent many ways to see the same scene. This difference can bring more information to our network. We decide to combine this information in our module (see Fig 5.3) to increase the probability of highlighting the edges in the network. We have tried the following operators: Prewitt, Sobel, Laplacian, Kirsch, and their combinations (Sobel-Kirsch-Laplacian, Sobel-Kirsch-Prewitt, and Kirsch-Prewitt-Laplacian).

The EEM module takes the original low-resolution degraded image as input and outputs $\Upsilon$.

$$\begin{aligned} \Upsilon &= EEM(I_{dlr}) \\ &= \Gamma_1 \otimes \Gamma_2 \otimes \cdots \otimes \Gamma_n \end{aligned} \tag{5.16}$$

where $\Gamma_i$ is the $i$th edge extractor, $n$ the number of edge extractors and $\otimes$ is the concatenation operator.

Then $F_{EM}$ is concatenated with the original low resolution degraded image $I_{dlr}$. So we have:

$$\Lambda = I_{dlr} \otimes \Upsilon \tag{5.17}$$

#### 5.3.2.2 Shallow feature extractor module

For shallow features extraction, we use one convolutional layer as proposed by [290]. This is also a difference in our model compared to RDN. Given that $F'$ is composed of very different information we first fuse them in a $1 \times 1$ convolutional neural layer. So, we have:

$$\begin{aligned} \Psi_0 &= Fu\left[\Lambda\right] \\ \Psi_1 &= SF\left[\Psi_0\right] \end{aligned} \tag{5.18}$$

where $Fu$ is a $1 \times 1$ convolutional layer and $SF$ a $3 \times 3$ convolution layer.

#### 5.3.2.3 Non-linear mapping

The non-linear mapping module allows learning the non-linear mapping between $I_{dlr}$ and $I_{hr}$.

This part is inspired by RDN [18]. Their dense residual network can extract hierarchical features through contiguous memory, local feature fusion, local residual learning, and global residual learning. Their model is also based on DenseNet [326] and MemNet [327].

The global residual learning induces:

$$\zeta = \Psi_0 + \Lambda \tag{5.19}$$

where

$$U = DFu\left(\Phi_1 \otimes \Phi_2 \otimes \cdots \otimes \Phi_D\right) \tag{5.20}$$

Where $DFu$ expresses dense feature fusion composed of a $1 \times 1$ convolutional layer followed by a $3 \times 3$ convolutional layer, $\Phi_i$ the output of the $i$th residual block and $D$ the number of residual blocks. The output of the $i$th residual block $(\Omega_i)$ is defined as follows:

$$\Phi_i = \Omega_i\left(\Omega_{i-1}\left(\ldots \Omega_1\left(\Psi_1\right)\ldots\right)\right) \tag{5.21}$$

where $\Omega_i = \Omega_{i-1} + B_i$. $B_i$ is the $i$th block containing $C$ (Convolutional + ReLU) layers followed by a $1 \times 1$ convolutional layer. The output of $B_i$ is defined as follows:

$$B_i = Fu_i(FC_{i,0} \otimes FC_{i,1} \cdots \otimes FC_{i,N}) \tag{5.22}$$

where $Fu_i$ is a $1 \times 1$ convolutional layer and

$$FC_{i,j} = ReLU\left(FC_{i,j-1} \otimes FC_{i,j-2} \otimes \cdots \otimes FC_{i,0}\right) \tag{5.23}$$

where $ReLU$ is the non-linear activation function and $FC_{i,j}$ the $j$th $3 \times 3$ convolutional of the $i$th block. The input of the first residual block is $\Psi_1$.

Each convolutional layer $FC$ with input $\iota$ has bias $b$ and weights $W$ in such a

way that:

$$FC = W \times \iota + b \tag{5.24}$$

#### 5.3.2.4 Upsaling module

The upscaling module is inspired by ESPCN [328]. It is followed by a $1 \times 1$ convolutional layer and a $3 \times 3$ convolutional layer.

## 5.4 Experiments

In our application, the goal of the super-resolution method is to increase the resolution of our low resolution thermal images by a factor of 4: 80x60 pixels to 320x240 pixels.

### 5.4.1 Settings

#### 5.4.1.1 Dataset

The generalization of the deep network model depends directly on the data. Our main goal is indoor surveillance, and this is why we focused only on indoor thermal datasets. In [19], Wu et al. proposed a thermal infrared video benchmark for various visual analysis tasks. The dataset, proposed [19], is composed of various scenes, cameras views, and sequences. Among all these sequences, we considered two sequences focussed specifically on indoor situations: atrium-test (Atrium) composed of 8000 images and lab1-test-seq1 (Lab1) composed of 24000 images. The resolution of the images is 512 × 512 (cropped from 1024 × 640).

We used various views and sequences from Lab1 to construct our training dataset. Thus, we end up having 894 images.

For testing, we use 30 images from multiple perspectives and sequences of Atrium.

#### 5.4.1.2 High/low resolution image pairs

The training dataset is composed of pairs of high/low-resolution images. The low-resolution image is created by applying the degradation model (Equation 5.3) to a high-resolution image.

For each high resolution image, we randomly select a blur kernel standard deviation $\eta \in \{\eta_{min}, \cdots, \eta_{max}\}$ and Gaussian noise variance $\sigma^2 \in \{\sigma^2_{min}, \cdots, \sigma^2_{max}\}$. To implement the degradation model, we down-sampled a high resolution blurred image with an additive noise.

#### 5.4.1.3 Training settings

We follow the settings of [290], so we used $32 \times 32$ sliding patches from each low-resolution degraded image with the corresponding high-resolution patches. We used a stride of $16$. We also proceed with data augmentation by randomly flipping and rotating the patches. The batch size is set to $64$, and we train the network for $100$ epochs. For each epoch, we have $680$ iterations. Our model EFTS is implemented on top of Tensorflow, and the initial learning rate is set to $1e-4$. To update the weights, we used Adam optimizer.

#### 5.4.1.4 Comparison methodology

Given that we knew the ground truth (the high-resolution image), we evaluated the impact of our model by three metrics: the Peak Signal to Noise Ratio (PSNR), the Structural Similarity Index (SSIM), and the Edge Preservation Index (EPI) [329]. PSNR is a global quality index where SSIM and EPI are supposed to better understand the perceived change in structural information.

We took 30 images from Atrium database, and for each image we generated 456 low resolutions images with blur $\eta \in [0.2, 4]$ and Gaussian noise $\sigma^2 \in [5, 50]$. We had so 13680 degraded low-resolution images. For each methods, we computed the average of these three metrics over these 13680 images.

Even if the SSIM/EPI metrics are supposed to be related to the perceived quality, we also made some qualitative visual comparisons between the highly resolved image and the reconstructed one. To better highlight the reconstruction methods' impact on edge preservation, we create an Edges Map of each image using the Sobel operator. This allowed us to make some qualitative visual inspection of the edge preservation or degradation.

### 5.4.2 Depth of the network

With a deeper non-linear mapping model, we should normally be able to obtain better results. In [18], the authors show that using big values of D (number of residual blocks) and C (number of convolutional layers per residual block), the performance of the network is better. In their implementation, they use $D = 16$ and $C = 8$. However, for real-time execution purposes, we tried smaller values of $D$ and $C$. We have tried the following combinations $D3C1$, $D5C3$, and $D7C5$, but grid search also could be performed.

Table 5.1 illustrates the performance of each one of the network depths. It is noticeable that $D5C3$ outperforms both $D3C1$ and $D7C5$. When the number of layers increases, the number of network parameters also increases. Given very similar training data, a highly complex function fits the training data better than a less complex

one. Such complex function will memorize the training data, over-fitting, and the model performs poorly on the unseen data resulting in high generalization error. So, these results are overall due to the type of our dataset. Our focus is on the indoor scene super-resolution of people. Such scenes are limited to human shapes and contain less information than their visible counterparts.

| | D3C1 | D5C3 | D7C5 |
|---|---|---|---|
| PSNR/SSIM | 9.07/0.9549 | **39.37/0.9588** | 38.99/.9573 |

**Table 5.1:** Average PSNR and SSIM of 3 combinations of D (number of residual blocks) and C (number of convolutional layers). The best two results are highlighted in bold and underlined, respectively.

| | SKL | SKP | KPL |
|---|---|---|---|
| PSNR/SSIM | **39.39/0.9588** | 9.37/0.9576 | 39.21/.9586 |

**Table 5.2:** Average PSNR and SSIM of 3 combinations of edge operators (S Sobel, K Kirsch, L Laplacian and P Prewitt). The best two results are highlighted in bold and underlined, respectively.

### 5.4.3 Edge extraction module

We have investigated different types of combinations of edge operators to see which one is more suitable for super-resolution. We compare $SKL$ (Sobel, Kirsch, Laplace), $SKP$ (Sobel, Kirsch, Prewitt), and $KPL$ (Kirsch, Prewitt, Laplace). We use the same type of experiment as in section 5.4.2.

Table 5.2 illustrates that the model $SKL$ outperform $SKP$ and $KPL$. $KPL$ gives the second-best SSIM while regarding PSNR, $SKP$ gives the second-best results.

The fact that $SKL$ gives better results than $SKP$ can be explained by the fact that the Prewitt operator is derived from Sobel. So Prewitt operator does not bring more information to the network than the Sobel operator. In $SKP$ model, Sobel and Prewitt's operators are bringing almost the same kind of information about the edges.

$KPL$ is very close to $SKL$ in therms of SSIM, but the difference is more noticeable regarding PSNR. The main difference between these models is that Prewitt replaces Sobel. In [330], the authors reported that although Prewitt is similar to Sobel, there are differences in their spectral responses. As shown in table 5.2, our results demonstrate that noise suppression characteristics are better with Sobel than with Prewitt.

For all these reasons, we used $SKL$ (Fig. 5.4). As illustrated by this figure, we first extracted edges using the three operators. For Sobel and Kirsh operators, we have computed the edge magnitudes. The output of the edge extraction module is the concatenation of the results of the three operators.

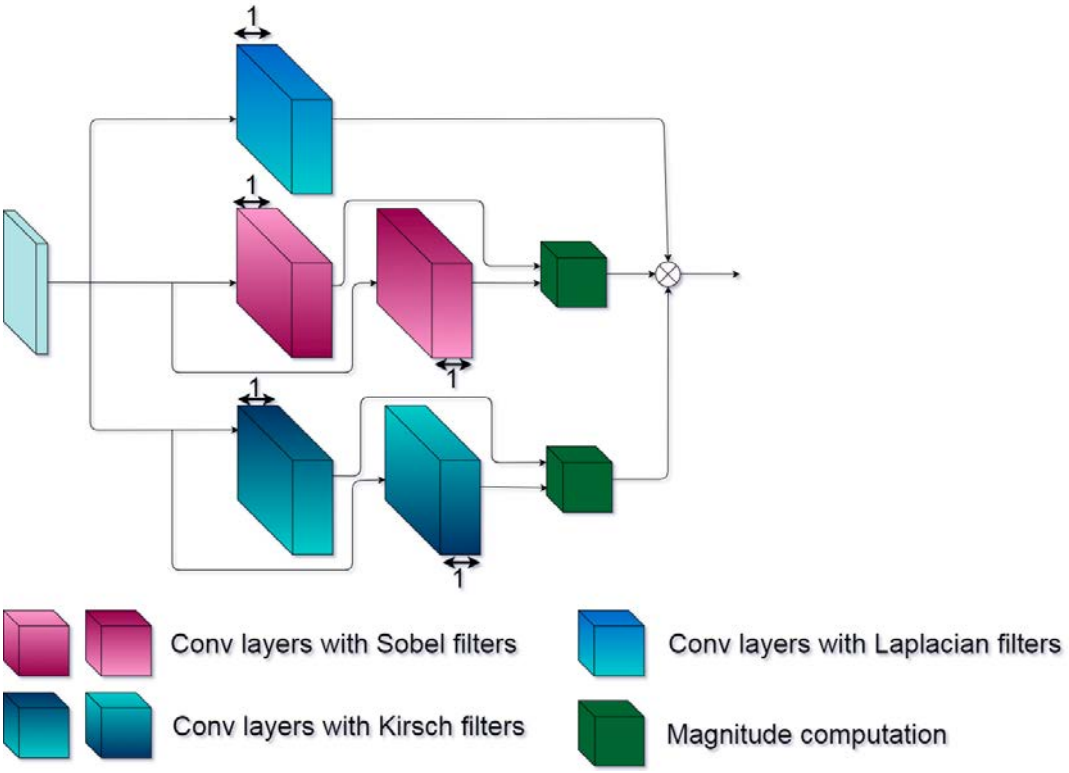



**Figure 5.4:** Edge extraction module

For our network, we use the model designed in Fig. 5.3. We use five residual blocks with 3 (convolutional+ RELU) layers in each. For all convolutional layers, the kernel size is $3 \times 3$ except fusion layers with kernel size set to $1 \times 1$.

### 5.4.4 Comparison with state-of-the-art methods

We evaluated our proposed EFTS method against other existing, state-of-the-art techniques in the literature. First, we compared our method with the methods developed for thermal images (TEN [16], CNN [17]) and as well as those which are developed for visible images (VDSR [284], LapSRN [302], RDN [18]). We use the same training dataset for all these models and the parameters they used in their respective papers for a fair comparison.

Most of these models source codes were available online except TEN [16] and CNN [17]. For these models, we have implemented their codes based on what is reported in [16, 17]. These re-implementations achieved expected Super-resolution results similar to those reported in the original articles. For RDN, we used the same number of blocks as ours, that is to say, $D = 5$ and $C = 3$. All comparisons are made for a scale of $\times 4$.

Table 5.3 shows quantitative comparisons with methods TEN and CNN (for thermal images) and VDRS, LapSRN, and RDN (for visible images). As evaluation

**Figure 5.5:** SISR using a blur kernel of 3 and a noise of 50

| Degradation | | Methods | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $\eta$ | $\sigma^2$ | Bicubic | VDSR [284] | LapSRN [302] | RDN [18] | TEN [16] | CNN [17] | EFTS |
| 1 | 5 | 39.27/0.9499 | 39.99/0.9670 | 39.51/0.9601 | 41.22/0.9703 | 37.42/0.9602 | 40.30/0.9659 | **41.65/0.9718** |
| | 35 | 35.50/0.8447 | 38.58/0.9503 | 38.45/0.9398 | 40.34/0.9643 | 36.56/0.9293 | 39.19/0.9498 | **40.71/0.9655** |
| 2 | 5 | 38.41/0.9339 | 39.49/ 0.9667 | 38.77/0.9526 | 39.71/ 0.9661 | 37.90/0.9486 | 40.00/ 0.9551 | **41.19/0.9677** |
| | 35 | 34.90/0.8149 | 38.08/0.9484 | 37.86/0.9301 | 39.59/0.9573 | 36.94/0.9143 | 38.98/0.9278 | **40.28/0.9592** |
| 3 | 5 | 36.28/0.9145 | 38.06 /0.9540 | 36.90/0.9398 | 38.80/ 0.9597 | 37.27/0.9337 | 37.87/0.9397 | **39.20/0.9614** |
| | 35 | 33.80/0.7847 | 36.76/0.9436 | 36.29/0.9144 | 38.17/0.9506 | 36.40/0.8968 | 37.23/0.9233 | **38.68/0.9518** |

**Table 5.3:** Comparison of EFTS vs state-of-the-art methods in terms of PSNR/SSIM. The best two results are highlighted in bold and underlined, respectively.

| Degradation | | Methods | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $\eta$ | $\sigma^2$ | Bicubic | VDSR [284] | LapSRN [302] | RDN [18] | TEN [16] | CNN [17] | EFTS |
| 1 | 5 | 0.9369 | 0.9598 | 0.9435 | <u>0.9619</u> | 0.9546 | 0.9562 | **0.9620** |
| | 35 | 0.9350 | 0.9591 | 0.9347 | <u>0.9610</u> | 0.9536 | 0.9555 | **0.9615** |
| 2 | 5 | 0.9505 | 0.9593 | 0.9464 | <u>0.9606</u> | 0.9532 | 0.9548 | **0.9609** |
| | 35 | 0.9453 | 0.9582 | 0.9377 | <u>0.9592</u> | 0.9522 | 0.9537 | **0.9594** |
| 3 | 5 | 0.9509 | 0.9579 | 0.9479 | <u>0.9578</u> | 0.9519 | 0.9528 | **0.9585** |
| | 35 | 0.9471 | 0.9561 | 0.9393 | <u>0.9566</u> | 0.9509 | 0.9521 | **0.9567** |

**Table 5.4:** Comparison of EFTS vs state-of-the-art methods in terms of EPI. The best two results are highlighted in bold and underlined, respectively.

| Degradation | | Methods | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $\eta$ | $\sigma^2$ | Bicubic | VDSR [284] | LapSRN [302] | RDN [18] | TEN [16] | CNN[17] | EFTS |
| 1 | 5 | 24.99/0.8234 | 26.48/0.8196 | 25.54/0.7554 | <u>27.42/0.8315</u> | 25.61/0.7432 | 26.54/0.8040 | **27.64/0.8375** |
| | 35 | 23.92/0.6329 | 26.06/0.7449 | 25.40/0.6866 | <u>27.04/0.8123</u> | 25.22/0.5969 | 26.31/0.7523 | **27.27/0.8171** |
| 2 | 5 | 24.95/0.6335 | 26.17/0.8040 | 24.87/0.6956 | <u>26.98/0.8189</u> | 24.70/0.6695 | 25.52/0.7360 | **27.13/0.8219** |
| | 35 | 23.88/0.4386 | 25.73/0.7275 | 24.79/0.6192 | <u>26.35/0.7945</u> | 24.42/0.5171 | 25.38/0.6806 | **26.49/0.7959** |
| 3 | 5 | 24.10/0.4733 | 25.68/0.7876 | 24.16/0.6318 | <u>26.09/0.7992</u> | 23.91/0.6069 | 24.40/0.6676 | **26.39/0.8010** |
| | 35 | 23.35/0.2580 | 25.35/0.7352 | 24.10/0.5482 | <u>25.51/0.7648</u> | 23.70/0.4545 | 24.32/0.6133 | **26.10/0.7767** |

**Table 5.5:** Comparison of EFTS vs state-of-the-art methods in terms of PSNR/SSIM of the edge maps. The best two results are highlighted in bold and underlined, respectively.

metrics, we use PSNR and SSIM for images degraded by different noise and blur kernel values. Among thermal image-based methods, CNN gets the closer results to EFTS while TEN is diverging. The performance of RDN is very close to EFTS. Such results can be explained by the fact that EFTS uses residual blocks like RDN, but its edge extraction module performs better in comparison. TEN is the shallower network with only $4$ layers tends to provide less reconstruction quality than Bicubic interpolation for the values $(1, \sqrt{5})$ and $(2, \sqrt{5})$. Overall, EFTS performs better than CNN with a noticeable amount of dB PSNR in most cases. TEN is diverging, giving sometimes worst results compared to bi-cubic interpolation.

While PSNR and SSIM are essential for comparing the original and reconstructed images, the PSNR/SSIM between the Edges Map of these two images also brings more information in the quality of the reconstruction. Many thermal image applications are based on edge extraction. Therefore, we evaluated our method's performance by computing the PSNR/SSIM of the Edges Maps. As shown in table 5.3 and table 5.5 the performance of EFTS is also better than RDN. EFTS and RDN performance are more stable to degradation variations than TEN and CNN. In these methods PSNR decreases almost for 2 dB from $[1, \sqrt{5}]$ to $[3, \sqrt{35}]$. The Edge Preservation Index $(EPI)$ calculates the number of edges preserved in an image after applying each method to the original low-resolution image. Table 5.4 confirms the results reported in table 5.3 and table 5.5. It is noticeable that EFTS and RDN results are very close, but EFTS still outperforms TEN and CNN.

Figure 5.5. shows qualitative comparisons between EFTS, RDN, CNN, and TEN. In the reconstructed images and their Edges Maps, it is noticeable that EFTS is more able to enhance edges than the other methods. The edges extracted by RDN are comparable to edges extracted by EFTS, but we can see that our model responds equally to edges and does not enhance certain parts of edges while weakening other parts. Some artifacts can be seen, while our reconstructed images contain no artifacts. TEN and CNN results proved that these methods are not suitable for edge-based thermal applications with this degradation model.

### 5.4.5 Application on a real low-resolution camera

Our model's primary goal is to apply super-resolution in real-world applications with very low-resolution thermal images. This is why we have acquired indoor images using Lepton 2. The resolution of such images is $80 \times 60$. Fig 5.6. shows the qualitative results super-resolution of such images. The originals images are very noisy and practically unusable. By performing super-resolution, we want to get more information about the scene.

Fig 5.6. points out super-resolution applied without ground truth. We used the same trained network as in earlier sections. It is easily noticeable that EFTS allows

reconstructing more details than CNN and TEN. The output images of these later methods contain some artifacts. CNN provides better results than TEN, and the reconstructed image is less blurred.

Training our model with several degradation model settings allowed our network to generalize better. So such a network can be used for very low-resolution images $(80 \times 60)$.

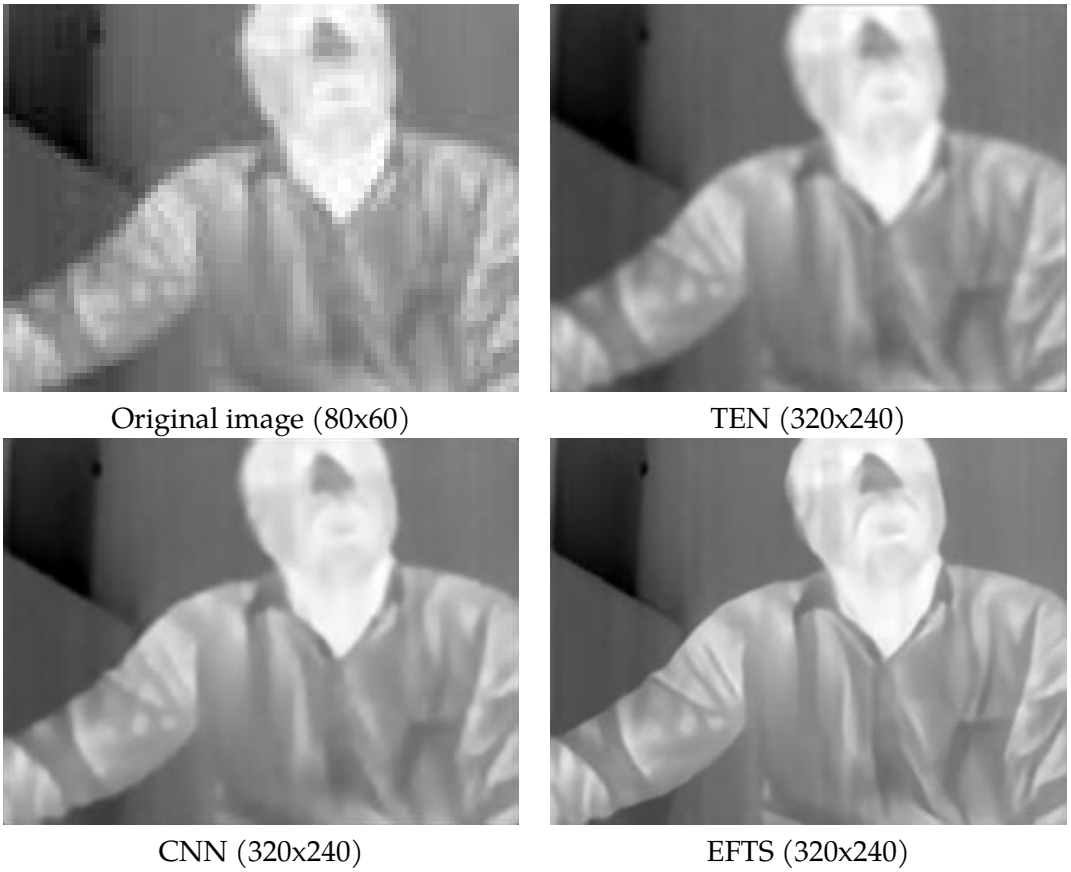



**Figure 5.6:** Super-resolution of low-resolution thermal image of a person sit in front of the camera

## 5.5 Conclusion

We proposed a network to perform thermal image super-resolution to handle several kinds of degradation via a single model. Unlike previous thermal image super-resolution methods, we use residual blocks and, above all, an edge extraction that allows us to obtain stronger reconstructed edges. Moreover, we evaluated the performance of our proposed model on PSNR/SSIM/EPI of reconstructed images and their edges maps. All these evaluations metrics confirm that the edge extraction module improves the results.

The edge extraction module comprises three edge extractors (Sobel, Prewitt, Laplacian) that are concatenated with the original low-resolution image and are fused to extract shallow features. The results on real very low-resolution images acquired from Lepton2 show that we can significantly enhance the resolution of such images.

Here, our degradation models included isotropic blurring and Gaussian noise; however, we should consider that thermal images are also affected by other degradation models such as motion blur. To increase our network generalization and real-world performance, we must take into account such degradation. Moreover, in indoor surveillance, it is possible to associate two thermal sensors together or a thermal sensor with another type of sensor. It could be possible to use disparity to enhance thermal image resolution even further.

# 6 Detection of fallen person

> Sometimes it takes a good fall to really know where you stand.
>
> Hayley Williams

## 6.1 Introduction

In the Chapters 3, 4 and 5 we proposed a set of approaches to perform and improve stereo-vision. Our current pipeline is defined in Fig 6.4: super-resolution, robust stereo calibration, features extraction, sub-pixel matching, and finally, 3D reconstruction of the scene. But given that we do not know where the ground is, 3D reconstruction does not allow us to distinguish a lying person from a fallen person. Besides, even if the stereo system is parallel with the ground plane, the fallen person's detection might not be accurate.

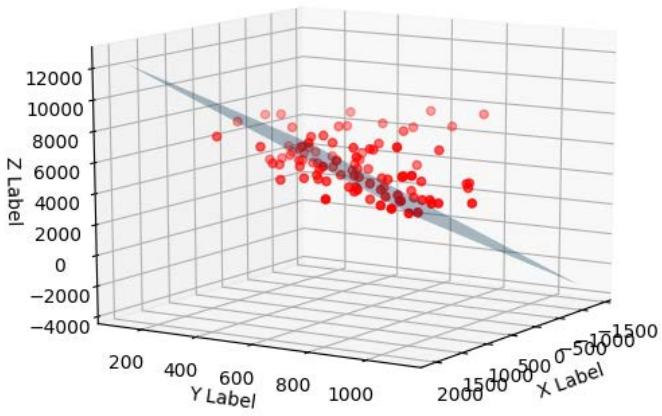



**Figure 6.1:** 3D points of the ground for the dataset Poseidon

To resolve this problem, one can propose determining the ground plane and computing the distance between a given 3D point and the ground plane. Using a hot bulb on the ground, it could be possible to triangulate and construct the ground plane.

We took the image of the hot bulb at many positions in each plane. The bulb

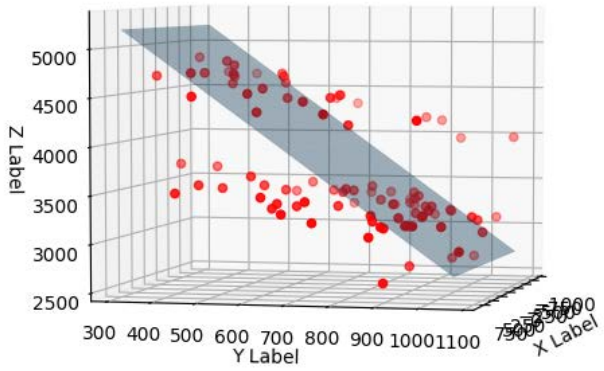



Figure 6.2: 3D points of the ground for the dataset Thales

was the hottest object, so the segmentation is very straightforward, and it is possible to detect the center of gravity of the bulb easily. So we collected two datasets:

— Poseidon: composed of 97 image pairs of the ground;

— Thales: composed of 95 images pairs of the ground.

For each of these datasets, the position of the camera is different. We fitted each plane using the RANSAC algorithm [331].

The Fig 6.1 and Fig 6.2 show the 3d points distributions and the fitted plane using RANSAC. These figures show that the 3D points are quantified. Besides, Fig 6.3 shows that there is an error around 50 cm. Such an error is very consistent. Given that the average height of a person is from 160 cm to 180 cm [332], we could not use this method to detect a fallen person.

Such results can be explained by the low-resolution aspect of our camera which leads us to big triangulation errors. A solution could be to bypass the low-resolution by super-resolution. This chapter will unplug the super-resolution module and investigate if it is possible to detect a fallen person by learning the ground plane. Thus, we add two new modules (Fig 6.5): Ground plane detection and Fall detection. The new framework is named $TSFD$ for Thermal Stereo Fall Detection.

First, given that our device is not wearable and preserve privacy, our solution is not invasive. Besides, by learning the ground, we do not explicitly learn what a fall is, so our solution is not influenced by data unbalancing. Moreover, using stereo thermal

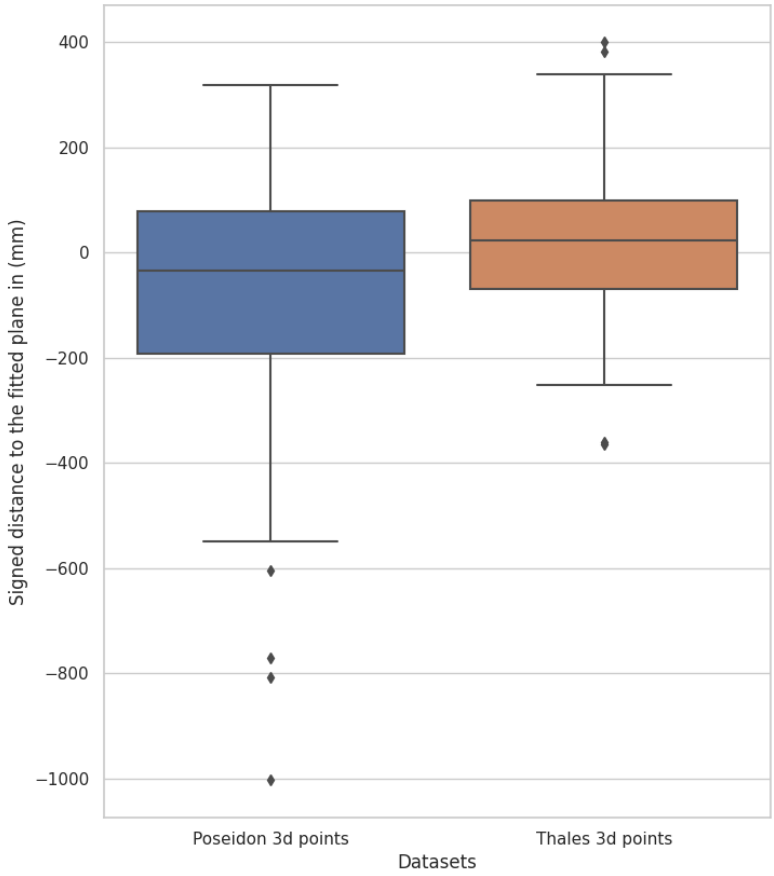



**Figure 6.3:** Box plot of the signed distance between 3D points and fitted planes

cameras enabled us to segment human bodies easily. To the best of our knowledge, it is the first time fall detection is performed using such a method. Our contributions are three-fold:

1. We proposed a stereo-based solution which does not need any camera calibration;
2. We then provided a solution to learn the ground plane using very low-resolution thermal images efficiently;
3. Finally, we provided a solution estimating the threshold to get a good classifier to detect if a person is on the ground or not.

The chapter is structured as follows: Section 6.4 presents our framework $TSFD$. It details the implementation of the features extraction method for thermal images based on phase congruency, the stereo matching method, the point classification, and the generic fall detection. In Section 6.5, we discuss, analyze, and explain the results. Finally, section 6.6 concludes the chapter and gives perspectives.

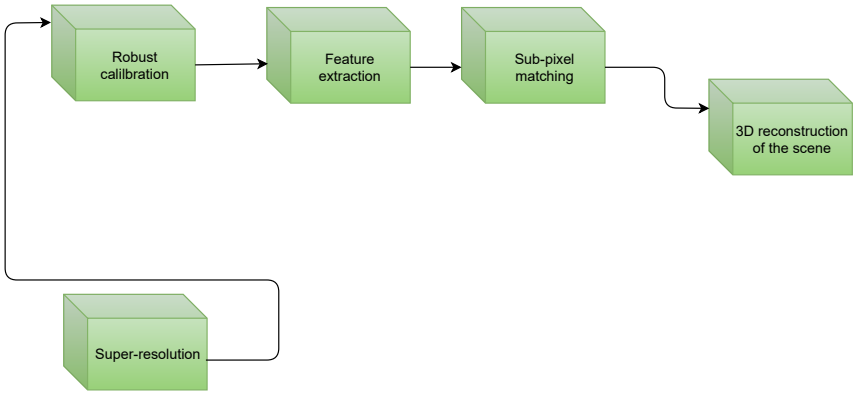



**Figure 6.4:** Stereo vision pipeline: super-resolution, robust stereo calibration, features extraction, sub-pixel matching and finally 3D reconstruction of the scene

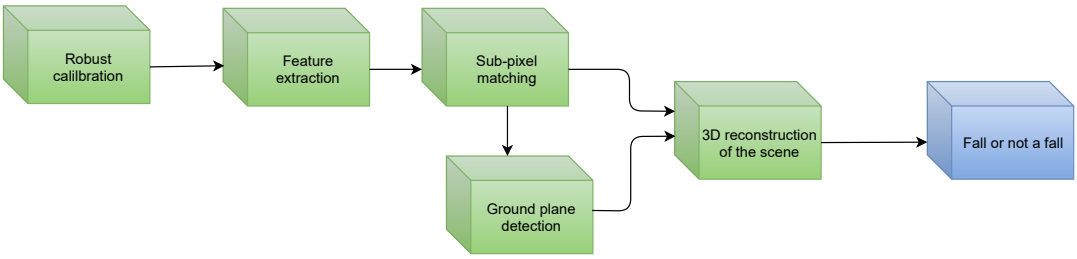



**Figure 6.5:** Fall detection pipeline

## 6.2 State of the art on fall detection

Many works propose fall detection solutions based on ambient sensors [98], vision-based sensors [99] and wearable sensors [97]. Each of these sensors has advantages and drawbacks, as we showed in the Chapter 1.

Regarding vision-based methods, RGB cameras are the most used. In [333], Foroughi et al. propose a fall detection method composed of 4 modules: object segmentation, feature extraction, motion classification, and fall occurrence detection. However, their segmentation method is very straightforward, given that in the dataset, there is no occlusion and no furniture. In [334], Yang et al. use RGB cameras combined with Kinect to determine a fall by shape analysis of 3D depth images. Contrary to RGB cameras, depth cameras preserve the anonymity of people. In [335], Rougier et al. propose a fall detection method using depth videos. They use two types of features: human centroid height relative to the ground and body velocity to be robust to occlusion. They perform ground plane detection, person segmentation, and localization, and, finally, fall detection.

By combining various types of sensors, it is possible to increase accuracy. In [336], Kwolek and Kepski propose a fall detection system combining an accelerometer and a Kinect through a fuzzy inference. Using an SVM, they have got an accuracy of 98.33% and a sensitivity of 100%. Moreover, more recently, in [337], Halima et al. propose depth and thermal image fusion in fall detection. Using an adaptive weighted particle filter, they can track people even with fast motion (fall), partial occlusion, and scale variation.

In [139], Kido et al. use a thermal camera to detect falls in toilets. The camera is placed on the ceiling. Each thermal image is down-sampled into $9 \times 9$ by average down-sampling. Then, they perform discrimination analysis [338] on these 81 values. This method's main drawback is that it is adapted to very confined space and cannot be used for large rooms. In [339], Sixsmith and Johnson propose a system to identify seniors fall and emitting an alarm when a fall occurs. Their method is based on an elliptical-contour gradient-tracking system. To detect falls, they classify vertical velocity-based features using a neural network. In [340], Wong et al. propose a fall detection system based on binary segmentation and human shape statistics. Their method achieves an accuracy of 96.15%. In [341] and [342], Han and Bhanu go further by trying to recognize human activities such as walk/run using thermal cameras and also Spatio-temporal information. However, they use in their work a monocular system that does not take advantage of depth information. With the increasing popularity of deep networks, many authors propose to learn fall directly from thermal images. In [21], Quero et al. propose a method based on convolutional neural networks to detect falls. They used images resolution of $32 \times 31$. They experiment in a small room, so they test two situations single occupancy and multiple occupancies. Their method gets an accuracy of 91.93% for a single occupancy context, which drops to 85.70% when considering multiple occupancies. In [22], Nogas et al. propose a fall detection method based on CNN (Convolutional Neural Network) and LSTM (Long Short-Term Memory). They formulate the fall detection problem as anomaly detection. None of these two proposed methods exploit depth information to detect a fall. We think that

these results can be improved by inferring from more than one view.

Sometimes for vision-based techniques, before performing the fall detection, one can first detect the ground plane. Ground plane detection is a well-studied problem in computer vision for visible images or Kinect data. **However, how to determine the ground plane in low-resolution thermal images?**

Most of the time, such a problem is not related to fall detection but rather to robot navigation. For example, in [343], Pears and Liang propose to track visible images corner point, frame by frame, and use co-planar relation to segment ground. In [344], Zhou and Li assume that a fixed camera and the robot can only rotate. To ensure robustness, they use normalized homography. In [345], Bojian Liang et al. rectify images using the reciprocal-polar technique; they perform ground plane segmentation by fitting a sinusoidal model of the reciprocal-polar space. In [346], the authors present in parallel a first pipeline (SURF features extraction, feature matching, robust homography estimation) and a second pipeline (image segmentation, segments comparison). They use robust homography to identify *ground* features and *not ground* features, and then the segments are updated. Ground segmentation is performed using Multi-scale Mean-shift Clustering. In the reported results, they try to estimate the robot's distance to an object, but the error goes from 0.03 m (when the object is 1.6 meters away) to 0.67 m (when the object is at a distance of 6 m). In [347] Low and Manzanera propose a framework composed of three phases:

1. The initialization phase: they define a world model using a distribution fusion method based on super-pixels;
2. The operation phase: new images are classified using Markov Random Fields;
3. The update phase: the woald model is updated.

Recently, with Kinect, ground detection became easier, and most of the time, it is associated with human detection of fall detection [127, 335, 348, 349]. For example, in [335], Rougier et al. use Kinect to detect the ground plane using a V-disparity approach.

Regarding thermal images, to the best of our knowledge, no work has yet been done to identify the ground. Usually, the ground cannot be distinguished in thermal images: even for human eyes, segmenting the ground can be hard in such images. One solution could be to place a heat-emitting object on the floor. For stereo thermal images, it could be possible to fit the ground plane using this object placed in multiple places. Then using triangulation, it could be possible to model the ground plane. **But how accurate is triangulation applied to our thermal images?**

The distance z from the camera to a given 3D point is given by the Equations 4.17 and 4.18. In [228], Maier-Hein et al. apply stereo-vision for medical imaging. They find

depth estimation to be most inaccurate when the observed object is more than 10 to 15 times the baseline away. Given our 16 cm baseline, we can expect depth estimation to be increasingly inaccurate when the object is further than 160 cm away (from the camera). Regarding the indoor and outdoor environment, in [350] and [351], Mur-Artal Raul and Tardós Juan D empirically prove that depth is accurate when the depth is less than about 40 times the stereo baseline. Assuming this holds for our setup, the maximum observation distance increases from 160 cm to 480 cm. While Mur-Artal Raul and Tardós Juan D use images with a resolution of $1240 \times 376$, Maier-Hein et al. stipulate this assumption whatever the resolution. In [4] and the Chapter 4, we propose a sub-pixel stereo matching method to compensate for the low resolution of their thermal images. In some experiments, we find that an error of $\pm 1$ pixel can lead to an error of $\pm 50$ cm in depth estimation. Anyway, it isn't easy to use our cameras to perform 3D vision for ground plane detection. Since the 3D reconstruction approach is not working well for our cameras, we considered a learning-based approach. So, we learned the geometry of the room, and given a 3D object, our framework could be able to determine how many percent of this object is closed to the ground.

Most of the time, datasets used in the fall detection system are unbalanced. Typically in most of the datasets available in the literature, there are more *no fall* than *fall* occurrences [29]. For example, in [352], there were 828 frames of falls out of 36 391 frames. Learning from such an unbalanced dataset is not easy. There are many solutions to deal with such issues as data-level [353], algorithm level [354] and ensemble methods [355]. Even if these methods are efficient, we thought that it might be possible to deal with a small number of falls in the dataset by not learning what is a fall but how a person is closed to the ground.

## 6.3 Notations

Given a video $Vid$ composed of a list of $N$ frames, we tried to perform static fall detection for each frame. We defined some terms as:

— Vid $= \left\{(I_0^{left}, I_0^{right}), \ldots, (I_N^{left}, I_N^{right})\right\}$ where $I_t^{left}$ and $I_t^{right}$ are respectively the left and the right images acquired at time t;

— $\Omega_i$ the set of matches for the $i$th image pair. $\Omega_i = \left\{(P_0^{left}, P_0^{right}), \ldots, (P_M^{left}, P_M^{right})\right\}$ where $P_j^{left}$ and $P_j^{right}$ represent respectively the $j$ th left and right matches (2D points);

— For a specific left frame, the segmentation of the hottest object can output more than 1 cluster of points. Each cluster of points is noted $Object_k$.

— Size$(Object_k)$ is the number of pixels belonging to $Object_k$.

— $Dist_{a \leftrightarrow b}$ is the minimal euclidean distance between $Object_a$ and $Object_b$;

— Given $\Omega = \left\{(P_0^{left}, P_0^{right}), \ldots, (P_m^{left}, P_m^{right})\right\}$ for a specific images pair, $\Lambda = \{v_0, \ldots, v_m\}$. $v_j = 1$ when $(P_j^{left}, P_j^{right})$ represent a 3D point on the ground and 0 when it is not;

— $percent_i$ is the percentage of 3D points on the ground for the $i$ th image pair. It is the percentage of $v_j$'s equal to 1 for a given $\Lambda_i$;

— $Pred[i]$ is the boolean prediction of fall or not a fall for the $i$ th image pair;

— $\gamma$ is the phase congruency threshold;

— $\kappa$ is the distance between a 3D point and the ground;

— $T$ is the threshold on the percentage of points on the ground over which an object is considered as laying on the ground. In this chapter, we used two thresholds: $T_{ad}$ which is *ad hoc* defined threshold and $T_{pred}$ which is pre-determined threshold;

## 6.4 Materials and method

To perform fall detection in stereo thermal images, we want to exploit the depth information to enrich the prediction. A naive solution could be: (1) calibrate the cameras, (2) estimate the ground plane using a hot bulb placed at least on three locations on the ground, (3) for any stereo frame, predict a fall/no fall by computing the distance $\kappa$ of each 3D point with the ground plane (e.g. if someone is laying on the ground ($\kappa_t \approx 0$) or is falling if $\frac{\delta\kappa}{dt} \leq 0$ for a dynamic context). Unfortunately, given our small stereo baseline and the low camera resolution, it is challenging to get a precise triangulation result, even with sub-pixel stereo matching [4].

Rather than trying to deal with incorrect plane fitting, we changed our track. For a specific 3D point, the value of $\kappa$ answers to the question "is this given 3D point on the ground or not?".

This led us to define a new ground detection framework ($TSFD$) composed of 3 main parts (Fig. 6.6):

1. **Stereo pipeline:** a set of matched points of the hottest object (Section 6.4.1) (Fig 6.7).
2. **Points classification**: given that we did not choose not to fit a plane representing the ground, we had to find an algorithm $A$, trained to classify at each time if a pair of matched points is close to the ground plane or not (Section 6.4.2).
3. **Generic fall detection**: using statistics on the output of the previous step, we could easily infer whether an object lies on the ground or not (Section 6.4.3).

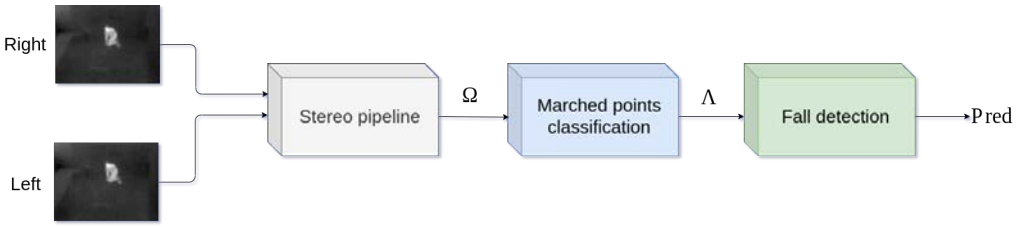



**Figure 6.6:** Framework overview

### 6.4.1 Stereo pipeline

The cameras we used are characterized by noise. This is why it is essential to find a more robust way to extract features. Such features must not only be robust to noise but also to brightness change. As written in the Chapter 1, at odd times, the brightness is changing suddenly.

To handle these issues, we proposed the following stereo pipeline composed by:

1. ***Robust feature extraction***: phase congruency to extract features.
2. ***Binary segmentation*** and ***stereo matching***: We applied binary segmentation and we performed similarity matching using Lades similarity. The output is a set of matched points.
3. ***Outliers suppression***

#### 6.4.1.1 Features extraction

Regarding features extraction and stereo-matching, our framework is built on top of the method proposed in [4]. For each image, the phase congruency and its moments are computed. Phase congruency is defined as reported in Section 4.4.1.

#### 6.4.1.2 Stereo-matching

In this framework, we decided not to calibrate the stereo-cameras. So our stereo pair are not rectified. This means that the matching can occur in both directions $x$ and $y$. We performed the matching using an assumption that the disparity range is $\in [-5, 5]$ pixels in $x$ and $\in [-5, 5]$ pixels in $y$. These ranges could be updated in a multi-frame-based framework.

The matching is performed in the phase congruency maximum moment $M$-space (Eq. 4.24). The threshold $\gamma$ thus becomes very important. Using a low value of $\gamma$ guarantees a fine grain matching at a disadvantage of computation time and robustness (see Section 4.4.2 for details and discussion).

For a feature in $I_{left}$, the corresponding features is searched in $I_{right}$ within the disparity windows using Lades similarity.

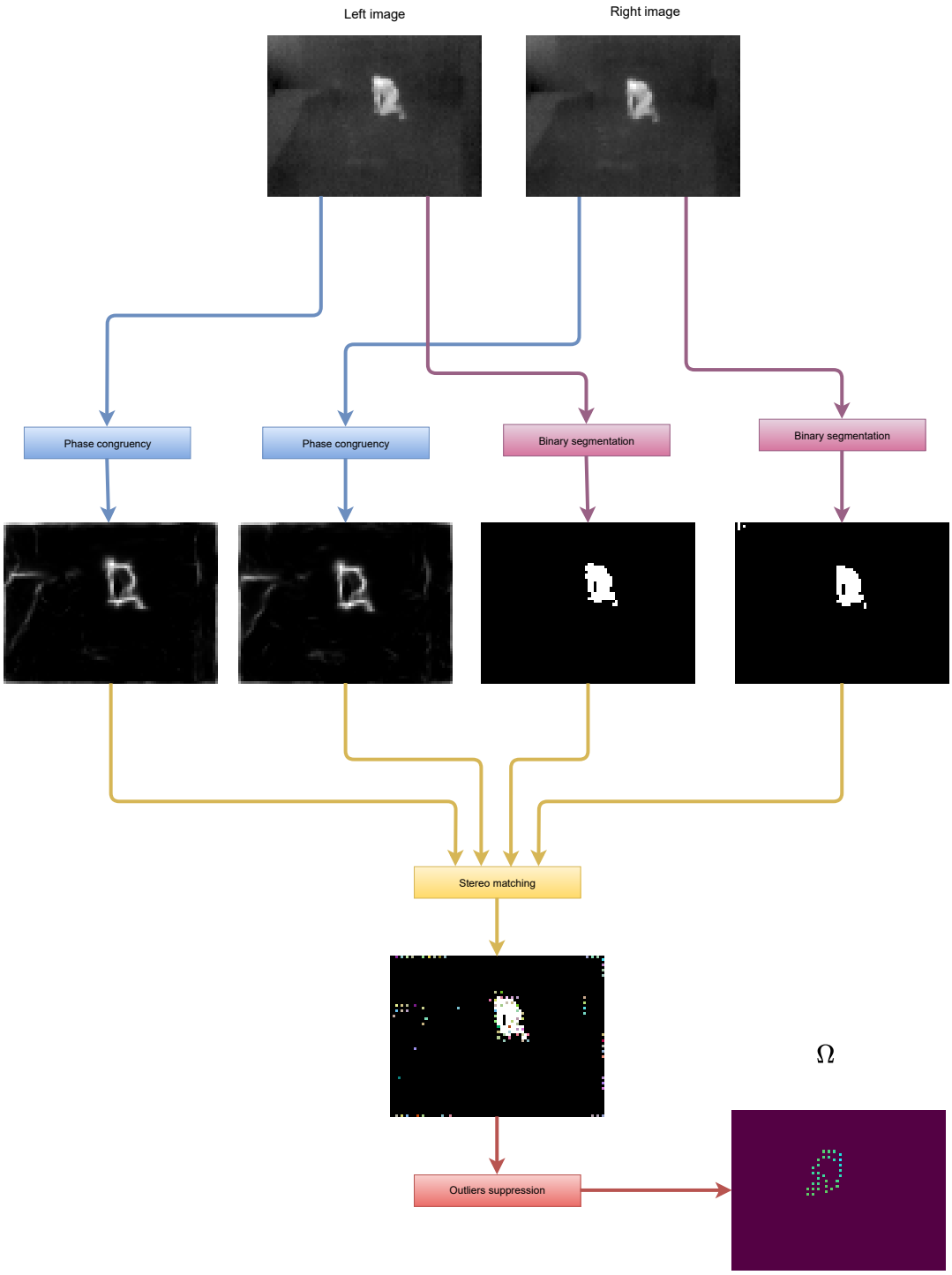



**Figure 6.7:** Stereo pipeline overview (*best viewed in colors*)

We took into account matching constraints such left-right consistency, uniqueness, continuity, and ordering constraints.

#### 6.4.1.3 Binary segmentation and outliers correction

Given that the values range of a human body in thermal cameras is relatively stable, some *ad hoc* threshold can be set to segment human shape. To discard miss-segmentation and smaller hot objects, we clustered together with the segmented areas using area size and distance criterion.

A morphological dilation of 5 pixels is applied to the binary shape in order to produce a mask, and only the features that are located inside this mask are kept. Some outliers have also been discarded in terms of disparity (using the classical definition of outliers in a boxplot proposed by Tukey [356]).

At this step, the output is $\Omega$ as the set of matched points belonging to the person.

### 6.4.2 Points classification

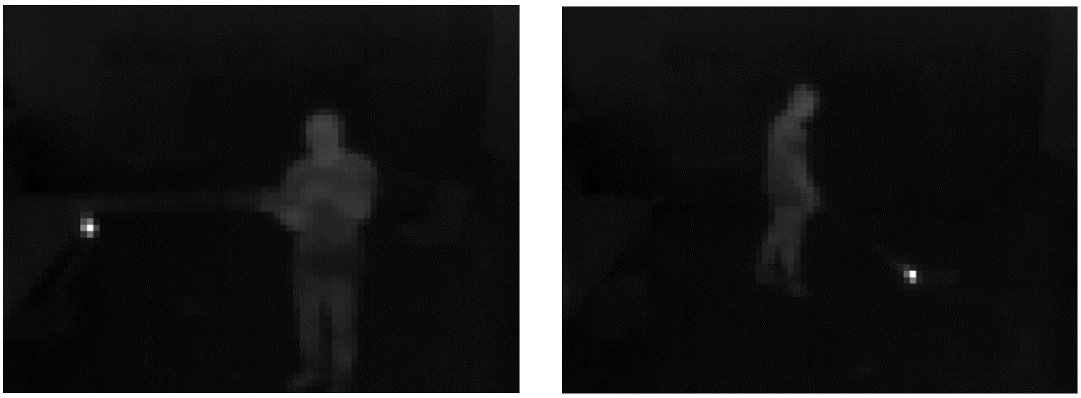
Our main idea is now to determine whether the matched points in $\Omega$ are on the ground or not. For this, we needed an algorithm $A$, which will learn to classify whether a given matched point is on the ground or not. Because we could not perform accurate stereo reconstruction [4], our strategy will be to perform supervised learning.

This section presents the learning process and investigates two classifiers: (1) SVM ([357]) applied on point coordinates; and (2) a proposed deep learning-based model applied to image information.

#### 6.4.2.1 Learning the ground

The goal is to learn whether a pair of matched points is on the ground or not. But as stated in Section 6.1, it is almost impossible to detect the ground in low-resolution thermal images. So, the main idea of the learning will be to set a hot-spot (a bulb) on the ground and not on the ground. The figure Fig 6.8 shows some images where a person holds a lamp by a long handle. The lamp is easy to segment because it is usually the brightest object. We also set minimal surface criteria to handle occlusion of the lamp.

Given that falls are rare, direct acquisition of daily activities will lead to highly imbalanced data for the training of the classifier. Our learning strategy allowed us to have a balanced dataset containing as many points on the floor as not on the floor.



(a) Not on the ground (b) On the ground

**Figure 6.8:** Left views of images of the bulb

#### 6.4.2.2 Classification strategies

The learning dataset is composed of images pair of spots while the output of our stereo-pipeline is composed of a set of the 2D coordinates of the matched features.

To associate these two types of information (images vs. coordinates), we investigated two strategies:

1. coordinate-based strategy: From the images of the learning dataset, we extracted the coordinates of the hotspot centers. Then, a standard SVM classifier is trained on these coordinates;
2. image-based strategy. To integrate the specifications (low-resolution, noise) of our images in the classifier, we had to perform the learning directly on the thermal images. But if we wanted to use the classifier, we should convert the coordinates of matched points into some image information. So one pair of coordinates will give a pair of images used as input of the classifier. To handle this more complex image classification problem, we proposed a deep learning-based model inspired by DenseNet [20].

**Coordinate-based strategy:** In the learning set, we estimated the coordinate of the hotspot is estimated as following: 1) we search the brightest points in the image, 2) the coordinate of the hotspot is then estimated as the center of gravity computed in a small window around the brightest point. We trained a linear SVM using as input these coordinates $P_{gc} = \{(x_l, y_l), (x_r, y_r)\}$ (where $(x_l, y_l)$ and $(x_r, y_r)$ represent respectively the coordinates in the left and right images). The learning set is labeled as on the ground or not on the ground. To linearly separate the points, SVM determines a hyperplane that separates the points located on the ground from those which are not.

For the inference, the SVM uses a set $\Omega$ as input and outputs a set $\Lambda$ of classified

matched points.

**Image-based strategy** In the previous strategy, the coordinates of the hotspot is estimated as a center of gravity. This strategy assumed that there is a Gaussian distribution around the brightest point. Unfortunately, this is rarely the case; we needed to let the algorithm adapt it to the point distribution.

We proposed an adapted version of DenseNet [20] (DGD for Dense Ground Detection) to predict from a pair of thermal images whether a point is on the ground or top of the ground. The network's inputs are the left and right views of the hot lamp, and the output is the classification of the point (on the ground or not).

The network we proposed is shown in Fig 6.9. Given an image pair, $I_{left}$, $I_{right}$, we first applied in parallel a 7x7 kernel. After this step we got respectively $Shallow_{left}$ and $Shallow_{right}$. We then concatenated them and applied a 3x3 kernel followed by a 1x1 kernel to fuse these two views. The rest of the network is mainly DenseNet [20]. The network is composed of dense blocks separated by transition layers. In each dense block, there are bottleneck layers. As reported in the original paper, we used BN-ReLU-Conv(1x1)-BN-ReLU-Conv(3x3) for each bottleneck.

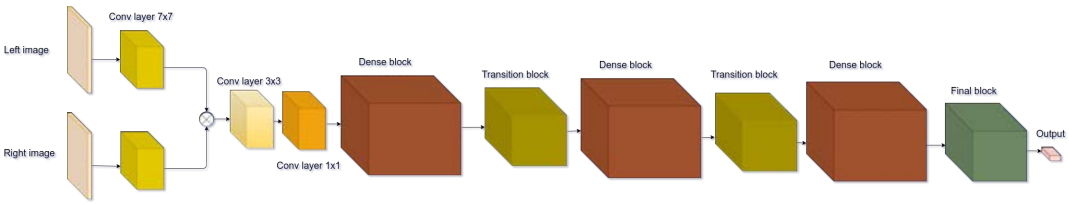



**Figure 6.9:** Ground plane detection based on DenseNet

During inference, for each matched point of $\Omega$ we created a pair of images (left and right) by setting and sampling a Gaussian distribution centered on the point coordinates. We set the standard deviation of the Gaussian at 3 pixels (Fig 6.10). $\Lambda$ is the concatenation of the classification for each matched point.

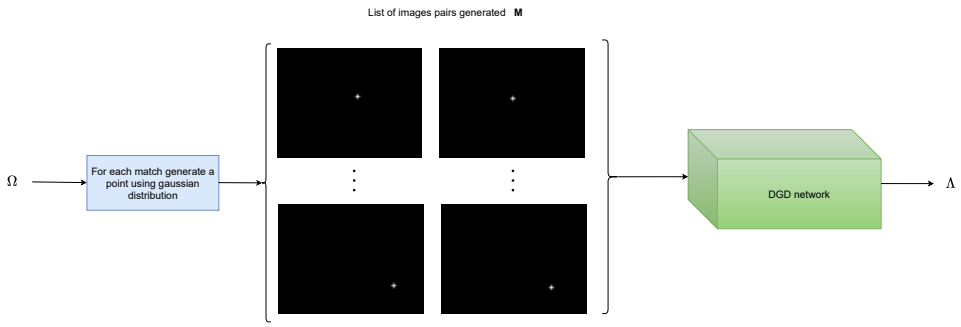



**Figure 6.10:** Fall detection process using DGD

### 6.4.3 Generic fall detection

In a static context of a fall, a human shape's features must be closed to the ground (classified as on the ground). On the contrary, only a few features (feet) should be classified as being on the ground when the person is standing or sitting.

So, it is straightforward that a threshold-based classifier can detect when a person is lying on the ground or not. If the percentage of the object's features on the ground is higher than a certain threshold of $T$, this object is on the on-ground else we could say that this object is not on the ground. This threshold can be set heuristically ( ad hoc $T_{ad}$) or inferred ($T_{pred}$) from a sequence of another person.

## 6.5 Results and discussion

### 6.5.1 dataset

To evaluate our method $TSFD$, we acquired and annotated several datasets (Table 6.1):

— Bug-saalle (Ember in Mossi language): This dataset comprises 6000 images pairs where the bulb is on the ground and 6000 images pairs where the bulb is not on the ground. This dataset has been used to evaluate the points classification methods.

— Tuulmen Puiir (Shared heat in Mossi language): In this dataset, we tried to see if $TSFD$ is generalizable. This dataset is composed of a learning set of 12000 images pairs as above plus four sub-datasets acquired with four different persons. The size of the persons was from 1.6 meters to 1.8 meters.

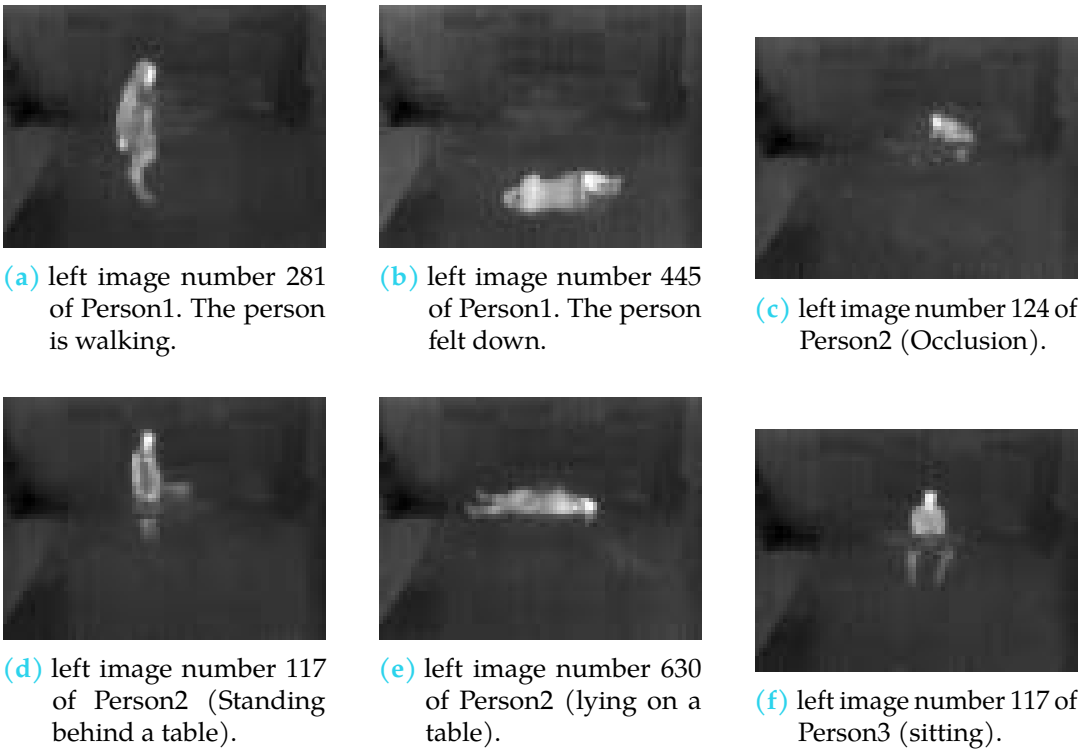
In all sub-datasets, the persons were standing, walking, sitting, and falling. Our dataset is imbalanced as most of the fall detection dataset. For example, in [22], there were 828 frames of falls out of 36 391 frames. These datasets are available on demand.

### 6.5.2 Comparison metrics

We manually annotated each dataset to establish the ground truth. So for classification we could define True Positive ($TP$), True Negative ($TN$), False Positive ($FP$) and False Negative ($FN$). To compare the methods we used the metrics as follow:

— Sensitivity measures the proportion of actual positives that are correctly identified as such :

$$Sensitivity = \frac{TP}{TP + FN} \tag{6.1}$$



(a) left image number 281 of Person1. The person is walking.

(b) left image number 445 of Person1. The person felt down.

(c) left image number 124 of Person2 (Occlusion).

(d) left image number 117 of Person2 (Standing behind a table).

(e) left image number 630 of Person2 (lying on a table).

(f) left image number 117 of Person3 (sitting).

**Figure 6.11:** Left images pair selected on the sub-datasets

— Specificity which measures the proportion of actual negatives that are correctly identified as such:

$$Specificity = \frac{TN}{TN + FP} \tag{6.2}$$

— Accuracy which indicates the proportion of correct predictions:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.3}$$

— F1-score which is a measure of a balanced test accuracy:

$$F_1 = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{6.4}$$

The F1 score is the harmonic mean of the precision and recall.

### 6.5.3 Implementation of the framework

The stereo pipeline has been implemented in C++. We used the libraries FFTW [358], OpenCV [192] and Eigen [359].

For the SVM classifier, the code has been written in Python, and we used the implementation provided by sklearn [360].

Table 6.1: List of datasets we used in this chapter

| dataset name | Type target | Folder |
|---|---|---|
| Bug-saalle | Points | Bug-saalle |
| Tuulmen Puiir | Points | Tuulmen Puiir-points |
| | Person | Person0 (1869 images with 236 falls)<br>Person1 (1709 images with 254 falls)<br>Person2 (1628 images with 296 falls)<br>Person3 (1506 images with 394 falls) |

For DGD, the code has been written in Python on top of Tensorflow [361]. To train our model, we set the number of bottleneck layers in each dense block to respectively 1, 2, 4, and 8. The transition layer improves the model compactness by reducing the number of feature MAPS [20]. We set the reduction rate $\theta = 0.5$, so dividing by two the number of feature map output of the precedent dense block. We used a grow rate k = 24. The initial learning rate is set to $1e^{-4}$, and we used AdamOptimizer [362] to update the network weights.

### 6.5.4 Comparison of classifiers

In order to compare the classifiers, as recommended in [363], we used 5 replications of 2-fold cross-validation (5x2cv) paired $t$ test applied on Bug-saalle dataset.

For each replication, we randomly shuffled the data. However, we trained and tested SVM and DGD with respectively the same training and testing data.

Fig 6.12 shows the box plots of the accuracy of our trained algorithm using 5x2cv. It is noticeable that DGD (median value: 0.976) outperforms SVM (median value : 0.965). This assessment is confirmed by a p-value of 0.00028. The fact that the input of DGD (images pair) is not the same as for SVM (pair of coordinates) could explain such results. Indeed, by inputting the lamp's left and right view, DGD can determine the real center of gravity even if it does not follow a Gaussian distribution. DGD can use the neighborhood to predict if a point is on the ground or not. However, this gain of accuracy (0.976 vs. 0.965) seems a bit marginal relative to the implementation complexity of DGD.

Our method can detect whether a point is on the ground or not with an accuracy of 0.97. It is an efficient way to estimate the ground implicitly on a very low-resolution image. And our method does not need any camera calibration or complicated learning process. We just needed to move a hot bulb in a room.

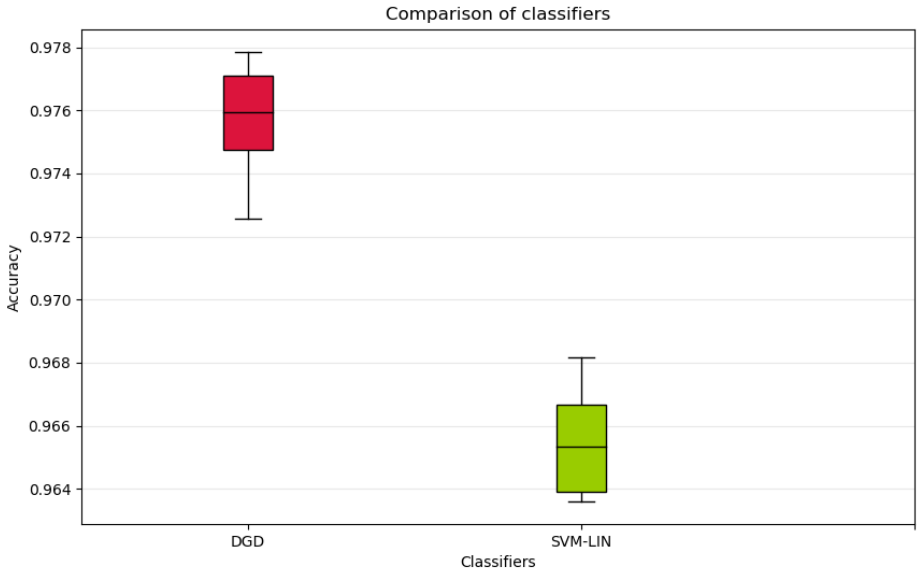
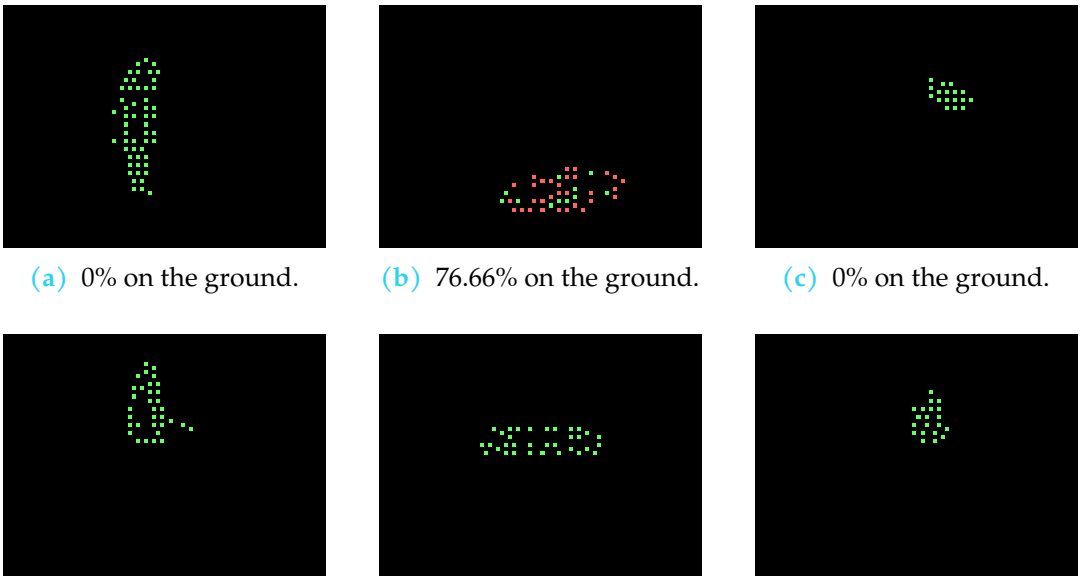


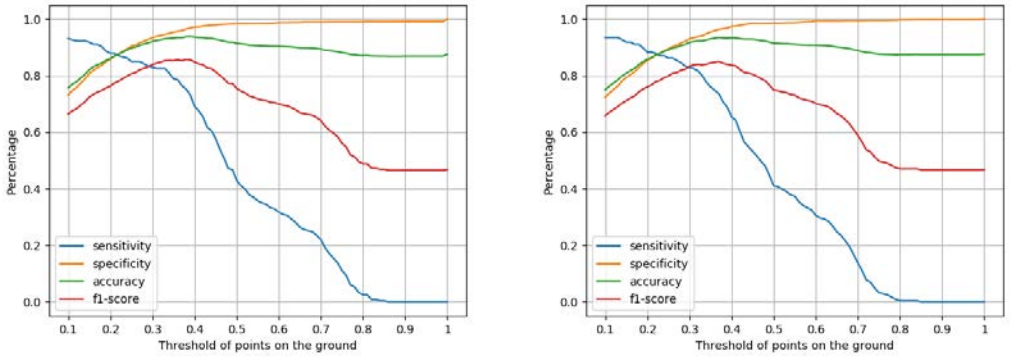

**Figure 6.12:** Boxplots of the accuracy measured using 5x2-fold cross-validation (5x2cv) on Bug-saalle dataset. Comparison between DGD and SVM.



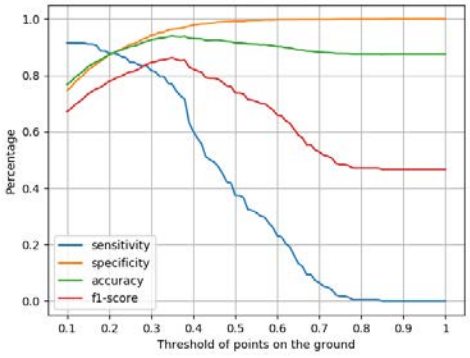
(a) 0% on the ground. (b) 76.66% on the ground. (c) 0% on the ground.

(d) 0% on the ground (e) 0% on the ground. (f) 0% on the ground.

**Figure 6.13:** Outputs of $TSFD$ on the images shown in Fig 6.11. Green points represent points top of the ground while Red points represent points which are classified as on the ground. (*best viewed in colors*)




(a) $\gamma = 0.01$, the maximum value is at $T = 0.38$ (b) $\gamma = 0.1$, the maximum value is at $T = 0.37$




(c) $\gamma = 0.3$, the maximum value is at $T = 0.35$

**Figure 6.14:** Variation of the values of sensitivity, specificity, F1-score and accuracy according to the threshold $\gamma$ using $DGD$ on Person0. (*best viewed in colors*)

### 6.5.5 Proof of the concept of the fall detection framework

The Fig 6.13 shows some results of $TSFD$ when applied to the selected images displayed in the Fig 6.11. On these images, we can qualitatively notice that a person lying on the floor (Fig 6.13b) has more features classified to be on the ground than the other persons' pose. We were also pleasantly surprised by the that $TSFD$ were able discriminate a person who is lying top of the ground (Fig 6.13e) with a person who is lying on the ground (Fig 6.13b). Such cases are usually difficult to discriminate on methods based on the posture of the people.

These qualitative results strengthened us in the idea that TSFD can detect falls. In the first stage, we tuned $TSFD$ to find the optimal parameters set. We then evaluated how these parameters set tuned on one person can be inferred to detect falls for other people.

#### 6.5.5.1 Parameter tuning

Two parameters affect the performance of $TSFD$:

1. $\gamma$: the phase congruency threshold. A low $\gamma$ results in a higher amount of features to match but at the expense of matches robustness. We had to determine the $\gamma$ to have a good balance between the number of features and robustness.
2. $T$: the threshold on the percentage of points on the ground. Intuitively, a low threshold increases the number of false-positive rates while a higher threshold will increase the false-negative rate.

The performance will be assessed in terms of sensitivity, specificity, accuracy, and F1-score. For this, we used the Tuulmen Puiir dataset. After having trained the DGD point classifier on the Tuulmen Puiir-points image dataset, we tuned the fall detection method on the first sub-dataset Person0. For the tuning, we choose three values of $\gamma \in \{0.01, 0.1, 0.3\}$ and we vary the value of $T$ from 0.1 to 1 by a step of 0.1. For each value of $\gamma$, Fig .6.14 displays the variations of sensitivity, specificity, accuracy and F1-score according to $T$. We can notice at a glance that $\gamma$ has only a little influence on these scores. On the contrary, $T$ directly influences the metrics values.

By using small $T$, we will get higher values of sensitivity at the expense of specificity, accuracy, and F1-score, and for a higher value of $T$, we will get higher specificity at the expense of sensitivity and F1-score. But, we can notice that for $T$ in a range of 0.3 and 0.4 we have a good trade-off between the scores. This behavior is directly visible on the F1-score which presents a global maximum in this range.

The idea is now to use this behavior to estimate an optimal threshold of $T_{pred}$. This optimal value can be determined by picking the threshold, giving the highest value of F1-score. We chose the F1-score because it is a balanced measure between the

precision and the recall. It is also a common measure when the dataset is imbalanced as for ours, as shown in [364]. Depending on the value of $\gamma$ the optimal value $T_{pred}$ is respectively 0.38, 0.37 and 0.35 for $\gamma \in \{0.01, 0.1, 0.3\}$.

#### 6.5.5.2 Inference to another person

In this experiment, we tried to verify whether a threshold $T_{pred}$ learned on one person could be inferred to another person. For this, we applied the $T_{pred}$ learned on Person0 (0.38, 0.37 and 0.35 for respectively $\gamma \in \{0.01, 0.1, 0.3\}$) to Person1, Person2 and Person3. The results were compared to those obtained using an ad hoc threshold of $T_{ad} = 0.25$. We intuitively chose 0.25, considering that if a quarter of a person is on the ground, we can say that he is lying on the ground.

Table 6.2 presents the sensitivity, specificity, F1-score and accuracy for each $T \in \{T_{pred}, T_{ad}\} \times \gamma \in \{0.01, 0.1, 0.3\}$. On this table we can see that $T_{pred}$ does not always outperform $T_{ad}$. Such a result shows that $T_{pred}$ is data dependent.

We went further by computing the optimal threshold for each sub-dataset (Person1, Person2, Person3). The results are displayed in Table 6.3. We can notice that the optimal threshold for Person3 is very different from Person0, Person1, and Person2. On the opposite, Person1 and Person0 optimal thresholds are very close. The optimal value of $T$ depends on the cameras' position and the height of the person.

For a real-world application, we cannot directly learn the threshold on a senior. So, either we use an ad hoc threshold, or we can build an atlas of thresholds depending on the height and the morphology of different people and pick the most suited threshold for a specific senior.

**Table 6.2:** Performance of $TSFD$ depending on the parameters $\gamma$ and $T$ for each dataset. In each line the best values (F1-score) are set in **bold**.

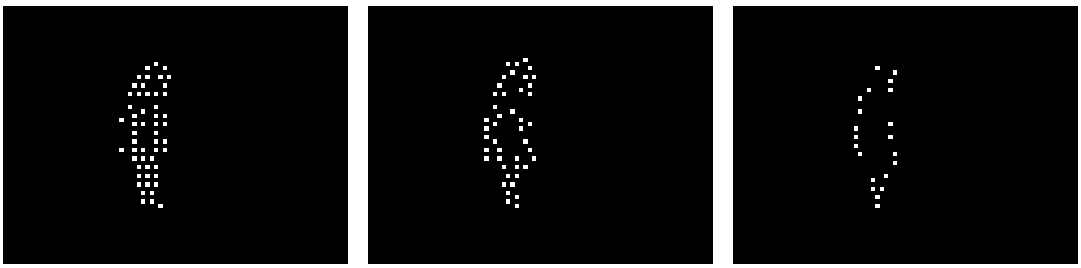
| datasets | Metric | 0.01 | | 0.1 | | 0.3 | |
|---|---|---|---|---|---|---|---|
| | | $T_{ad}$ | $T_{pred}$ | $T_{ad}$ | $T_{pred}$ | $T_{ad}$ | $T_{pred}$ |
| Person1 | Sensitivity | 0.94 | 0.91 | 0.91 | 0.88 | 0.87 | 0.77 |
| | Specificity | 0.97 | 0.99 | 0.95 | 0.98 | 0.91 | 0.96 |
| | F1-score | 0.92 | **0.96** | 0.89 | 0.93 | 0.81 | 0.85 |
| | Accuracy | 0.96 | 0.98 | 0.95 | 0.97 | 0.91 | 0.94 |
| Person2 | Sensitivity | 0.97 | 0.79 | 0.97 | 0.79 | 0.83 | 0.57 |
| | Specificity | 0.97 | 0.98 | 0.94 | 0.98 | 0.96 | 0.98 |
| | F1-score | **0.93** | 0.91 | 0.90 | 0.91 | 0.88 | 0.81 |
| | Accuracy | 0.97 | 0.96 | 0.95 | 0.95 | 0.94 | 0.92 |
| Person3 | Sensitivity | 0.89 | 0.51 | 0.91 | 0.57 | 0.73 | 0.58 |
| | Specificity | 0.98 | 0.99 | 0.94 | 0.98 | 0.96 | 0.98 |
| | F1-score | **0.94** | 0.80 | 0.90 | 0.81 | 0.87 | 0.82 |
| | Accuracy | 0.96 | 0.88 | 0.93 | 0.89 | 0.91 | 0.89 |

Table 6.3: The optimal threshold according to the sub-dataset and $\gamma$

| | 0.01 | 0.1 | 0.3 |
|---|---|---|---|
| Person0 | 0.38 | 0.37 | 0.35 |
| Person1 | 0.35 | 0.34 | 0.33 |
| Person2 | 0.30 | 0.30 | 0.24 |
| Person3 | 0.21 | 0.22 | 0.23 |

#### 6.5.5.3 Influence of $\gamma$

Table 6.2 also allows us to analyze the influence of $\gamma$. If we look specifically on the F1-score, $\gamma = 0.01$ always provided the best results. Such a result can be explained by the fact that $\gamma = 0.01$ provides more matches which are intended to represent (more or less) the human shape. Fig 6.15 represents the projected matches for the image pair of Fig 6.11a using respectively (a) $\gamma = 0.01$, (b) $\gamma = 0.1$ and (c) $\gamma = 0.3$. It is noticeable that $\gamma = 0.01$ outputs more matches than $\gamma = 0.1$ and $\gamma = 0.3$. We even got 3 times more matches with $\gamma = 0.01$ than with $\gamma = 0.3$. By linking this figure with Table 6.2, we can understand that the F1-score is directly related to the number of matches. $\gamma = 0.3$ always gives the worst results in therms of accuracy and F1-score whatever $T$. Indeed using $\gamma = 0.3$ the number of matches is not sufficient to well describe the whole human shape (Fig 6.15c). So, $\gamma = 0.3$ should be avoided.



(a) 53 matches for $\gamma = 0.01$ (b) 43 matches for $\gamma = 0.1$ (c) 20 matches for $\gamma = 0.3$

Figure 6.15: Example of matches using different values of $\gamma$ of the image pair Fig 6.11a

We went further in comparing the influence of $\gamma$ by estimating the mean computation time of the stereo matching process. We launched the stereo matching process 1000 times on the image used in Fig 6.15. The execution was performed on an $8 \times$ Intel Xeon CPU E5620 2.4 GHz. Table 6.4 shows the mean value of the execution time for feature extraction and stereo matching.

We already noticed in Chapter 4 that the computation was the same whatever gamma. However, the value of gamma has a direct impact on the number of extracted features. A high number of features makes the stereo matching process longer and more complicated and directly increases the computation time.

Regarding computation time, $\gamma = 0.3$ (1.25 ms) is 7 times faster than $\gamma = 0.1$ (7.73 ms) which is 6 times faster than $\gamma = 0.01$ (41.17 ms). If we took a closer look at $\gamma = 0.01$ and $\gamma = 0.1$ we do not have the same order of magnitude regarding the number of matches and the computation time (41.17 ms vs. 7.73 ms and 53 matches vs. 43 matches). $\gamma = 0.01$ features are denser than with $\gamma = 0.1$, leading to higher complexity of the matching process.

**Table 6.4:** Execution time

| | | ≈ Execution time |
|---|---|---|
| Features extraction | $\gamma \in \{0.01, 0.1, 0.3\}$ | $13.32 \pm 1$ ms |
| Stereo matching | $\gamma = 0.01$ | $41.17 \pm 3.5$ ms |
| | $\gamma = 0.1$ | $7.73 \pm 0.67$ ms |
| | $\gamma = 0.3$ | $1.25 \pm 0.02$ ms |

To summarize, $\gamma = 0.01$ gave the highest scores at the expense of the computation time. However, if the computation time is an issue, $\gamma = 0.1$ is a trade-off between speed and fall detection efficiency.

### 6.5.6 Comparison with methods in state of the art

We tried to compare our scores to those reported in the literature. The methods presented in [21] and [22] are the closest to our method. Both methods try to detect the fall (without classifying other human activities) on low thermal cameras with deep learning without detecting the ground. But contrary to our method, both use a monocular high-resolution thermal image. Moreover [22] uses 8 frames instead of 1 (Table 6.5). For comparison, we computed on the whole images of Person1, Person2, and Person3 the same scores as reported in [21] (accuracy) and [22] (Area Under the Curve of the ROC curve).

As noticeable in Table 6.5, our method seems to be very competitive with better accuracy and a higher AUC as reported in [21, 22]. Contrary to the other methods, we are using stereo-vision, which helps us better learn the ground position by leading the network to triangulate. For example, this allowed us to discriminate a person on the ground (after a fall) from a person lying at a certain height above the ground (sleeping on a couch). Our method's main disadvantage is that we have to relearn the point classifier for each position of the stereo-system. But the process of learning the ground position is very fast. Indeed, taking bulb images roughly takes no more than 30 min when the training process takes no more than 1 hour for $DGD$ and a few minutes for SVM.

Moreover, contrary to other methods, ours is not based on explicit learning of fall cases. Most of the datasets containing falls are imbalanced, which is a real issue in machine learning [29].

| Methods | Sensors | Camera position dependent | Algorithms | ROC AUC | Accuracy | Resolution | Frames |
|---|---|---|---|---|---|---|---|
| [21] | Mono | no | CNN (3 layers ) | | 0.857 | 32x31 | 1 |
| [22] | Mono | no | CNN LSTM | 0.83 | | 64x64 | 8 |
| Ours | Stereo | yes | CNN + Threshold | 0.94 | 0.94 | 80x60 | 1 |

**Table 6.5:** Comparison with results in the state of the art (Unfair comparison)

## 6.6 Conclusion

In this chapter, we presented a framework to perform fall detection using very low-resolution stereo thermal images. The framework $TSFD$ is composed of three main parts: stereo-matching, points classification, and threshold-based fall detection. This framework does not require any prior calibration of the cameras. Though based on deep learning, our method does not need any training on explicit falls.

Our method's key point is the points classification step, which consists of learning if a feature is on the ground or not. To handle this step, we proposed a linear SVM method or a network $DGD$ witch we had derived from DenseNet. Based on the classification, whether a point is on the ground or not, our approach determined the percentage of an object on the ground to detect fall or not fall.

Even if $TSFD$ gave better results than state of the art approaches, taking into account, the sequence dynamic could improve these results. So we need to acquire more data for many subjects and in a domestic context.

# 7 Activity recognition

He got the action, he got the motion
Yeah, the boy can play
Dedication, devotion
Turning all the night time into the day

Dire Straits

## 7.1 Introduction

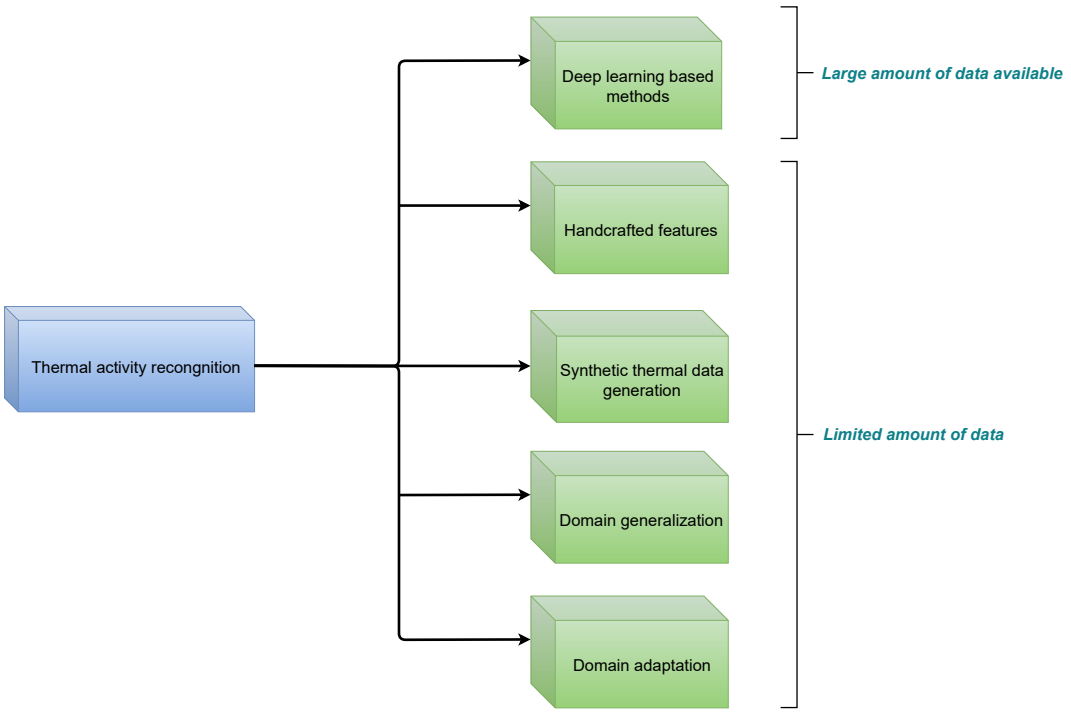



**Figure 7.1:** Human activity recognition methods

In the previous chapters, we addressed fall detection through static analysis of image pair. Such a framework is useful to detect falls, but it is essential to consider many frames and their temporal relationship for a better analysis of senior activities. For example, as we discussed in chapter 1, it is essential to monitor senior activities rather than only detect falls.

Activity monitoring allows to not only detect falls but also to prevent them. Indeed, some hints can help to predict a fall through a fine-grained analysis. This is

also important for frailty detection. Besides, activity recognition could be essential to differentiate falls. All type of falls does not have the same consequences and implications. Moreover, some activities, such as hand pick-up, can prove that the senior is healthy.

We proposed handcrafted features that could deal with few data. But recently, deep learning methods show many improvements to some extent. This is why we chose to take a glance at this group of methods. But we faced an issue linked to the amount of dataset we had. Fig 7.1 depicts many strategies to recognize activity for thermal images. When there is a huge amount of dataset, modern deep learning methods can be used. In the literature, it has been proved that such methods can get outstanding results. But sometimes, in certain domains such as medical imaging or, in our case, thermal activity recognition, the amount of dataset is limited.

This chapter will consider ways to perform thermal-based activity recognition without enough dataset. One solution is to find a strategy to learn from visible images and then *transfer* this *knowledge* to thermal images. Indeed, hopefully, there is a huge amount of dataset of visible images for activity recognition. Even if there are some privacy issues with these devices, they are easily affordable than thermal cameras. For example, every modern smartphone incorporates a camera with a pretty good resolution.

The first strategy we tried concerns edges. While visible and thermal images represent different electromagnetic wavelengths but by computing the edges, we could represent as well as in the thermal or visible image by the *same image*. Our idea is that if we could learn activity from the motion of a given shape, it could be possible to *transfer* this *knowledge* to another shape, even if the two shapes come from different images modalities. We noted this strategy as *edge* strategy given that we first train on visible edges maps and infer using thermal edges maps without any adaptation.

Our second strategy involves simulated thermal images. It has been shown in many works that it is possible to enhance the neural network's generalization capability by data augmentation and training on simulated dataset[365, 366]. Thus, our main idea is to train a GAN to generate simulated thermal videos from visible videos. Then the activity recognition model is trained with simulated thermal videos and inferred using real clips.

The third strategy is similar to the first one, but rather than computing edges maps; we computed optical flow. Thus, the model is trained on visible image optical flow and inferred using thermal image optical flow without any adaptation.

In the last strategy, we tried to use a set of methods that have been constructed for training for a domain and inferring for another domain. This class of method is called domain generalization. These methods are similar to domain adaptation. In

domain generalization and our first three strategies, the model has only access to the source domain and must generalize well to an unseen (target) domain.

This chapter is structured as follows. Section 7.2 will exhibit the main contributions for activity recognition both for handcrafted methods as for deep neural networks , deep generative models for simulated video generation and domain generalization. Section 7.3 details ours approaches and strategies while Section 7.4 shows the datasets we used for simulated thermal data generation and activity recognition dataset. Section 7.5 shows some preliminary but yet promising results and Section 7.7 concludes the chapter and presents some perspectives.

## 7.2 State of the art

We have tackled many domains to recognize activities: activity recognition and simulated data generation, and domain generalization. This is why this section is divided into three categories.

### 7.2.1 Activity recognition

Activity recognition is widely used in many applications such as robotics [367], medicine [368], visual surveillance [369] and senior monitoring [370].

#### 7.2.1.1 Is an activity a set of activities?

An activity can be recognized as an activity as such or a combination of sub-activities or sub-events (Fig 7.2). Depending on this view-point, the activity recognition algorithm can be classified as single-layered or hierarchical.

**Single layered approaches** These approaches are trying to treat raw video data to recognized activities. They are trying to recognize primitive actions in order to detect more complex ones. Many methods use a sliding window to classify sequences. They are designed to classify short sequences such as walking, jumping, waving or clapping.

There are two types of single layered approaches:

(a) **Space-time:** These methods use both the X and Y dimension of the image with an extra dimension coming from time. Here an action is considered as a 3D $XYT$ space-time volume with 2D images stacked along time dimension (Fig 7.3). In these methods, each 3D volume represents an activity or a combination of activities. A model is then trained on these videos, and during testing, each new video is compared to the learned videos (represented by the trained model). These methods can be divided into three categories:

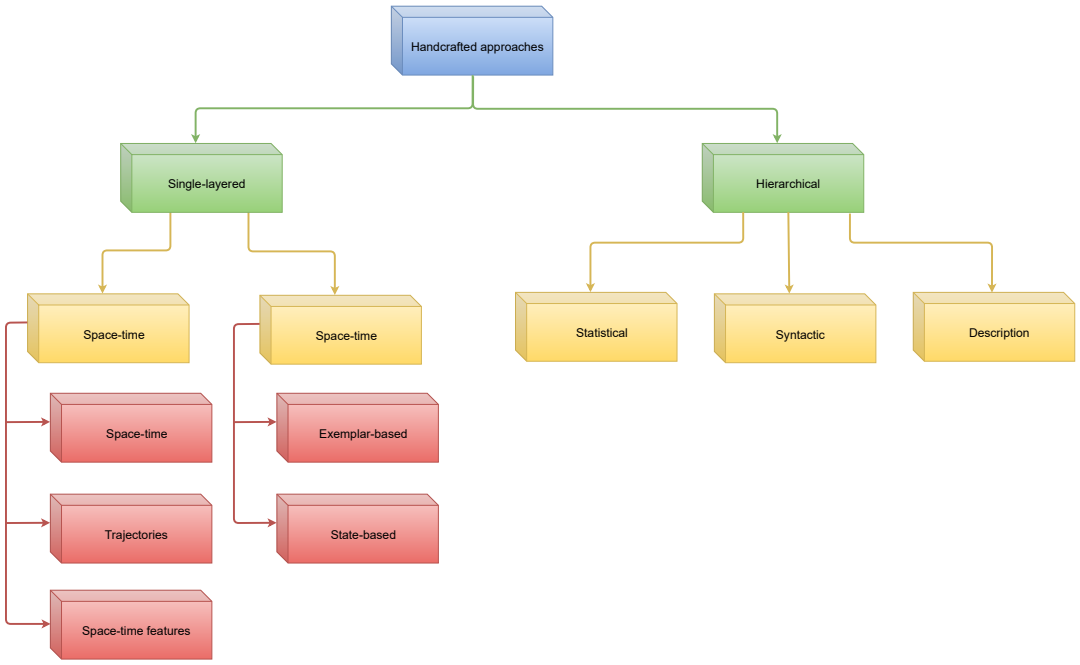



**Figure 7.2:** Handcrafted-based approaches for activity recognition

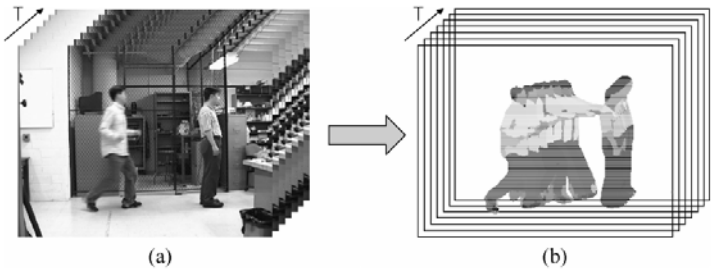



**Figure 7.3:** Example XYT volumes constructed by concatenating (a) entire images and (b) foreground blob images obtained from a punching sequence. (from [371])

(i) **Space-time volume**: Each activity can be represented either a trajectory (through the $XYT$ volume or the other dimensions) or a set of features extracted from these volumes or trajectories.

Regarding the method of recognizing activities through space-time volumes, they are trying to establish a similarity measurement between two volumes. This similarity is computed most of the time through matching.

In [372], Hu et al. propose a novel framework named SMILE-SVM for Simulated annealing Mul-tiple Instance LEarning Support Vector Machines. Their method uses motion history image feature, foreground image feature, and histogram of oriented gradients. In [373], Qian et al. use a similar idea but by combining global features and local features. They classify activities through a three-fold pipeline by firstly detecting people using background

subtraction, secondly extracting features, and finally training a multi-class SVM.

The main drawback of these approaches is that they are not working well when there is more than one person in the field of view and when the activity is spatially segmented. Besides, they are using a traditional sliding-window algorithm, which is computationally expensive. Finally, most of the time, these methods are not view-point invariant and cannot handle correctly speed and motion variation.

(ii) **Trajectories**: These methods take an activity as a set of space-time trajectories. A human is mostly represented by a set of 2D $(x, y)$ or 3D $(x, y, z)$ points in these methods. In [374], Johansson shows that a set of tracking joint positions can represent human activity.

In [375], Sheikh et al. propose a method using 13 joints trajectories in a 4D space $(x, y, z, t)$. They try to take into account of intra-class variability such view-point, speed and anthropometry of actors. An action $A$ can be written as $\mathbf{A} = [\mathbf{X}_1, \cdots, \mathbf{X}_p]$ where $\mathbf{X}_i = (X_{T_i}, Y_{T_i}, Z_{T_i}, T_i)^t$ and $p = 13n$ for n frames. Then an activity can be constructed by the linear combination of $\mathbf{A}_i$'s actions.

In [376], Yilmaz Alper and Shah Mubarak propose a view-point-free activity recognition method. They assume that two instances of the same activity performed from different view-points and different persons should have similar trajectories. They propose to locally compensate motion using the fundamental matrix estimated for each frame.

The main advantage of these approaches is that human activities can be analyzed in a fine-grained manner. But their performance depends a lot on the accuracy of extracted joints.

(iii) **Space-time features**: The principles of these methods are close to trajectories-based ones. But instead, to extract joints, they extract local features. Most of the time, local features are extracted from each frame, then they are matched together over time. This approach is facilitated by the fact that feature extraction has been widely studied in the literature.

In [377], Chomat Olivier and Crowley James L propose using local appearance descriptors to characterize an activity. They extract space-time features using Gabor filters, then they construct histograms of these features and finally apply Bayes rules.

In [378], Gorelick et al. calculate appearance-based local features by analyzing 2D shapes. They extract features from the solution to the Poisson equation. The combination of these local features forms a global space-time feature. They then use the nearest neighbor method to classify activities.

Other authors show that sparse local features can also be used for classification.

In [379], Laptev Ivan extend Harris corner detector to 3D volume. Given that this latter does not output features for certain frames, in [380] Dollár et al. propose an approach using cuboids of Spatio-temporal bounding boxes around each feature. Their feature detector is based on 2D Gaussian smoothing kernels on spatial dimension and 1D Gabor filters on the time dimension. Each extremum represents a feature around which they build a cuboid over time dimension. For each cuboid, they normalize values, compute the brightness gradient and the windowed optical flow. Finally, they construct a histogram of cuboids and classify activities by clustering these histograms through K-means. Many works in the literature are based on Dollar detector and Harris 3D detector.

In [381], Holte et al. propose a method based on optical flow to detect activities. Given a multi-view video, they first compute the optical flow for each view. Then, they fuse the optical flow using 3D Motion Context (3D-MC) and the view-invariant Harmonic Motion Context (HMC). Finally, these features are fused for action classification using normalized correlation.

Other authors use some classical features extractor. In [382], Klaser et al. propose to recognize activities using 3D histograms of oriented gradients (HOG). They extend HOG by computing 3D quantized orientations on space-time volumes.

In [383], Laptev et al. improved they method described previously [379] by concatenating normalized HOG and optical flow on the extracted features. They use this multichannel information in a non-linear SVM as an activity classifier.

In [384], Yu et al. extend FAST corners into a new spatiotemporal interest point called Video FAST (V-FAST). They are detecting interest points throughout the spatial place $(X, Y)$, and the temporal planes $(X, T)$ and $(Y, T)$. V-FAST features are detected in the same way as in FAST. So given a video, they extract spatiotemporal volumes. From these spatiotemporal volumes, they construct spatiotemporal semantic Texton Forests, which are based on Random Forest. They combine Pyramidal Spatiotemporal Relationship Matches with Bag of Semantic Textons (BOST) to classify video.

In [385], Wang Heng and Schmid Cordelia propose the most efficient handcrafted method currently available. They propose a new feature called Improve Dense Trajectories (IDT). First, they extract SURF features and match them in consecutive frames. Then, they compute motion vectors by estimating optical flow. They remove camera motion by rectifying the frames through the homography using RANSAC. This allows them to remove trajectories caused by camera motion. Given a video, they extract

trajectory, histograms of oriented gradients, and optical flow and motion boundary histograms [386]. The histograms-based descriptors are then normalized using RootSIFT [387]. Finally, they use a non-linear SVM to classify these descriptors converted to a bag of features and Fisher vector. Given that these methods are based on local features, they are scale-, rotation-, brightness-, and translation-invariant. Indeed, local features have received a lot of attention in image classification literature.

(b) **Sequential:** These methods analyze videos as sequences of features to recognize activities. From each image, they extract features; then each video is represented by a sequence of features. Sequential methods can be divided into two categories:

(i) **Exemplar-based approaches** learn a set of features for each activity. Then during testing, each video is compared with the computed features, and similarity is measured. But a given activity can be played in many ways; these models are built to be flexible. Most of the time, the similarity is measured using Dynamic Time Warping (DTW). In [388], Yacoob Yaser and Black Michael J apply PCA to each sequence to extract a set of eigenvectors per video. They use DTW to match sequences. While this method works at the gesture level, in [389], Veeraraghavan et al. propose an activity recognition method targeting action-level activities. Their method is based on frames where the background has been subtracted. They consider that the same class activity can vary in speed in their approach. Each activity is represented by a nominal activity trajectory and a function space of all possible time warping. The nominal activity trajectory of two different activities should be different. Moreover, each activity is warped in an interval and is constituted of units.

(ii) **State model-based methods** consider an activity statistically as a sequence of states. Most of the time, to build models, Hidden Markov Models (HMMs) or dynamic Bayesians networks are used. In these models, given a video, each frame represents a state, and a feature vector represents each state. Sequences of states and transitions represent an activity. By training models on the transitions, it is possible to recognize activities by solving *evaluation problem* given a new video. In [390] Yamato et al. are the first to propose an activity recognition method using Hidden Markov Models. They apply their method on background-subtracted frames, and their method is working at the action-level. Hidden Markov Models are a Markov process with unknown parameters. They represent each class by an HMM. Then the parameters of each model are estimated by training it on a labeled dataset. One of the main drawbacks of HHM is that they can only learn activities where only one person is performing. To take into account multi-person actions, in [391], Oliver et al. propose to use coupled Hidden Markov

Models (CHMM) to recognize activities involving more than one person. Most of the time, each HMM represents an agent but there some dependencies between the two HMMs. To improve HMMs and CHMMs, in [392], Brand et al. propose coupled Hidden semi Markov Models where each state has each own duration, unlike in HMMs and CHMMs.

Unlike space-time methods, sequential methods take into account relationships between features. On the one hand, exemplar-based approaches are more flexible than state-model approaches. Moreover, using DTW, they are able to compare non-linearly two activities. Finally, they require less training data compared to state-model approaches. On the other hand, given that they are probabilistic, state-model approaches require a lot of datasets to be trained. But this probabilistic aspect can also be used for other processes.

**Hierarchical approaches** Hierarchical approaches are based on the principle that an activity is composed of a combination of sub-activities (sub-events) and so on. In most of the methods, sub-events are recognized using a single-layered approach. They can be divided into three categories: statistical analysis, syntactic and description-based approaches.

(i) **Statistical analysis approaches:** These methods are based on single-layered state-based methods.

In [393], Oliver et al. propose a layered Hidden Markov Model to recognize activities hierarchically. They construct a three layers model. The first layer detects atomic actions when the second layer detects their combinations, and so on. In a real-world activity, sub-events can be concurrent or sequential. This is why in [394], Shi et al. propose a propagation network taking into account concurrent and sequential sub-events. Their method is based on DBN (Dynamic Bayesian Network).

Given that they are based on state model methods, statistical analysis approaches also the same disadvantages. Indeed, they require a huge amount of dataset for training. Besides, they are not suitable to represent complex and no sequential activities.

(ii) **Syntactic approaches:** In these methods, sub-events must be first recognized, and then they are considered as *strings*. Most of the time, Context-Free Grammars (CFGs) or Stochastic Context-Free Grammars (SCFGs) are used to recognize the combination of atomic actions. In [395], Ivanov Yuri A. and Bobick Aaron F propose a probabilistic parser to identify activities.

The main drawback of syntactic approaches is that they struggle to learn concurrent activities. Besides, strings represent high-level activities that must be most of the time strictly sequential. Finally, these methods assume

that the activities follow the production rules a user provides. That is not always the case. Thus, they are not able to generalize well.

(iii) **Description-based approaches:** These methods try to describe an activity by analyzing temporally, spatially, and logically sub-events. For this, many authors propose to use Allen temporal predicates [396, 397]. These predicates define 13 relations between two intervals $X$ and $Y$. For example, $X$ *takes place before* $Y$ , $X$ *overlaps with* $Y$ etc. Most of the time, Context-Free Grammars (CFGs) are used. In [398], Nevatia et al. define a formal syntax using ontology to describe an activity. The main idea is to convert Allen predicates to something intelligible for a computer program. Such conversion can also be achieved by Petri nets [399] or event logic [400].

The main advantage of description-based methods is that they can represent and recognize complex human activities. Such representation can be very rich with sequential and concurrent sub-events. Unfortunately, they struggle to represent low-level activities such as gestures, even if there are some studies in this way.

Recently methods based on deep learning have achieved outstanding performance in activity recognition as in image classification. It is possible to classify these methods according to the used architecture, the type of input (Raw RGB, Dept, Optical flow), the loss function, etc. One typical issue of these methods is how they are dealing with RGB streams and Optical flow stream. Usually, some authors propose a model for only one of these modalities, a model for each of these modalities, or a single model for all these modalities.

In [401], Taylor et al. propose to use a Restricted Boltzmann Machine to extract Spatio-temporal features. Their architecture is shallow and is composed of a latent feature layer and a pooling layer. In [402], Ji et al. apply 3D convolutions to RGB videos. The convolution is applied to spatial and time dimensions. For regularization purpose, they add an auxiliary feature extractor. In [403], Wang et al. propose Trajectory-pooled Deep-convolutional Descriptor (TDD) using improved dense trajectories (IDT) [385] and CNN features [404]. First, they extract features from a trained version of the two-stream architecture proposed by Simonyan Karen and Zisserman Andrew. They then extracted trajectories from RGB videos using the method proposed by Wang Heng and Schmid Cordelia in [385]. Finally, extracted trajectories and deep learning features are combined to extracted TDD descriptors and an SVM classify features.

In [404], Simonyan and Zisserman propose a two-stream architecture for video recognition. Their network is thus composed of a spatial stream convolution network and a temporal stream convolution network. While the input of the first sub-network is RGB images, the second sub-network input is optical flow frames. The two sub-networks have the same architecture five convolutions followed by full convolutions

and, finally, a softmax layer. Finally, the output (softmax scores) of the sub-networks are combined in an SVM [405].

Given that video has a temporal aspect, many authors propose models based on LSTM [406]. In [407], Donahue et al. present a Long-term Recurrent Convolutional Network (LRCN) based on Long Short Term Mermory (LSTM) by first extracting features from each frame. Then, these features are used in an LSTM architecture. They also use two inputs, RGB and optical flow, that predictions are fused in the late stage.

In [408], Wang et al. propose a two-stream architecture with a single frame for the spatial network and ten optical flow frames for the temporal network. They fine-tune the pre-trained model, and to avoid over-fitting, they propose new data augmentation techniques through cropping (four corners and one center) and multi-scale cropping.

In [409], Tran et al. propose a C3D, a 3D convolutional neural network with eight convolutions, five max-pooling, and two fully connected layers. The prediction is made with a softmax layer. Their 3D convolutional architecture is more efficient than the previous Spatio-temporal architectures. They show that the performance of their model is increased while fine-tuning a pre-trained model. In [410], Feichtenhofer et al. propose a two-stream architecture fusing at the temporal and the spatial network at a middle level. They use a convolutional layer for the fusion. In [411], Yue-Hei et al. evaluate many types of fusion. Like the models proposed in [409], their model performance increases when adding IDT.

In [412], Wang et al. propose Temporal Segment Network (TSN) to recognize activities in the long term. Their model is composed of Spatial and temporal sub-network. Their network takes as input snippets extracted from the whole video. A given video outputs $n$ snippets, then $n$ models are trained. The outputs of respectively, spatial and temporal, are fused through segmental consensuses. These consensuses are finally combined to compute a final score. They also test several kinds of inputs, such as optical flow, warped optical flow, RGB, and RGB difference. In [413], Bilen et al. propose a new way to represent video with dynamic images. Dynamic images of a given video represent all the frames through the same image using rank pooling.

In [414], Carreira and Zisserman also propose I3D a two-stream architecture based on the image classification inception architecture [415]. They compare their approach vs. three architectures: an LSTM, a 3D-ConvNet, and A Two-stream. Fig 7.4 shows the I3D architecture and Fig 7.5 shows the inception module. The two-stream model is composed of two I3D models trained separately on optical flow and RGB. Then, the predictions are fused in the final layer. For optical flow, they used the TV-L1 algorithm [416]. While their model is very accurate, the main drawback of I3D is the computation load, which does not allow him to work in real-time.

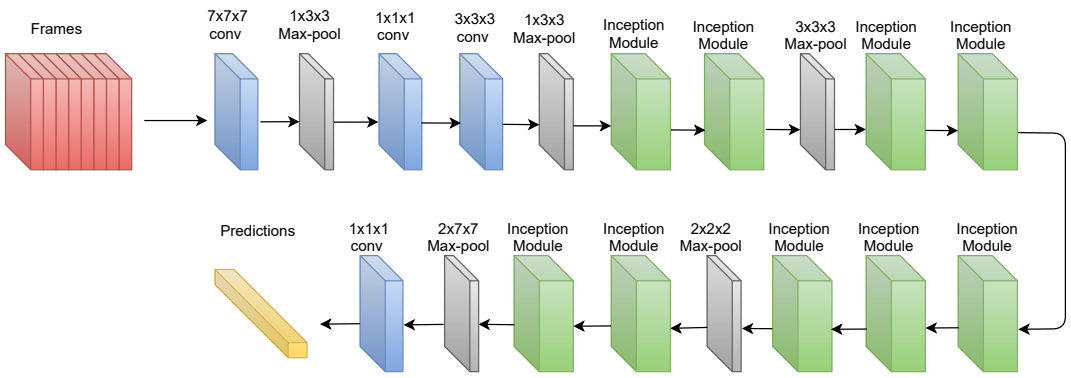



**Figure 7.4:** I3D architecture

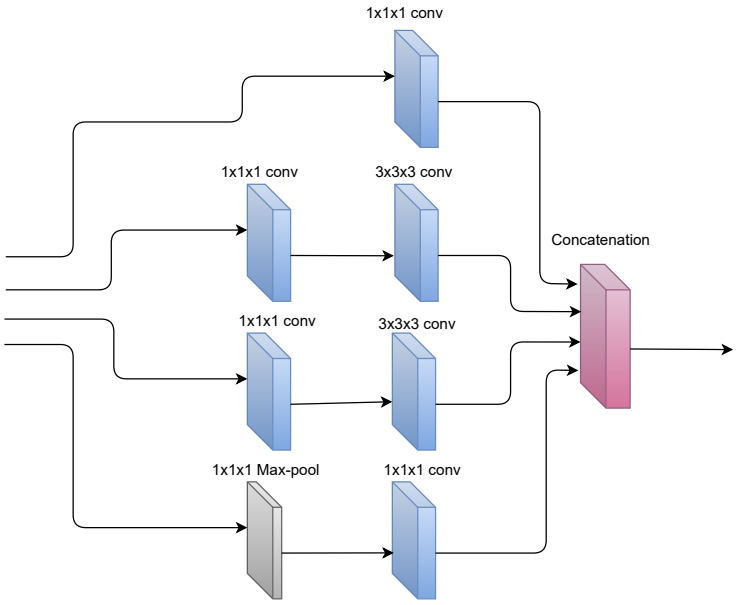



**Figure 7.5:** Inception module

In [417], Wang et al. propose a neural network that can classify videos which are weekly supervised (no temporal annotation). Their neural network is composed of three modules: *feature extraction*, *classification* and *selection*. They, first, extract features using the network proposed by [412] for the clips proposal. Then, for each clip proposal features, they apply a fully connected layer followed by a softmax layer. Finally, the predictions are combined. Their network, which is weakly supervised, gives competitive results compared to supervised approaches.

In [418], Wu et al. propose to train an architecture directly using compressed video (H.264, HEVC). Most of these codecs are based on three types of frames [419]: $I$-frames, $P$-frames, and $B$-frames. $I$-frames are regular frames while $P$-frames represent the change between current frames and previous ones. $P$-frames also contain motions vectors and $B$-frames, like $P$ frames, contain bidirectional motion vectors. For each $P$-frame, they compute the accumulated motion vectors and the accumulated residuals from this frame to the reference $I$-frame. They use ResNet-152 for $I$-frames and ResNet-18 for accumulated motion vectors and accumulated residuals. They also use the approach of temporal segments proposed by Wang et al. in [412] for long-term video classification.

Most of the time, it isn't easy to find the right architecture. A neural architecture search can tackle such a problem. In [420], Piergiovanni et al. propose a hybrid architecture based on inflated Temporal Gaussian Mixture (iTGM). The iTGM is based on the 1D Temporal Gaussian Mixture (TGM) proposed in [421]. Some parameters of their ResNet are fixed while others evolve; thus, the search space is limited. In [422] Ryoo et al. propose a similar approach by evolving a multi-stream neural network. Their network is composed of a convolution block by alternating 2D and (2+1)D residual modules.

In [423], Feichtenhofer et al. propose SlowFast, a neural network composed of two paths: slow patch way with low frame rate and fast pathway with a fast frame rate. The slow pathway uses a temporal stride $\tau$, thus its input is composed of all $n\tau$ frames with $n \in \{0, 1, \cdots\}$ . The fast pathway input is $\alpha$ times denser than the slow pathway frame-rate. Throughout the network, there are some lateral connections between the two pathways.

In [424] Wang et al. also propose an approach combining handcrafted IDT and deep features to detect activities. They show that Bag of Words and Fisher vectors that encode IDT features can be learned using a neural network. Moreover, this neural network is able to translate I3D RGB features to I3D optical flow features. Their network is composed of four sub-networks predicting, given the I3D RGB features, the first- and second-order Fisher Vectors, the Bag-of-Words, and High Abstraction Features. Each sub-network is composed of 2D convolutions or Fully connected layers and ends with a Power Normalization layer. To train the Fisher Vectors and Bag-of-Words

sub-networks, they computed ground truth values by extracting IDT features. These sub-networks output are then fused and to feed a Prediction Network (PredNet). They show that their network improves state-of-the-art performance on some benchmark datasets.

While I3D based approaches are accurate but computationally costly, in [425], Tran et al. propose to separate spatial and temporal components in a new spatiotemporal convolutional block named R(2+1)D. Like in [420], Tran et al. use a 2D convolution for spatial dimension and a 1D convolution for temporal dimension. They show that this architecture, based on Factorized Spatio-temporal convolutional Networks [426], is easier to optimize than 3D convolutions. Technically they replace $\nu_i$ 3D convolutional filters of size $(\nu_{i-1} \times t \times d \times d)$ with $\mu_i$ (2+1)D blocks composed of $\mu_i$ 2D convolutional filters of size $(\nu_{i-1} \times 1 \times d \times d)$ and $\nu_i$ 1D convolutional filters of size $(\mu_{i-1} \times 1 \times d \times d)$. They choose $\mu_i = \left\lfloor \frac{td^2\nu_{i-1}\nu}{d^2\nu_{i-1}+t\nu_i} \right\rfloor$. They show that their ResNet-based approach beats many state-of-the-art methods with lower computational complexity. As a conclusion of this review of deep learning activity recognition methods, we believe that I3D and R(2+1)D architectures are the most suited for our problem.

### 7.2.2 Generative models and simulated data generation

Data generation is one of the hottest topics in deep learning-based methods for recent years. This class of methods circumvents one of the obstacles of deep learning: the lack of data.

A Generative Adversarial Network (GAN) is composed of two networks:

— a generator $G$ which must generate some simulated dataset as close as possible to real data

— a discriminator $D$ which must be able to distinguish real data and generated data

GAN is a minimax optimization problem. This problem can be formulated as follows:

$$\min_{G \in \mathcal{G}} \max_{D \in \mathcal{D}} V(G, D) \tag{7.1}$$

where $\mathcal{G}$ and $\mathcal{D}$ are respectively the generator and the discriminator functions sets and $V(G, D)$ is the objective. Let $p_z$, $p_r$ and $p_g$ be respectively a random probability distribution, the probability distribution of real data and the probability distribution of generated data. Given a distribution $z \approx p_z$, $G$ generates dataset with a probability distribution $p_g$. During training, the main goal is to make $p_g$ and $p_r$ as *similar* as

possible. The objective function is:

$$\min_G \max_D \mathbb{E}_{x\sim p_r} \log\left[D(x)\right] + \mathbb{E}_{z\sim p_z} \log\left[1 - D(G(z))\right] \tag{7.2}$$

The main drawback of GANs is that they are hard to train and difficult to evaluate. Depending on the type of dataset to generate, sometimes it is hard for the two players to reach a Nash equilibrium. And the generator may fail to learn when the real dataset distribution is too complex. Also, depending on the type of dataset, it can be relatively hard to assess the quality or the dissimilarity between real dataset distribution $p_r$ and generated dataset distribution $p_g$. Since the first work about GANs in 2014 [427], most of the works are focused on *improving training* or *developing of GANs for real-world applications*.

Regarding the improvements, GANs can be classified into two types: architecture-based and loss-based. In this section, we will focus on the generation of simulated images and videos.

#### 7.2.2.1 Image-to-Image translation

Let be $\mathcal{A}$ the source domain and $\mathcal{B}$ the target domain. In [428], Hertzmann et al. propose image-to-image translation using a non-parametric texture model. Regarding image to image translation, there two approaches: paired and unpaired image-to-image translation.

When paired samples are given for the domains $\mathcal{A}$ and $\mathcal{B}$, the translation can done in a supervised way. In [429], Isola et al. propose pix-2-pix an image-to-image translation model using conditional GAN. The loss function is as follows:

$$\mathcal{L}_{cGAN} = \mathbb{E}_{x,y}\left[\log D(x,y)\right] + \mathbb{E}_{x,z}\left[\log\left(1 - D(x, G(x,z))\right)\right] \tag{7.3}$$

Their model is based on the neural network proposed by [430]. The used generator is based on U-Net [296] with skip connections between $i$th layer and $n - i$ layer.

#### 7.2.2.2 Video-to-Video translation

While image-to-image translation approaches work well, they do not capture the dynamic and spatiotemporal relationship to translate a video from domain $\mathcal{A}$ to domain $\mathcal{B}$. Video-to-video synthesis has been applied in many computer vision fields like video super-resolution [289], video matting and blending [431] and video inpainting [432].

Unconditional video synthesis has also been studied in state of the art. This topic

is linked to our topic, even if these models use a random vector. In [433], Vondrick et al. propose Video GAN (VGAN) generating video from a random vector. In [434], Tulyakov et al. propose MoCoGAN, an unconditional recurrent GAN using motion and sub-spaces content. Video style transfer is also related to video-to-video synthesis.

In [435], Wand et al. propose GAN-based video-to-video synthesis using spatiotemporal adversarial objective. Let $\alpha_1^T = \{a_1, a_2, \cdots, a_T\}$ be a sequence of source video frames from domain $\mathcal{A}$. Let be $\beta_1^T = \{b_1, b_2, \cdots, b_T\}$ be a sequence of source video frames from domain $\mathcal{B}$. The main idea is to learn a mapping function between domains $\mathcal{A}$ and $\mathcal{B}$ that would be able to convert $\alpha_1^T$ to a sequence $\tilde{\beta}_1^T = \left\{\tilde{b}_1, \tilde{b}_2, \cdots, \tilde{b}_T\right\}$ so that the conditional distribution of $\beta_1^T$ given $\alpha_1^T$ as follows:

$$p(\tilde{\beta}_1^T | \alpha_1^T) = p(\beta_1^T | \alpha_1^T) \tag{7.4}$$

They train a generator $G$ as $\beta_1^T = G(\alpha_1^T)$ and optimize it using the minimax problem:

$$\max_D \min_G \mathbb{E}_{\beta_1^T, \alpha_1^T} \left[\log D(\beta_1^T, \alpha_1^T)\right] + \mathbb{E}_{\alpha_1^T} \left[\log \left(1 - D(G(\alpha_1^T), \beta_1^T)\right)\right] \tag{7.5}$$

Where $D$ is the discriminator.

In order to incorporate temporal relationship between frames, they make a Markov assumption:

$$p(\tilde{\beta}_1^T | \alpha_1^T) = \prod_{t=1}^{T} p\left(\tilde{b}_t | \tilde{\beta}_{t-L}^{t-1}, \alpha_{t-L}^{t}\right) \tag{7.6}$$

That is to say video frames can be generated according to:

— current frame $a_t$

— past $L$ source frames $\alpha_{t-L}^{t-1}$

— past $L$ generated frames $\tilde{\beta}_{t-L}^{t-1}$

They train a feed-forward network $F$ as $\tilde{b}_t = F\left(\tilde{\beta}_{t-L}^{t-1}, \alpha_{t-L}^{t-1}\right)$ to estimate the optical flow. Thus they add an optical prediction network $W$, an generator $H$ to hallucinate images and $M$ a mask prediction mask as follows:

$$F\left(\tilde{\beta}_{t-L}^{t-1}, \alpha_{t-L}^{t}\right) = (1 - \tilde{m}_t) \odot \tilde{w}_{t-1}(\tilde{b}_{t-1}) + \tilde{m}_t \odot \tilde{h}_t \tag{7.7}$$

where $\odot$ is the element-wise product and:

— $\tilde{w}_{t-1} = W(\tilde{\beta}_{t-L}^{t-1}, \alpha_{t-L}^{t})$
— $\tilde{h}_t = H(\tilde{\beta}_{t-L}^{t-1}, \alpha_{t-L}^{t})$
— $\tilde{m}_t = M(\tilde{\beta}_{t-L}^{t-1}, \alpha_{t-L}^{t})$

In their experiments, they set $L = 2$.

The final network goal is not only to synthesis video but also from an image. Thus they use two discriminators $D_I$ (conditional image discriminator) and $D_V$ (conditional video discriminator). While $D_I$ is conditioned by the image from domain $\mathcal{A}$, $D_V$ is conditioned by the optical flow from a list of frames from domain $\mathcal{A}$. That is to say $D_I$ is trained to output 1 for $(b_i, a_i)$ and 0 for $(\tilde{b}_i, a_i)$ while $D_V$ is trained to output 1 for $(\beta_{t-K}^{t-1}, w_{t-K}^{t-2})$ and 0 for $(\tilde{\beta}_{t-K}^{t-1}, w_{t-K}^{t-2})$ (with $K$ consecutive real frames).

They define two sampling operators for images and video as follows:

— image sampling operator $\phi_I$ with $\phi_i(\alpha_1^T, \beta_1^T) = (a_i, b_i)$ where $i \in [0, T]$
— video sampling operator $\phi_V$ with $\phi_V\left(w_1^{T-1}, \beta_1^T, \alpha_1^T\right) = \left(w_{i-K}^{i-2}, \beta_{i-K}^{i-1}, \alpha_{i-K}^{i-1}\right)$ where $i \in [K+1, T+1]$

The final objective of the network is as follows:

$$\min_F \left( \max_{D_I} \mathcal{L}_{\mathcal{I}}(F, D_I) + \max_{D_V} \mathcal{L}(F, D_V) \right) + \lambda_w \mathcal{L}_w(F) \tag{7.8}$$

where:

— $\mathcal{L}_{\mathcal{I}}$ is the conditional loss function defined in [429] defined as follows:

$$\mathbb{E}_{\phi_I(\beta_1^T, \alpha_1^T)} \left[ \log D_I(b_i, a_i) \right] + \mathbb{E}_{\phi_I(\tilde{b}_1^T, \alpha_1^T)} \left[ \log(1 - D_I(\tilde{b}_i, a_i)) \right] \tag{7.9}$$

— $\mathcal{L}_V$ is defined as follows:

$$\begin{aligned} &\mathbb{E}_{\phi_V\left(w_1^{T-1}, \beta_1^T, \alpha_1^T\right)} \left[ \log D_V\left(\beta_{i-K}^{i-1}, w_{i-K}^{i-2}\right) \right] \\ +&\mathbb{E}_{\phi_V\left(w_1^{T-1}, \tilde{\beta}_1^T, \alpha_1^T\right)} \left[ 1 - \log D_V\left(\tilde{\beta}_{i-K}^{i-1}, w_{i-K}^{i-2}\right) \right] \end{aligned} \tag{7.10}$$

— $\mathcal{L}_w$ is defined as follows:

$$\mathcal{L}_w = \frac{1}{T-1} \sum_{t=1}^{T-1} \left( \|\tilde{w}_t - w_t\|_1 + \|\tilde{w}_t(b_t) - b_{t+1}\|_1 \right) \tag{7.11}$$

We use this model to generate simulated data because, in their experiment, the authors get better results compared to pix2pix.

### 7.2.3 Domain generalization

One of the problems we face for thermal-based activity recognition is the absence of datasets. Original domain generalization is used to adapt a model trained with a dataset into another dataset. Domain generalization is linked to domain adaptation and few-shot learning. But while in these techniques, the model sees the target data during training, in domain generalization, the model does not see the target dataset during training. For example, domain generalization can be used for a certain model trained for image classification using a hospital dataset. Then, one wants to infer this model using images coming from other hospitals with different orientations.

The main reason why we want to apply domain generalization to our problem is that data acquisition takes time and annotation is time-consuming. Being able to train a model using RGB video and inferring on Thermal videos will save us time. Indeed, there is a huge number of datasets for RGB images, but thermal datasets are sparser.

Domain generalization can be tackled in many ways: robust feature space learning, model architectures designed to enable robustness to domain shift, and methods optimizing standards architectures to find a robust minimum.

A $j$th domain is defined as $\{(d_i, x_i, y_i)\}_{i=0}^{n} \sim (\mathcal{D}_j, \mathcal{X}, \mathcal{Y})$ where $\mathcal{X}$ and $\mathcal{Y}$ are the set of data inputs and labels. Generally we have $m$ sources domains. The main idea of domain generalization is to learn to classify a data from unseen domain $d_{us} \notin D_m$ with $D_m$ the set of available domain during training.

#### 7.2.3.1 Domain invariant features

The main idea is to learn a representation $\Phi$ such that $P(\Phi(x^d))$ is the same whatever the domain. In [436], Muandet et al. propose a method called Domain-Invariant Component Analysis (DICA) to learn an invariant transformation by minimizing dissimilarity between domains. By using two reproducing Hilbert space kernels (RKHSes) on $\mathcal{X}$ and $\mathcal{Y}$, respectively. Their main idea is to find a low-dimensional subspace that minimizes the dissimilarity across source domains. They also propose an unsupervised version. If DICA works well for datasets such as Parkinson's Telemonitoring, it does not give satisfactory results for more complicated dataset such as images or videos.

In [437], Ganin et al. propose a domain generalization method based on domain adaptation. The main idea is to determine features that is invariant whatever the source domain and can be used for the main learning task. They use an adversarial neural network composed of a features extractor, a task classifier $G_f$, and a domain classifier $G_d$, as shown in Fig 7.6. Thus, they train DANN with two loss functions.

In [438], Li et al. propose a domain generalization method based on adversarial

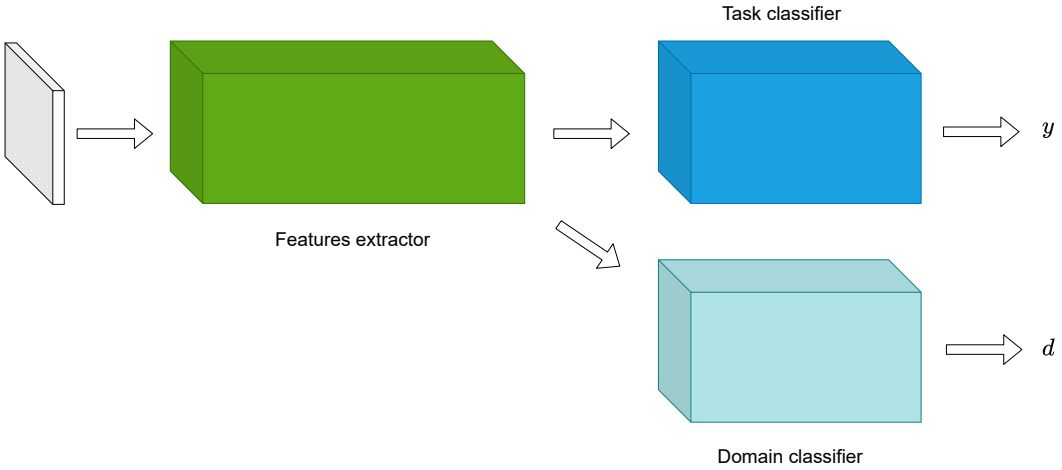



Figure 7.6: DANN

auto-encoders by imposing a Maximum Mean Discrepancy (MMD). They use an adversarial auto-encoder to extract features that must not be discriminative in terms of domain shift but discriminative to class labels. Their model named MMD-AAE is then composed of an adversarial auto-encoder and of two classifiers. If $x$ is the input of the network and $h$ the hidden code of the auto-encoder, $dn(x)$ the marginal distribution of $x$, $en(h|x)$ and $en(x|h)$ are considered as respectively the encoding and the decoding distribution and the posterior distribution $en(h)$ is as follows:

$$en(h) = \int_x en(h|x)dn(x)dx \tag{7.12}$$

The adversarial auto-encoder is trained to minimize the error between $en(h)$ and $dn(h)$ (the encoder is $En$ and the decoder is $Dn$). During training all domains using the same $En$ and $Dn$. Given m source domains, the auto-encoder loss is as follows:

$$\mathcal{L}_{ae} = \Sigma_{i=1}^{m} \left\| \hat{X}_i - X_i \right\|_2^2 \tag{7.13}$$

where $\hat{X}_i = D(H_i)$ and $H_i = E(X_i)$.

In order to avoid over-fitting, the authors introduce a prior distribution $p(h)$ to regularize the adversarial network by matching $q(Q()x))$ with $p(h)$ (The generator of their model is $Q$).

The loss of their model is as follows:

$$\min_{C,Q,P} \max_{D} \mathcal{L}_{err} + \lambda_0 \mathcal{L}_{ae} + \lambda_1 \mathcal{R}_{mmd} + \lambda_2 \mathcal{J}_{gan} \tag{7.14}$$

where $\mathcal{R}_{mmd}$ is an MMD-based regularization term, $\mathcal{J}_{gan} = \mathbb{E}_{h\sim p(h)}\left[\log D(h)\right] + \mathbb{E}_{x\sim p(x)}\left[\log(1 - D(Q(x)))\right]$, $D$ the discriminator and $\mathcal{L}_{err}$ is the loss of the classifier. As a prior distribution, they use $h \sim Laplace(\eta)$ where $\eta$ is an hyper-parameter.

They evaluate their method on a combination of benchmark datasets PASCAL VOC2007 [439], LabelMe [440], Caltech-101 [441] and SUN09 [442]. Successively one dataset is used as target while the other are as sources.

In [27], Albuquerque et al. also propose to make training distributions indistinguishable. They define a domain as $\langle \mathcal{D}, f \rangle$ where $\mathcal{D}$ is the probability distribution over $\mathcal{X}$ and $f$ the deterministic labeling function defined as $f : \mathcal{X} \to \mathcal{Y}$. They also define $h \in \mathcal{H}$ with $\mathcal{H}$ a set of candidate hypothesis with $f : \mathcal{X} \to \mathcal{Y}$.

A risk $R$ or source error can associated to $h$ on domain $\langle \mathcal{D}, f \rangle$ as follows [443]:

$$R[h] = \mathbb{E}_{x\sim\mathcal{D}}\mathcal{L}[h(x), f(x)] \tag{7.15}$$

with $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to R_+$ representing how different $h(x)$ and $f(x)$.

In [443], Ben-David et al. show that, using *covariate shift assumption*, $R[h]$ on the target domain is bounded by:

$$R_T[h] \leq R_S[h] + d_{\mathcal{H}}\left[\mathcal{D}_S, \mathcal{D}_T\right] + \lambda \tag{7.16}$$

Where $\lambda$ is the minimal total risk over both domains for a given $h$.

$d_{\mathcal{H}}\left[\mathcal{D}_S, \mathcal{D}_T\right]$ is the $\mathcal{H}$-divergence defined in [443, 444]. The $\mathcal{H}$-divergence is defined as follows:

$$d_{\mathcal{H}}\left[\mathcal{D}_S, \mathcal{D}_T\right] = 2 \sup_{\eta\in\mathcal{H}} \left| Pr_{x\sim\mathcal{D}_S}\left[\eta(x) = 1\right] - Pr_{x\sim\mathcal{D}_T}\left[\eta(x) = 1\right] \right| \tag{7.17}$$

In [445], Ben-David et al. show that it is possible to estimate $d_{\mathcal{H}}\left[\mathcal{D}_S, \mathcal{D}_T\right]$ from the error of a binary classifier. If there $N_S$ source domains, in [446], Zhao et al. show that:

$$R_T[h] \leq \sum_{i=1}^{N_S} \alpha_i \left( R_S^i[h] + \frac{1}{2} d_{\mathcal{H}}\left[\mathcal{D}_T, \mathcal{D}_S^i\right] \right) + \lambda_\alpha \tag{7.18}$$

where $\mathcal{D}_S^i$ is the $i$th source domain ($i \in [1, \cdots, N_S]$), $\lambda_\alpha$ is the minimum total cost and $\alpha_i \in [0, 1]$ is a weighting coefficient such that $\sum_{i=1}^{N_S} \alpha_i = 1$.

As shown in [27], given a meta-distribution $\mathfrak{D}$ from which source and target domains are drawn, $h^* \in \mathcal{H}$, the meta-risk $R_{\mathfrak{D}}[h]$ is defined as follows:

$$\begin{aligned} h^* &= \arg\min_{h \in \mathcal{H}} \\ R_{\mathfrak{D}}[h] &= \mathbb{E}_{\mathcal{D} \sim \mathfrak{D}} \left[ \mathbb{E}_{x \sim \mathcal{D}} \mathcal{L}[h(x), f(x)] \right] \end{aligned} \tag{7.19}$$

In domain adaptation, the $\mathcal{H}$-divergence is estimated between the source and target domain. Rather than computing pair-wise divergence between all source domains (computationally costly), they propose *one-vs-all* classification. So each source has a discriminator and the $l$th discriminator estimates $\sum_{l \neq k} d_{\mathcal{H}}[\mathcal{D}_S^k, \mathcal{D}_S^l]$.

The proposed model is composed of three modules: an encoder $E$ with parameters $\phi$, a task classifier C with parameters $\theta_C$ and a set $D_k$ discriminator with parameters $\theta_k$. Thus, the objective function of their model is:

$$\min_{\phi, \theta_C} \max_{\theta_1, \cdots, \theta_{N_S}} \mathcal{L}_C \left( C(E(x; \phi); \theta_C), y_C \right) - \sum_{k=1}^{N_S} \mathcal{L}_k \left( D_k(E(x; \phi); \theta_k), y_k \right) \tag{7.20}$$

where $y_C$ is the task label for the sample data $x$, and $y_k$ is the domain label that is to say $y_k = 1$ if $x \sim \mathcal{D}_S^k, 0$ otherwise

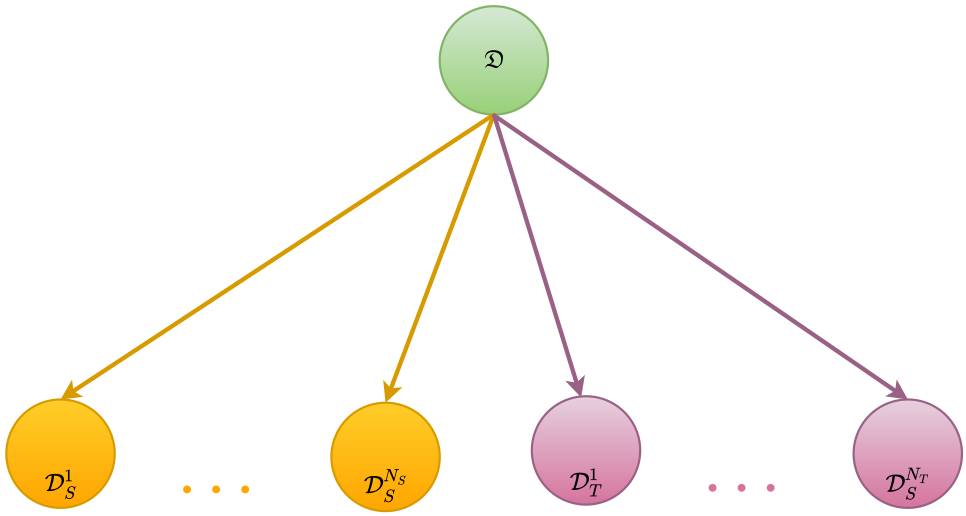



**Figure 7.7:** Meta-distribution $\mathfrak{D}$ from which source and target domain are drawn (from [27]).

#### 7.2.3.2 Hierarchical models

These models use an hierarchical of model parameters and each domain has its model given a domain-agnostic and a domain specific parameter. In [447], Khosla et al. propose a method learning two types of parameters: a dataset-dependent weight called *bias vectors* and a parameter common to all datasets called *visual world* weights. For each dataset $D_i$ ($i \in [0, \cdots, m]$) they introduce a visual world object model $\mathbf{w}_{vw}$ and a set of bias vectors $\Delta_i$. They define the relationship between weights as follows:

$$\mathbf{w}_i = \Omega\left(\mathbf{w}_{vw}, \Delta_i\right) \tag{7.21}$$

where $\mathbf{w}_i \in \mathbb{R}^m$ is the weight vector for dataset $D_i$.

Their method is based on SVM, and they use two hyper-parameters $C_1$ and $C_2$ to control the two constraints on the visual world and individual datasets. They evaluate their method on the same datasets as [438].

In [448], Li et al. propose a domain generalization method and a new benchmark dataset called PACS. The new dataset is composed of three sub datasets: Photo, Art painting, Cartoon and Sketch. They train the same model ignoring domain difference. Given $m$ domains and $N_i$ labeled instances $\left\{\left(x_j^i, y_j^i\right)\right\}_{j=1}^{N_i}$ with $x_j^i$ the input data and $y_j^i$ the class label. The loss is defined as follows:

$$\mathcal{L}_c = \underset{\theta_1,\cdots,\theta_m}{\arg\min} \frac{1}{m} \sum_{i=1}^{m} \frac{1}{N_i} l(\hat{y}_j^i, y_j^i) \tag{7.22}$$

where $\theta_i$ is the model parameter for $i$th dataset and $l$ is the loss function. They assume that there is a optimal parameter $\theta^\star = \theta_1 = \cdots = \theta_m$ representing an universal model for all sources datasets.

#### 7.2.3.3 Data augmentation

The main idea behind these methods is to add training data in order to improve the generalization to new domains. In [449], Shankar et al. propose a method called CROSSGRAD, which performs data augmentation using a Bayesian setting and train a task and domain classifiers. They show that perturbations in domain improve the generalization to unseen domains and that data augmentation is better than adversarial domain training. Besides, $x$, $y$ and $d$ they introduce a new variable $g$ from Bayesian setting in such way that $d$ *causes* $g$ which, combined with $y$, *causes* $x$. The main idea is to perturb $g$ to $g'$ during training to augment $x$ to $x'$. During perturbation of $x$, the domain classifier's loss is intended to change the most, while $y$ must remain unchanged. The task classifier loss on $x$ is combined with the $x'$es training loss in a cross-gradient training way. They apply their method to character recognition across

fonts, handwriting recognition across authors, MNIST across simulated domains, and spoken word recognition across users. In [450], propose an iterative domain augmentation using adversarial learning and the Wasserstein distance.

#### 7.2.3.4 Optimization algorithms

Another solution for domain generalization is modifying an existing algorithm to find a more robust minimum. Such a process can be done through meta-learning. In [451], Li et al. propose a domain generalization method based on meta-learning. During the training, they split source domains into meta-train domains and meta-test domains to enhance the backbone model's generalization capability.

## 7.3 Our approaches

This section presents our reflections and the work we have done on the detection of the senior's activity from the thermal cameras with the help of deep learning. One of our difficulties comes from the fact that we do not have enough data to train a network. Indeed, to our knowledge, nobody has tried to estimate thermal images' activities, and moreover, from low-resolution thermal images.

So we tried four different strategies to estimate the activity of people on thermal images (TIR) from training conducted on RGB images:

— Training on RGB edges then testing on TIR edges

— Training on RGB optical flow then testing on TIR optical flow

— Training on TIR images simulated from RGB images

— Domain generalization

### 7.3.1 Training on RGB edges then inferring on TIR edges

Our first approach ($Model_e$) is to train a model on RGB edges and then test the same model on TIR edges. Our first intuition is that RGB edges and TIR edges data distributions of the same scene are **closer** than RGB and TIR distributions (Fig 7.8). This approach can be considered as a domain generalization approach. The main idea is that the model has been trained on the same type of edges, so it could be able to generalize well.

During the training, the backbone network has no access to the TIR edges. We have experimented with three strategies:

— Computing edges using Sobel filters

— Computing edges using Laplace filters

— Computing alternatively edges using Sobel or Laplace filters. For each video, we randomly choose to compute edges using either Sobel of Laplace.

For the training and for the inference, the same strategy is applied.

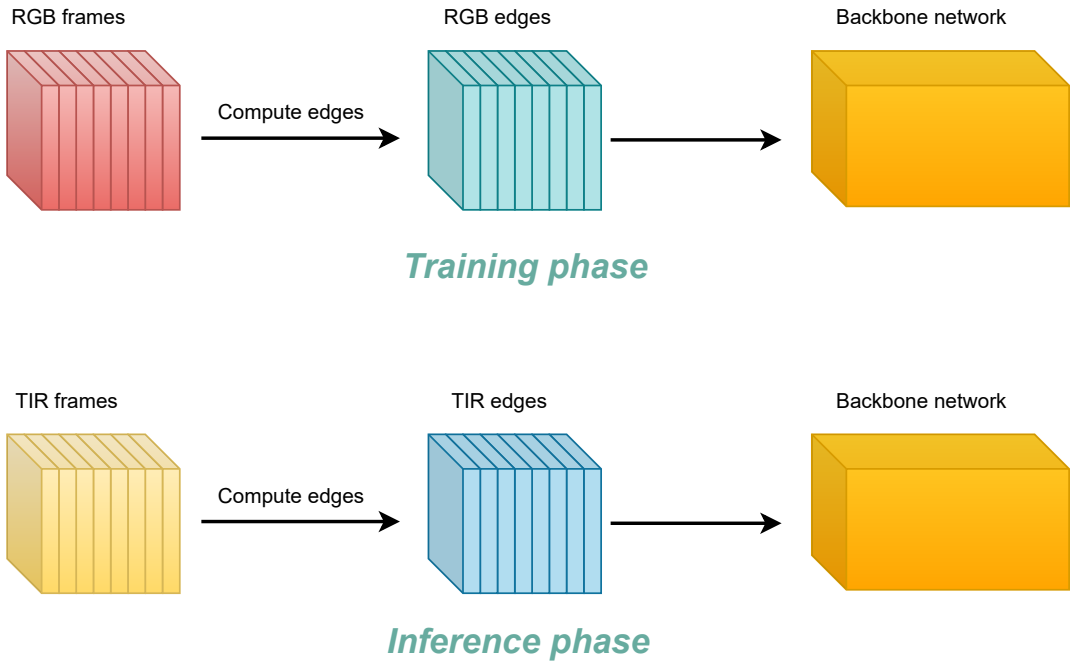



**Figure 7.8:** Edges model ($Model_e$)

### 7.3.2 Training on RGB Optical flow then inferring on TIR optical flow

This approach ($Model_{flow}$) is similar to the previous one. Rather than training on edges frames, we decided to train using optical flow extracted from RGB images (Fig 7.9). The main idea is to consider that RGB optical flow and TIR optical flow data distributions are **closer** than RGB and TIR dataset distributions. The backbone network should learn from RGB motion and then infer for TIR motion.

During the training, the backbone network also has no access to the TIR optical flow.

### 7.3.3 Training on simulated TIR then inferring on TIR data

In this approach, we tried to generate simulated thermal images. Thus we first trained a GAN model to generate simulated thermal images (Fig 7.10). For the GAN model, we used vid2vid. For training, we tested four simulation variants depending on the information we wanted to use for inference (TIR images or TIR edges) :

— RGB to TIR

— RGB Sobel to TIR

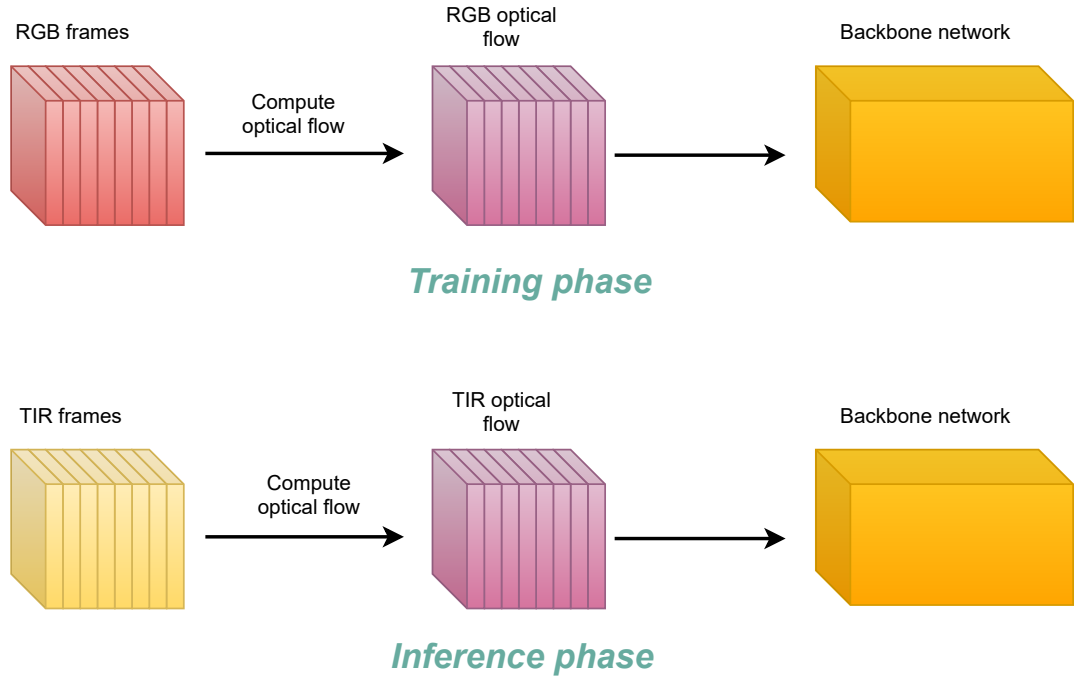



**Figure 7.9:** Optical flow model ($Model_{flow}$)

— RGB to TIR Sobel

— RGB Sobel to TIR Sobel

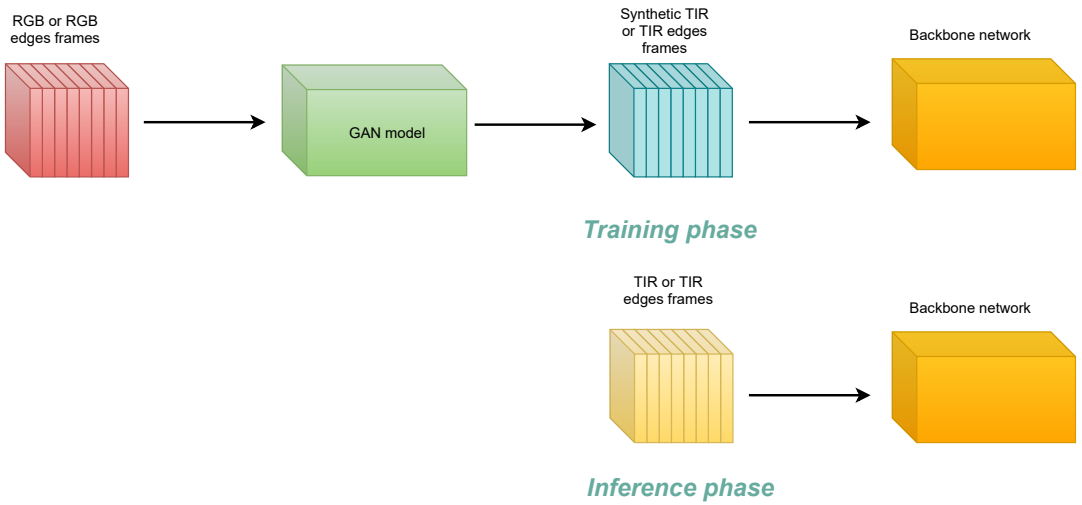



**Figure 7.10:** Simulation model ($Model_{sim}$)

### 7.3.4 Domain generalization

In this approach ($Model_{dm}$), we use the model proposed by Albuquerque et al. in [27]. We adapted their model to activity recognition by using the I3D model as a backbone.

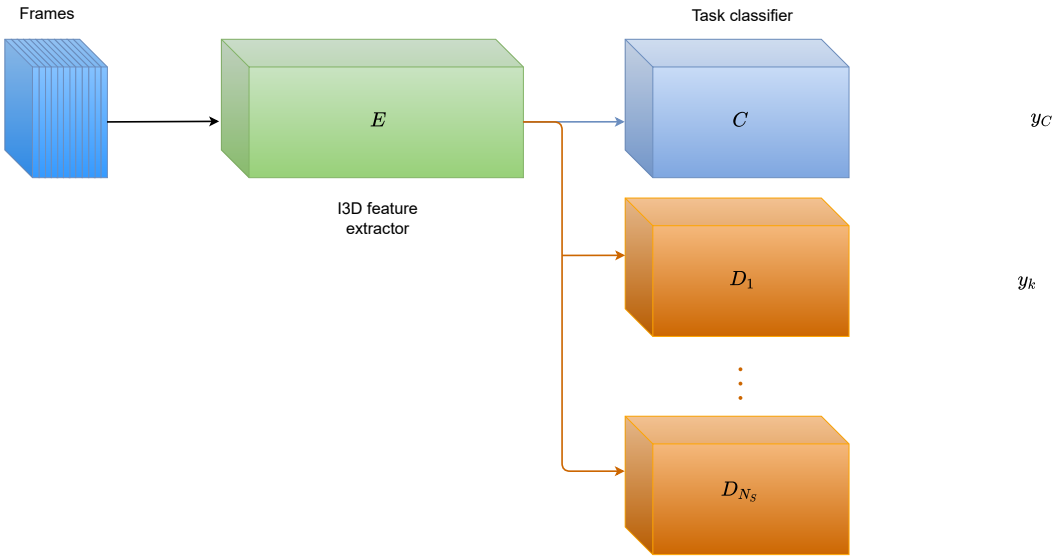



**Figure 7.11:** Domain generalization model ($Model_{dm}$)

Fig 7.11 shows the global model with the feature extractor, the classifier, and the discriminators. While the authors use data coming from distinct domains in the literature, we had a large dataset of RGB video and a dataset of TIR video in our situation.

We tried to simulate and add different domains using edge operators. Thus, we have five source domains:

— raw RGB videos
— Sobel with a Gaussian blurring size of 0 and an operator of size 3
— Sobel with a Gaussian blurring size of 3 and an operator of size 5
— Laplace with a Gaussian blurring size of 0 and an operator of size 3
— Laplace with a Gaussian blurring size of 3 and an operator of size 5

As a target domain, we applied Sobel with a Gaussian blurring size of 0 and an operator of 3 to TIR videos. Thus, the classifier will try to predict each video activity class while the discriminators will try to predict the domain of each video.

## 7.4 Datasets

### 7.4.1 Generating simulated thermal images

For simulated thermal image generation, there are many datasets available in the literature with corresponding RGB and TIR images. Table 7.1 gives a summary of these datasets.. We considered pair and unpair-image translations. But, given that we are concerned by an indoor situation, we only select dataset containing such a

situation. For paired images, we used a combination of images coming from the following datasets: CAMEL Dataset [24], VAP trimodal [452], LITIV 2017 [25] and Bilodeau [26], because the image from these datasets have a good resolution and there are various scenes. For video generation, we extract small sequences of 30 frames from each video clip. Thus, we use 2068 pairs of sequences for training and 828 for testing.

| Type | Dataset | Number of images | | Indoor images ? |
|---|---|---|---|---|
| | | RGB | TIR | |
| Paired | KAIST [453] | 50184 | 50184 | not |
| | CVC-14 [454] | 8473 | 8473 | not |
| | OSU Color Thermal [455] | 8545 | 8545 | not |
| | VAP trimodal [452] | 5924 | 5924 | yes |
| | Bilodeau [26] | 7821 | 7821 | yes |
| | LITIV 2012 [456] | 6325 | 6325 | yes |
| | LITIV 2017 [25] | 4300 | 4300 | yes |
| | CAMEL Dataset [24] | 44500 | 44500 | yes |
| | total | 136072 | 136072 | |
| Unpaired | VOT2016 [457] | 21455 | | not |
| | VOT2017 [458] | 4049 | | yes |
| | OTB [459] | 58610 | | yes |
| | ASL [460] | | 6490 | not |
| | Long-term [461] | | 47423 | not |
| | InfAR [462] | | 46121 | not |
| | total | 84114 | 100034 | |

**Table 7.1:** Dataset available in the literature

### 7.4.2 Activity recognition

#### 7.4.2.1 RGB videos

There a lot of datasets in the literature for activity recognition. But, our focus is on activity monitoring for fall detection. This is why we chose the dataset (CMDFALL) proposed by Tran et al. in [23]. They propose a multi-modal dataset with modalities as Depth frames, accelerometer and RGB frames. We are concerned by RGB frames. They conduct experiment for 50 subjects, seven view-points and 20 activities.

We modified the dataset for our purpose. First, they use 20 activities with some activities such as right fall and left fall. We have regrouped the activities that only

differ in the sense that they have been performed *left* or *right*. Besides, we did not consider one of the cameras, because it was placed on the ceiling. Moreover, given that their frame rate is 20fps, we temporally down-sampled the video to 5fps. As a result, we delete videos where the activities lasted less than 3 seconds. In the end, we got 6235 videos, which are distributed, as shown in Table 7.2.

| action id | action name | Number of videos (#) | # after over-sampling |
|---|---|---|---|
| 1 | walk | 1972 | 1972 |
| 2 | hand pick up | 319 | 1972 |
| 3 | lie then sit | 311 | 1972 |
| 4 | sit then stand up | 1017 | 1972 |
| 5 | crawl | 277 | 1972 |
| 6 | lie then fall | 834 | 1972 |
| 7 | sit then fall | 537 | 1972 |
| 8 | fall | 958 | 1972 |

**Table 7.2:** Number of videos per class for the dataset CMDFALL [23]

Unfortunately, after this data cleaning, the new dataset was imbalanced, and we had to find a solution to deal with learning from an imbalanced dataset (Table 7.2). In the literature, there are many solutions, but we chose random over-sampling with data augmentation. We used some traditional data augmentation techniques such as random flipping or random cropping. But we also introduced a new technique that should prevent the network from overfitting. Let $vid = \{I_1, \cdots, I_t\}$ be a video with length $t$ and frame size $m \times n$. For each $vid$:

— Randomly generate a rectangle with a random size $r_m \times r_n$ with $r_m \in [0, m]$ ans $r_n \in [0, n]$

— Randomly move the rectangle during the video

the camera set at the ceiling

By doing this, we are not only able to augment the dataset but we are also able to create some occlusions. Besides this technique (Fig 7.12), we add common data augmentation techniques such as random flipping, random cropping and random affine transformation.

#### 7.4.2.2 TIR videos

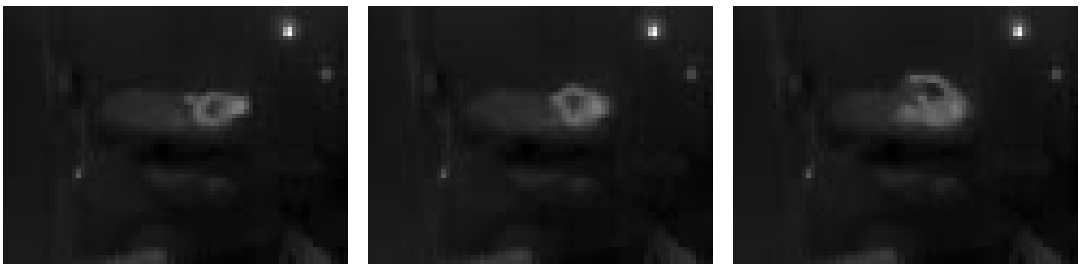
Our main purpose is activity recognition from thermal images. This is why we need to acquire data from our own cameras to evaluate the algorithms in relation to our

(a) Example 1 (b) Example 2

**Figure 7.12:** Examples of generated rectangle to augment data and induce occlusion

| action id | action name | Number of videos |
|---|---|---|
| 1 | walk | 91 |
| 2 | hand pick up | 91 |
| 3 | lie then sit | 81 |
| 4 | sit then stand up | 91 |
| 5 | crawl | 91 |
| 6 | lie then fall | 81 |
| 7 | sit then fall | 91 |
| 8 | fall | 91 |

**Table 7.3:** Number of videos per class for our dataset Baga



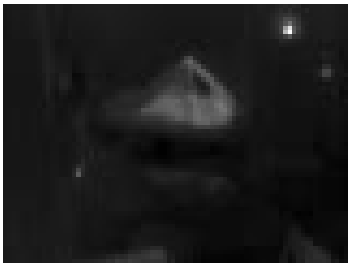
(a) Nº 0



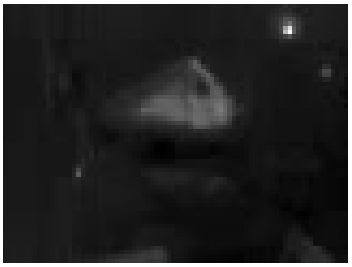
(b) Nº 1



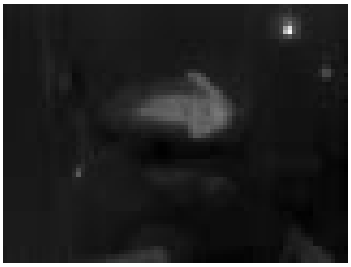
(c) Nº 2



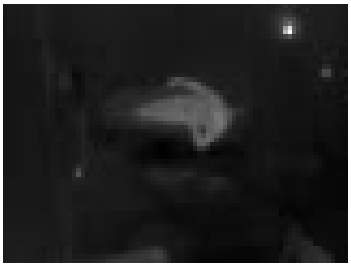
(d) Nº 3



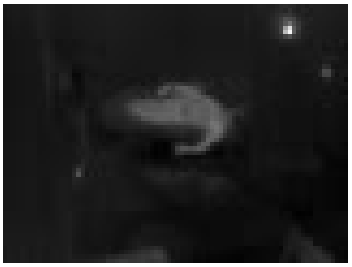
(e) Nº 4



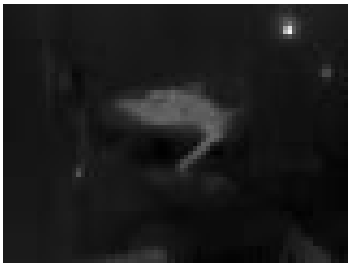
(f) Nº 5



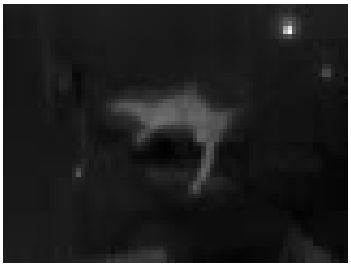
(g) Nº 6



(h) Nº 7



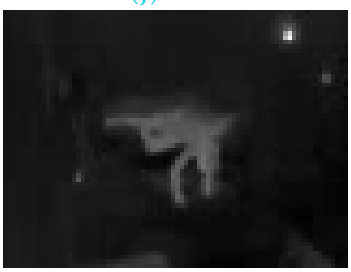
(i) Nº 8



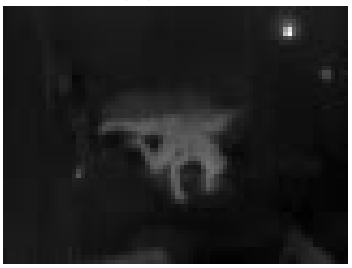
(j) Nº 9

(k) Nº 10

(l) Nº 11



(m) Nº 12



(n) Nº 13

(o) Nº 14

**Figure 7.13:** Baga dataset: frames during a fall from a bed

application's characteristics. We named this dataset Baga (fire in Mossi language) [1]. The dataset has been acquired using our stereo-system with three participants. We used various view-points. In contrast to CMDFALL, our videos have been recorded in real inhabited rooms. These rooms were furnished and therefore presented many occlusions to the cameras. Moreover, these data have been acquired for various room temperatures. Fig 7.13 shows an example of a simulated fall from a bed.

In our experiment, we only used the left camera images to avoid redundancy. We set the frame rate to 5fps, and each activity lasts at least three seconds. The characteristics of this dataset can be seen in Table 7.3.

## 7.5 Preliminary results

### 7.5.1 Implementation details

This section details the implementation details of each method.

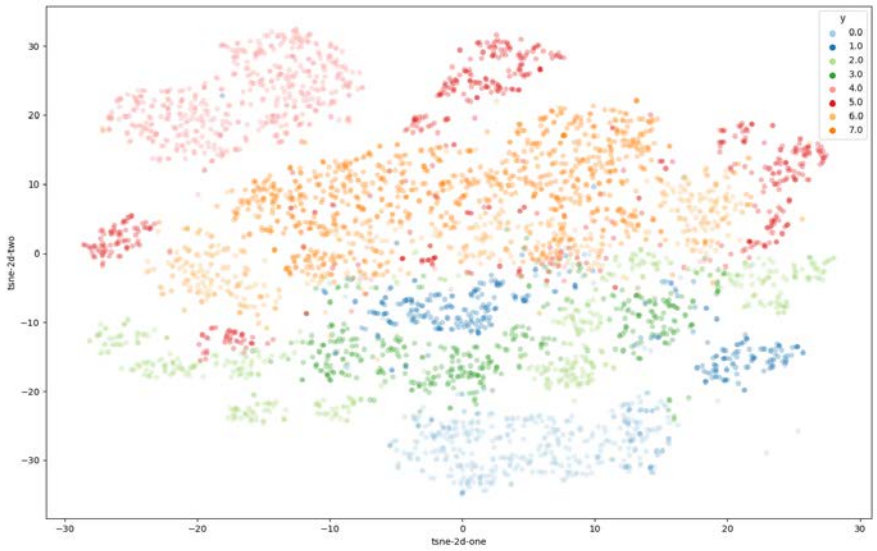
The implementation of the $Model_e$ is very straightforward. Indeed, we use publicly available code furnished by the original authors of I3D [2] and R(2+1)D [3]. But, we heavily modified these codes to adapt them for our low-resolution TIR images and our data augmentation techniques. The framework has been built on top of Pytorch [463]. For I3D, we used a pretrained model (Kinetics dataset), and we trained it for 10000 iterations with a batch size of 16. The initial learning rate is set to $0.01$ with multi-step decay. The model is optimized using stochastic gradient descent with a momentum of $0.9$ and a weight decay of $1e^{-7}$. Regarding R(2+1)D, we also used a pretrained model (Kinetics dataset), and we trained it for 20 epochs with a batch size of 32. The model is also optimized using stochastic gradient descent with a momentum of $0.95$ and a weight decay of $1e^{-3}$. The initial learning rate is set to $1e^{-3}$.

The implementation of the $Model_{flow}$ is similar to $Model_e$ excepted the fact that the input of the neural network is optical flow computed using the TV L1 method [464].

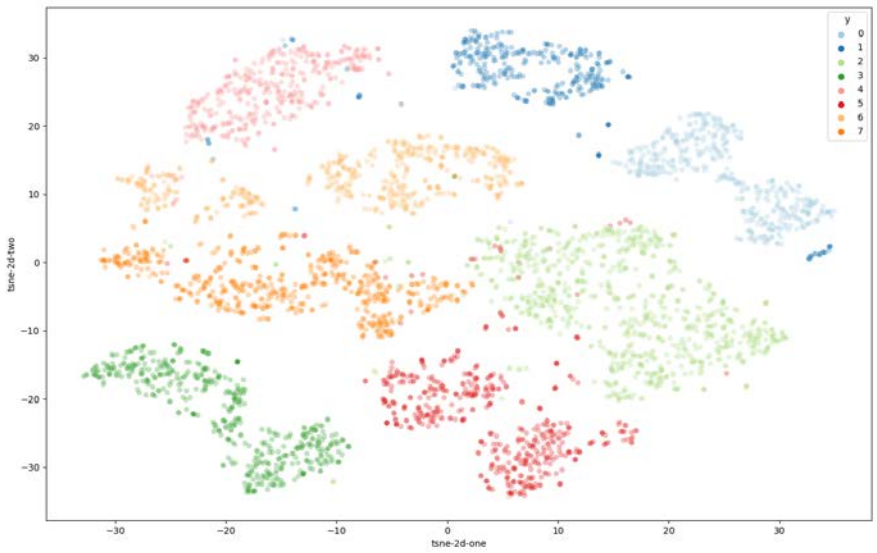
The implementation of the $Model_{sim}$ is based on the previous model, to which we attached during training a trained vid-2-vid model. During training, the parameters of the GAN model are frozen.

For $Model_{dm}$, we use a pretrained I3D backbone on the Kinetics dataset. The domain classifier is composed of two fully connected layers, followed by a ReLU and a dropout. There is also a last fully connected layer, followed by a softmax layer. The task classifier is composed of an average pooling followed by a dropout, a 3d convolutional

1. https://drive.google.com/file/d/1HOTDmWVp3A52pVAbWnJyFWzHHLYIEHFZ/view?usp=sharing
2. https://github.com/piergiaj/pytorch-i3d
3. https://github.com/microsoft/computervision-recipes

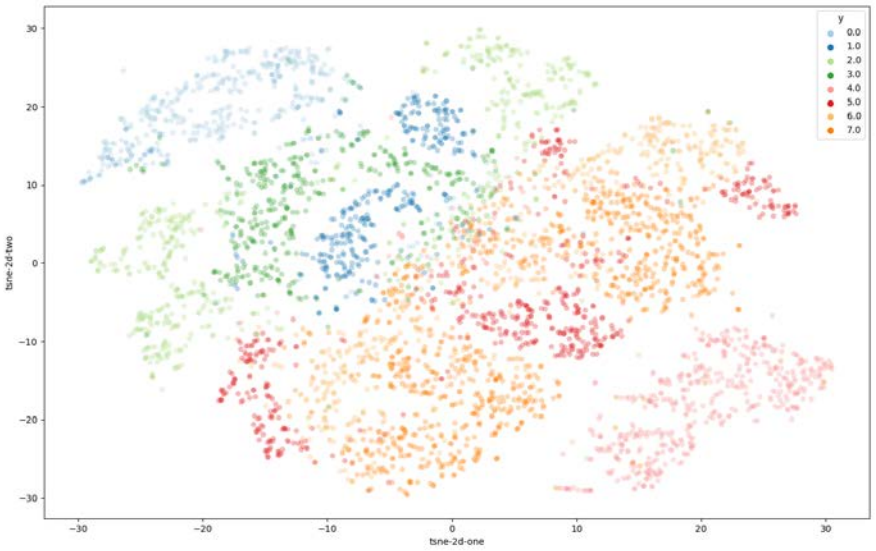



(a) t-SNE features for I3D $Model_e^{sobel}$

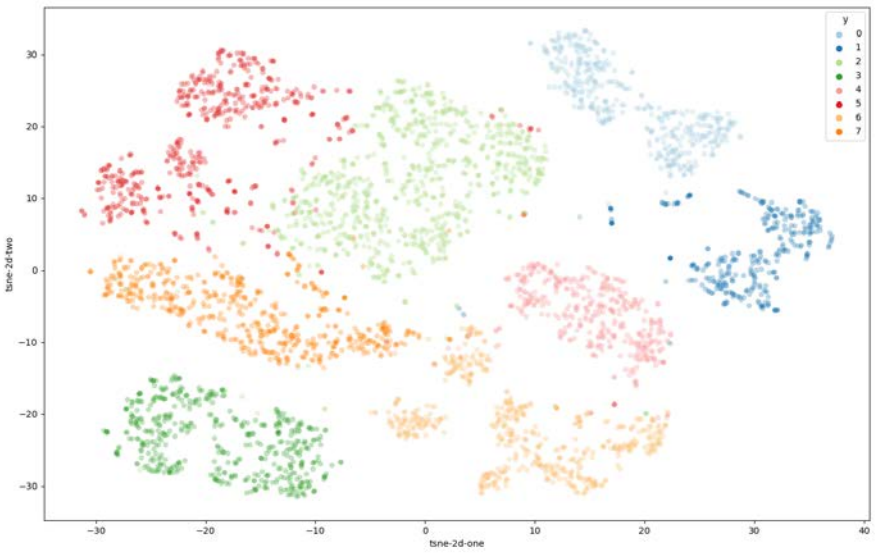



(b) t-SNE features for R(2+1)D $Model_e^{sobel}$

**Figure 7.14:** t-SNE features for $Model_e^{sobel}$

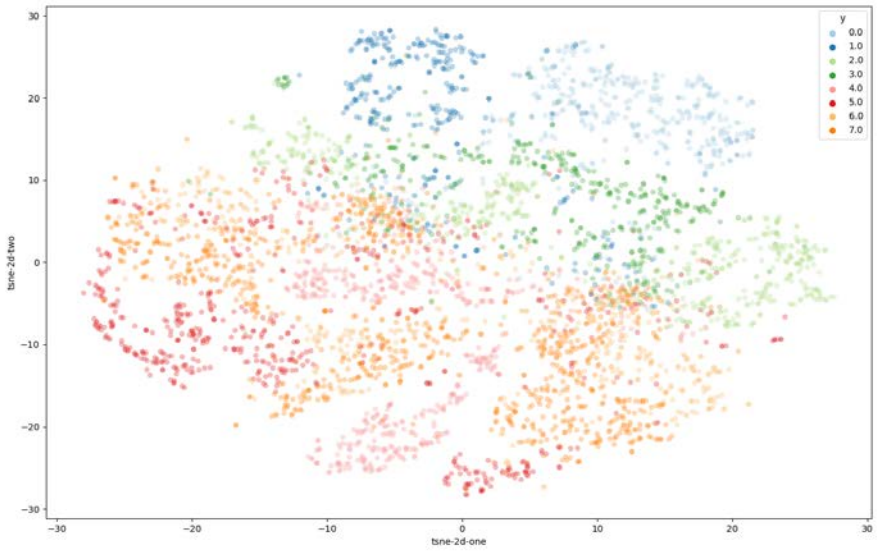



(a) t-SNE features for I3D $Model_e^{laplace}$

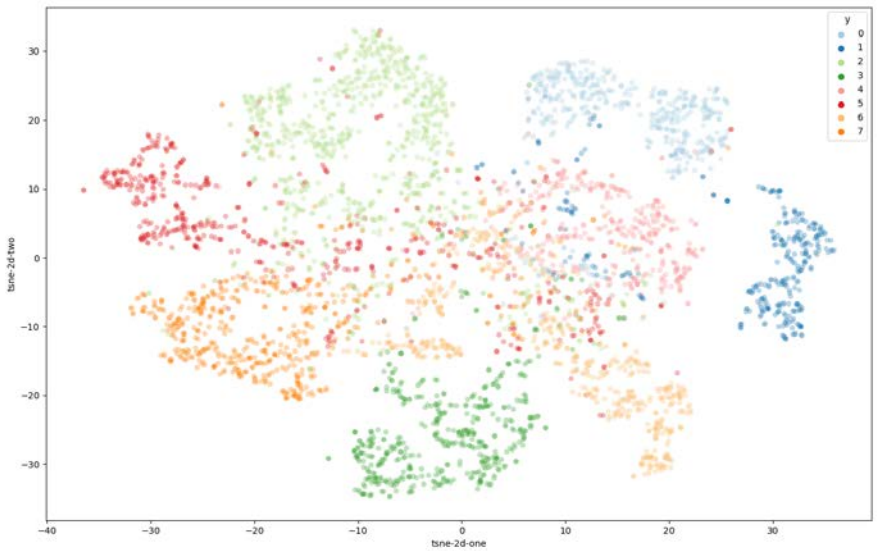



(b) t-SNE features for R(2+1)D $Model_e^{laplace}$

**Figure 7.15:** t-SNE features for $Model_e^{laplace}$

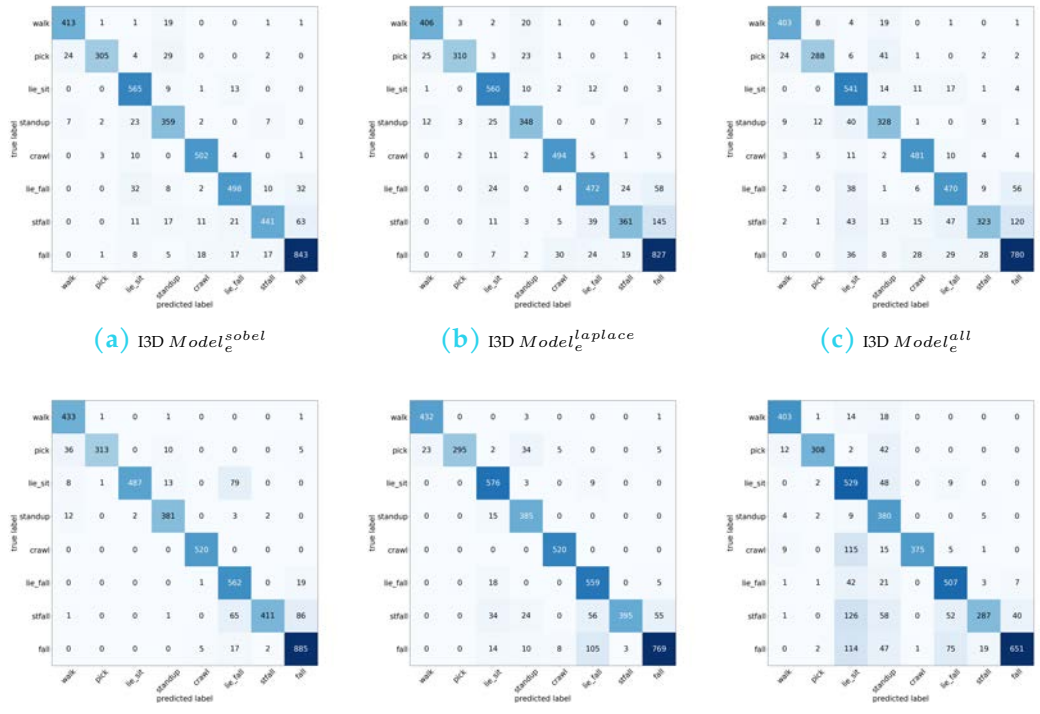



(a) t-SNE features for I3D $Model_e^{all}$




(b) t-SNE features for R(2+1)D $Model_e^{all}$

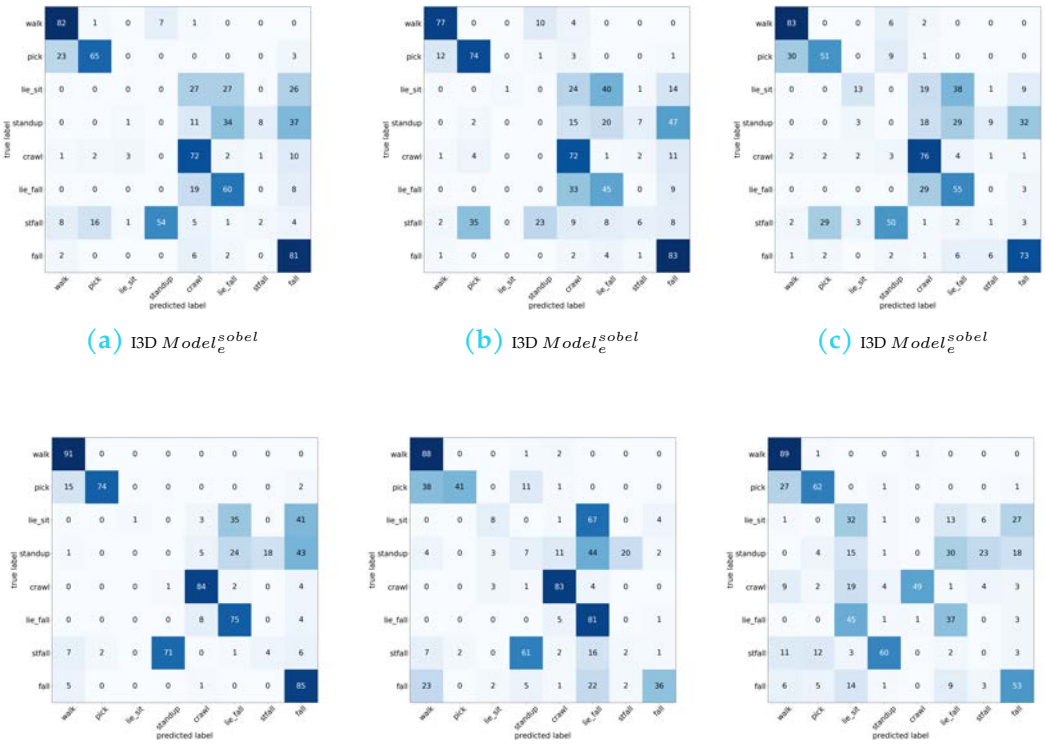
**Figure 7.16:** t-SNE features for $Model_e^{all}$




(a) I3D $Model_e^{sobel}$ (b) I3D $Model_e^{laplace}$ (c) I3D $Model_e^{all}$

(d) R(2+1)D $Model_e^{sobel}$ (e) R(2+1)D $Model_e^{laplace}$ (f) R(2+1)D $Model_e^{all}$

**Figure 7.17:** Confusion matrices on testing dataset




(a) I3D $Model_e^{sobel}$ (b) I3D $Model_e^{sobel}$ (c) I3D $Model_e^{sobel}$

(d) R(2+1)D $Model_e^{sobel}$ (e) R(2+1)D $Model_e^{sobel}$ (f) R(2+1)D $Model_e^{sobel}$

**Figure 7.18:** Confusion matrices on inference dataset (Baga)

layer, and a softmax layer. The model is trained using stochastic gradient descent for 50 epochs, and the initial learning rates are $0.01$ for both the task and the domain classifier.

### 7.5.2 I3D features vs R(2+1)D features

In all of the four models, there is a backbone network that extracts features. In this section, we will analyze the ability of these networks to extract the most effective features. Moreover, most of our models used edge information. We proposed to use several ways to assess this information (Sobel, Laplace, or both randomly-all). We wanted to compare the two networks' ability to extract effective features from the edges images and which edge estimation method gives the most discriminating information. In the case of $model_e$, we trained the three variants (Sobel, Laplace, all) with the two network architectures (I3D features and R(2+1)D) and then extracted features from the testing video clips. We considered the features which appear at the output of the one-but-last layer of the network. The effectiveness of the features was then visually analyzed after applying t-SNE using l2 distance [465]. For each class, a color is assigned. On the t-SNE, the network and edge extraction variant that gives the more discriminant features should present a graph with condensed classes in compact clusters and with clusters that are as separated as possible.

The Figures 7.14, 7.15 and 7.16 represent the result of TSNE on respectively $Model_e^{sobel}$, $Model_e^{laplace}$ and $Model_e^{all}$ features with I3D and R(2+1)D as backbones. For each class, a color is assigned. At a first glance, one can remark that the R(2+1)D features are more discriminant than I3D features. Even if there are some outliers for R(2+1)D, its features are visually more discriminant; excepted, however, for the R(2+1)D $Model^all_e$ features where the features are less well distributed. This is due to the fact that this model is not able to distinguish clearly which type of edge operator is used. On the contrary, regarding I3D features, it is $Model^all_e$, which seems to provide the most discriminant features.

In conclusion, from these figures, it appears that the R(2+1)D model should be used for activity recognition.

### 7.5.3 Comparison of approaches

This section is a first attempt to compare each model's generalization capacity and performance and each variant within a model. For this, we trained all the variants of models on the RGB data from the CMDFALL dataset and inferred them using our TIR data from the Baga dataset. Because Baga has ground truth (all the activities were annotated on the sequences), we were able to estimate the performance of our activity classifier in terms of confusion matrices and also using some quantitative scores like accuracy (eq. 6.3) and F1 score (eq. 6.4).

| Approach | Method | I3d | R(2+1)D |
|---|---|---|---|
| $Model_e$ | Sobel | 0.5077/0.4284 | 0.5806/0.5075 |
| | Laplace | 0.5021/0.4219 | 0.4853/0.4380 |
| | All | 0.4937/0.4371 | 0.4530/0.4291 |
| Simulated data $Model_{sim}$ | RGB-2-EDGETIR | 0.3352/0.2984 | 0.4586/0.4221 |
| | EDGERGB-2-EDGETIR | 0.1136/0.0879 | 0.2286/0.1621 |
| | EDGERGB-2-TIR | 0.1360/0.1089 | 0.3310/0.3418 |
| | RGB-2-TIR | 0.5035/0.4767 | 0.5806/0.5685 |
| Optical flow $Model_{flow}$ | TV-L1 | 0.4450/0.4256 | N/A |
| Domain generalization $Model_{dm}$ | Target-invariant domain generalization | 0.5035/0.4952 | N/A |

**Table 7.4:** Comparison of accuracy/f1 score of models (I3d/ R(2+1)D) trained on Sobel, Laplace or All (Sobel or Laplace) edges and on simulated generated images using vid-2-vid model

First of all, we evaluate each model's performance in the test domain; it is trained on. In Fig 7.17, we computed the confusion matrices using the testing dataset. We made the same experiments on inference dataset on Fig 7.18. The confusion matrices and accuracy/F1 score are displayed in Fig 7.18 and Table 7.4.

Globally, we can notice that the different models under-perform the inference dataset compared to the testing dataset. Such a result is due to the difference between the training/testing dataset and the inference dataset. In this situation, the main problem is that the trained model does not know how is the inference dataset. In our situation, the training dataset is composed of visible images, while the inference dataset is composed of thermal images. We thought that we could reduce the domain discrepancy by computing edges or create simulated thermal images from visible images as a straightforward strategy.

Nevertheless, by considering the performance of these relatively simple strategies, one can notice through Table 7.4 that between model based on edges, Sobel is the type of edge operator which gives the best generalization performance. Indeed, whatever I3D or R(2+1)D, Sobel gives an accuracy of 0.5077/0.5806 compared to Laplace, which gives 0.5021/0.4853. Such an interpretation is also reflected in the F1-score. Regarding the simulated dataset, the model showing the best performance is RGB-2-TIR.

Very surprising, the Table 7.4 shows that $Model_{flow}$ and $Model_{dm}$ does not outperform naive and simulated data approaches. These preliminary results show that the best models are using Sobel or simulated RGB-2-TIR dataset. To explain the results, we need to investigate the models more.

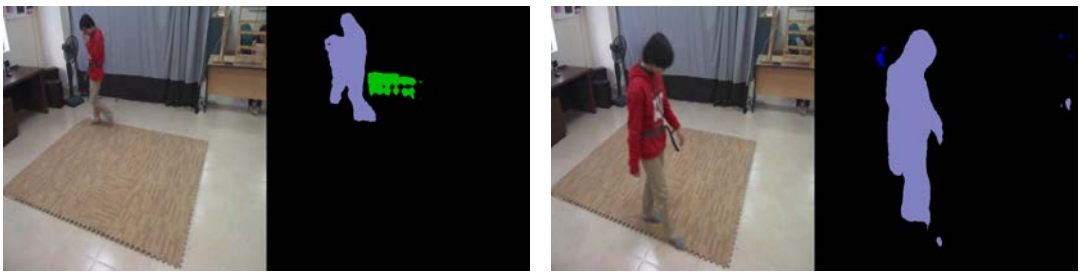
## 7.6 Improvement hints

The approaches we presented in this chapter can be improved in many ways:

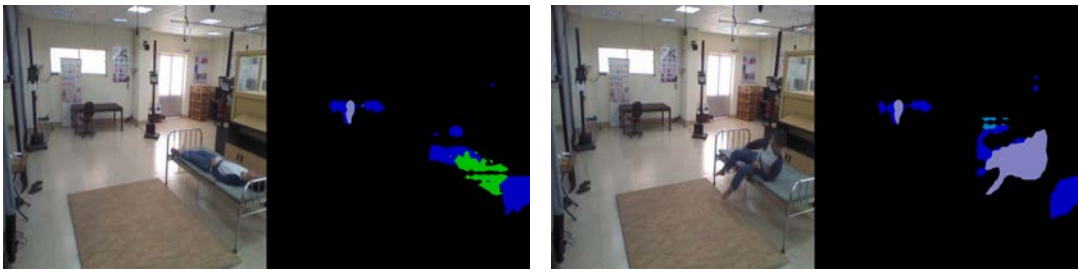
— The current simulated data are full of artifacts coming from the dataset used to train the GAN model. The GAN model can easily generate data from a given dataset that is closed to the training dataset. But when the testing dataset (here CMDFALL) scenes do not look like the training dataset, the GAN model is not performing well. One solution could be to use semantic input maps as many models in the literature. Such a solution can be easily be implemented on top of state-of-the-art segmentation frameworks such as deeplab [466] or detectron [467]. Indeed, if the image is correctly semantically segmented, the GAN model could be improved. Fig 7.19 shows the segmentation results performed by the deeplab-v3 model already trained.

Although this model has proved to be efficient in other contexts [466], it is currently unable to generalize well in the CMDFALL dataset. Improving this

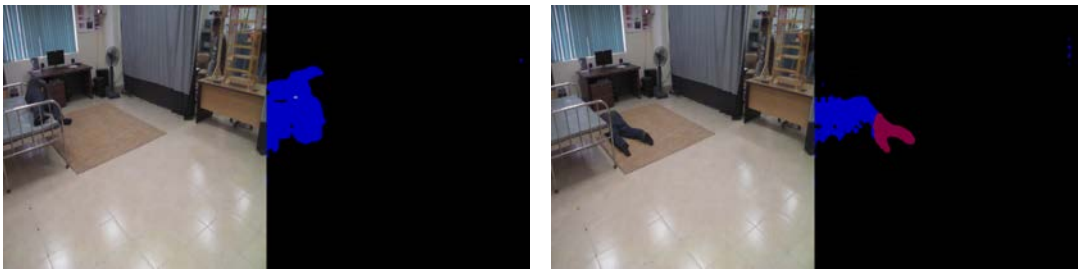
(a) Segmentation of a person lying on a bed



(b) Segmentation of a walking person



(c) Segmentation of a person lying on a bed



(d) Segmentation of a person falling from bed

**Figure 7.19:** Series of images from CMDFALL dataset semantically segmented using a pretrained deeplab-v3 model. The results are inconsistent.

performance is not straightforward, as it may require to segment CMDFALL manually or find a way to segment this dataset correctly, indeed, a Kafkaesque nightmare [4]. However, since the quality of segmentation highly affects overall performance, improving this segmentation is expected to lead to better results.

— Another solution to improve the results could be to enhance the performance of the domain generalization model. In this work, we used as domains: raw RGB frames, Sobel, and Laplace operators. It could be possible to improve the results by computing the optical flow of these domain frames (as done in [27]). Besides, we can also enhance the task and domain classifier. For example, the domain classifier we used has been designed for image classification.

Finally, perhaps "**Data is all you need**," and by acquiring more data, we will be able to perform domain adaptation or directly learn from a large amount of TIR videos. Even if data acquisition is time-consuming and the annotation, even more, more data could improve by a certain margin than the current results.

## 7.7 Conclusion

This chapter investigated and compared many strategies to perform activity recognition when we do not have enough dataset. Given a huge dataset of RGB sequences, we tried to train a neural network and investigate how such a *knowledge* could be *translated* to the thermal dataset.

The preliminary results showed that a simple approach based on the Sobel operator's edges and an approach using simulated TIR images from RGB gave promising results. But these results are only some preliminary studies; we need to investigate more to analyze the reason for these results and how we could improve this transfer learning from RGB to TIR sequences or images.

4. https://www.merriam-webster.com/dictionary/Kafkaesque

# 8

# Conclusion and perspectives

Trust yourself,
Trust yourself to do the things
that only you know best.
Trust yourself,
Trust yourself to do what's right
and not be second-guessed.

Bob Dylan

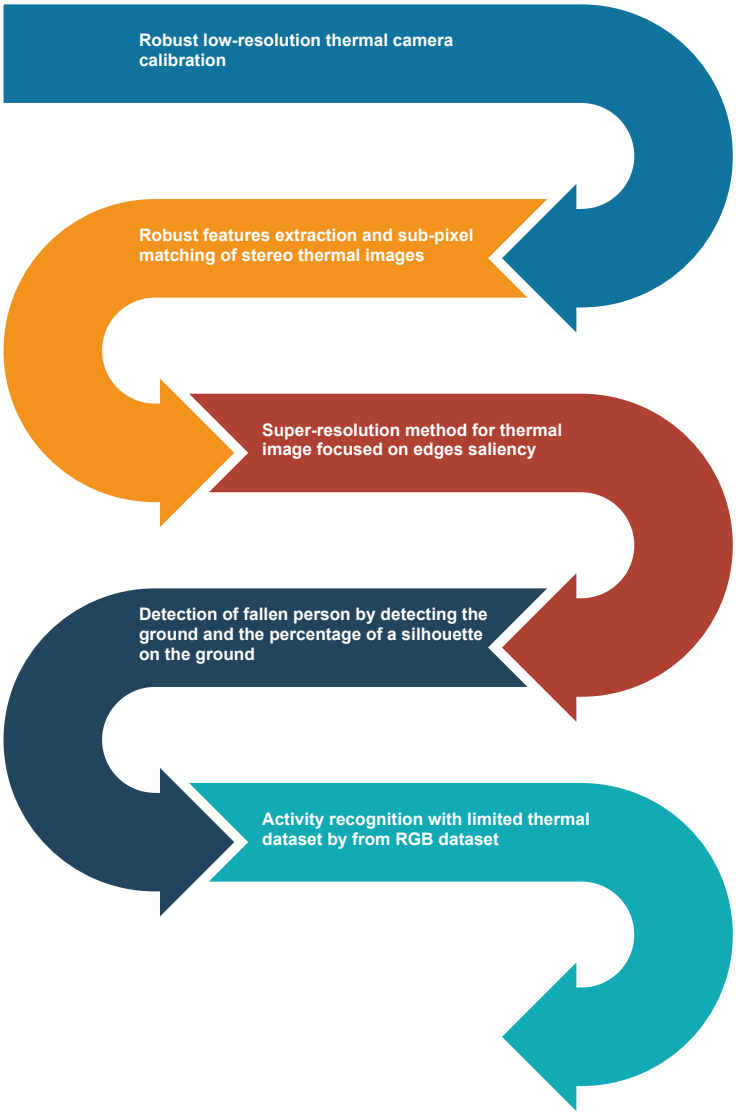



**Figure 8.1:** Summary of our contributions

The present work had as main objective to propose new methods (Fig 8.1) for fall detection and activity monitoring. The solutions we proposed and developed are based on both classical and deep learning methods. Five main contributions can be highlighted from this work.

The first contribution is manufacturing a simple yet effective stereo vision system and an active calibration grid. Despite the low-resolution aspect of our camera, we proposed a robust calibration method. We evaluated our solution by retrieving the calibration parameters and some physical values such as the cameras' focal length or the stereo baseline. The proposed algorithm was able to estimate the optimal number of calibration images needed to get a robust calibration.

Our second contribution is $ST$ (for Stereo Thermal) framework, a series of image processing steps for performing stereo vision on these low-resolution images. We compared the standard features extraction solutions and inspired by Hajebi et al., we proposed and implemented a feature extractor based on phase congruency. This implementation is fast and efficient by using SIMD optimization on top of the Eigen library. Even if phase congruency is computationally costly, in our implementation, we were able to extract features in 7 ms. We then improved the stereo matching by including the features extracted by phase congruency from the pair of images and correlating them with more robust constraints than classical techniques. Moreover, we compared our method to a traditional state-of-art method using ORB for features extraction and KNN for matching. We showed that our framework is more adapted to low-resolution thermal images in terms of the number of matches and accuracy. To make the matching more accurate, we proposed to apply phase correlation to get sub-pixel precision matching. In our low-resolution context, such an accurate sub-pixel estimation of the disparity is crucial to get a precise 3D localization by stereo-reconstruction.

Our third contribution is also going in the direction to alleviate the low-resolution issue of our camera. Thus, we proposed Edge Focused Thermal Super-resolution (EFTS) to apply super-resolution to thermal images by deep learning. First, we proposed a network that paid more attention to the edges in the images. For this, we preceded a model which have proven to be efficient for the super-resolution of visible images (RDN) by an edge extraction module. We also trained our model on pairs of thermal images composed of high-resolution images and low-resolution images produced from the high-resolution images using degradation models with random parameters. We evaluated the performance of our model using PSNR/SSIM/EPI, and these evaluations confirmed the importance of highlighting edges at the beginning of the the network. We tested many types of edge extraction methods and chose the combination of Sobel, Prewitt, and Laplacian operators. Unlike previous models, our model is trained for blind super-resolution to ensure robustness.

Once we had defined our stereo 3D reconstruction framework, we wanted to use it to detect seniors' falls. But even if we had more accurate results than the classical methods, the 3D reconstruction accuracy was insufficient to compensate for low resolution. For example, we set some hot features on the ground, but when we tried to fit a plane on the reconstructed 3D points, we observed some ground points localization errors of the order of 50 cm from the estimated plane. Such dispersion of 3D points has also been observed on silhouettes reconstruction. Then, to detect falls, we proposed to extend *ST* by *TSFD*. Unfortunately, given our low-resolution, the 3D reconstruction was not accurate enough to detect a human fall. This is why we proposed to detect a senior fall by learning the 3D position of points *on the ground* or *not on the ground*. We trained an SVM and a deep learning-based method to classify points pairs. We showed that *TSFD* is very efficient compared to other fall detection methods proposed in the literature. But, while *TSFD* can detect falls from stereo-pairs, it could be possible to enhance its performance by extending it to a multi-frame framework.

Finally, given that our previous contributions concerned static fall detection and handcrafted features, we proposed to explore some solutions to detect activities by multi-frame analysis. To bypass the sparsity of thermal-based activity monitoring dataset, we evaluated a set of solutions by (1) learning from edges of RGB images then inferring on edges of Thermal InfraRed (TIR) images (2) learning from RGB frames optical flow then inferring on TIR frames optical (3) learning from simulated TIR frames then inferring of real TIR frames and (4) by using a domain generalization technique. Unlike previous works only focusing on detecting falls, through these approaches, we tried to monitor activities.

Even if our activity recognition approaches are promising, such models are not reliable. To produce a model that could be deployed in a retirement home, we should be more accurate, and a lot of improvements are possible. Two global ways could reach such improvements: improve the current approaches or acquire more data for fine-tuning or transfer learning.

# Bibliography


1. Abbate, S., Avvenuti, M., Corsini, P., Light, J. & Vecchio, A. in *Wireless Sensor Networks: Application-Centric Design* chap. 9 (IntechOpen, Dec. 2010). https://doi.org/10.5772/13802.
2. CNET. *Heat seeker: Meet the thermal-imaging camera you can afford* https://www.cnet.com/news/he seaker-thermal-imaging-camera-for-the-masses/. 2014. (2018).
3. Zoetgnandé, Y. W. K., Fougères, A.-J., Cormier, G. & Dillenseger, J.-L. *Robust low resolution thermal stereo camera calibration* in *Eleventh International Conference on Machine Vision (ICMV 2018)* **11041** (2019), 110411D.
4. Zoetgnande, Y. W. K., Cormier, G., Fougères, A.-J. & Dillenseger, J.-L. Sub-pixel matching method for low-resolution thermal stereo images. *arXiv preprint arXiv:1912.00138* (2019).
5. Zoetgnande, Y., Dillenseger, J.-L. & Alirezaie, J. *Edge focused super-resolution of thermal images* in *2019 IEEE International Joint Conference in Neural Networks (IJCNN)* (2019), 1–12.
6. Harris, C. & Stephens, M. *A Combined Corner and Edge Detector* in *Procedings of the Alvey Vision Conference 1988* (1988), 147–151.
7. Tomasi, C. & Kanade, T. Shape and motion from image streams: a factorization method. *Proceedings of the National Academy of Sciences* **90,** 9795–9802 (1993).
8. Rosten, E., Porter, R. & Drummond, T. Faster and better: a machine learning approach to corner detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **32,** 105–119 (2008).
9. Calonder, M., Lepetit, V., Strecha, C. & Fua, P. *BRIEF: Binary Robust Independent Elementary Features* in *Computer Vision ECCV 2010* (Springer Berlin Heidelberg, 2010), 778–792.
10. Morrone, M. & Owens, R. Feature detection from local energy. *Pattern Recognition Letters* **6,** 303–313 (1987).
11. Kovesi, P. Image Features from Phase Congruency. *Videre J. Comput. Vision Res.* **1,** C3–C3. ISSN: 1041-1135 (1999).
12. Hajebi, K. & Zelek, J. S. J. *Sparse disparity map from uncalibrated infrared stereo images* in *3rd Canadian Conference on Computer and Robot Vision (CRV)* (2006), 17–17.
13. Wang, L., Zhang, C., Liu, Z., Sun, B. & Tian, H. *Image feature detection based on phase congruency by Monogenic filters* in *The 26th Chinese Control and Decision Conference (2014 CCDC)* (2014), 2033–2038.

14. Lades, M. *et al.* Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers* **42,** 300–311 (1993).
15. Karami, E., Prasad, S. & Shehata, M. Image matching using SIFT, SURF, BRIEF and ORB: performance comparison for distorted images. *arXiv preprint arXiv:1710.02726* (2017).
16. Choi, Y., Kim, N., Hwang, S. & Kweon, I. S. Thermal image enhancement using convolutional neural network. *IEEE International Conference on Intelligent Robots and Systems* **2016-Novem,** 223–230. ISSN: 21530858. http://m.koasas.kaist.ac.kr/handle/10203/216293%20https://www.scopus.com/inward/record.uri?eid=2-s2.0-85006371446&doi=10.1109%2FIROS.2016.7759059&partnerID=40&md5=dd1f6718802ba397eae3e57c0459baf2 (Oct. 2016).
17. Bhattacharya, P., Riechen, J. & Zölzer, U. *Infrared Image Enhancement in Maritime Environment with Convolutional Neural Networks.* in *VISIGRAPP (4: VISAPP)* (2018), 37–46. ISBN: 9789897582905. http://www.scitepress.org/Papers/2018/66187/66187.pdf.
18. Zhang, Y., Tian, Y., Kong, Y., Zhong, B. & Fu, Y. *Residual dense network for image super-resolution* in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018).
19. Wu, Z., Fuller, N., Theriault, D. & Betke, M. *A Thermal Infrared Video Benchmark for Visual Analysis* in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops* (IEEE, June 2014), 201–208. ISBN: 978-1-4799-4308-1. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6909984.
20. Huang, G., Liu, Z., Van Der Maaten, L. & Weinberger, K. Q. *Densely connected convolutional networks* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), 4700–4708.
21. Quero, J., Burns, M., Razzaq, M., Nugent, C. & Espinilla, M. *Detection of Falls from Non-Invasive Thermal Vision Sensors Using Convolutional Neural Networks* in *Multidisciplinary Digital Publishing Institute Proceedings* **2** (2018), 1236.
22. Nogas, J., Khan, S. S. & Mihailidis, A. *Fall detection from thermal camera using convolutional lstm autoencoder* in *Proceedings of the 2nd workshop on Aging, Rehabilitation and Independent Assisted Living, IJCAI Workshop* (2018).
23. Tran, T.-H. *et al. A multi-modal multi-view dataset for human fall analysis and preliminary investigation on modality* in *2018 24th International Conference on Pattern Recognition (ICPR)* (2018), 1947–1952.
24. Gebhardt, E. & Wolf, M. *Camel dataset for visual and thermal infrared multiple object detection and tracking* in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (2018), 1–6.
25. St-Charles, P.-L., Bilodeau, G.-A. & Bergevin, R. *Mutual foreground segmentation with multispectral stereo pairs* in *Proceedings of the IEEE International Conference on Computer Vision Workshops* (2017), 375–384.

26. Bilodeau, G.-A., Torabi, A., St-Charles, P.-L. & Riahi, D. Thermal–visible registration of human silhouettes: A similarity measure performance evaluation. *Infrared Physics & Technology* **64,** 79–86 (2014).
27. Albuquerque, I., Monteiro, J., Falk, T. H. & Mitliagkas, I. Adversarial target-invariant representation learning for domain generalization. *arXiv preprint arXiv:1911.00804* (2019).
28. Dobashi, H., Tajima, T., Abe, T. & Kimura, H. *Fall detection system for bather using ultrasound sensors* in *Proceedings of the 9th Asia Pasific Industrial Engineering & Management Systems Conference* **1865** (1860).
29. Khan, S. S. & Hoey, J. Review of fall detection techniques: A data availability perspective. *Medical engineering & physics* **39,** 12–22 (2017).
30. Johnson, J. *WHAT IS A FALL?* https://www.hqsc.govt.nz/assets/Falls/PR/ARRC-mini-collaborative/LS1-what-is-a-fall-Feb-2013.pdf (2020).
31. Zecevic, A. A., Salmoni, A. W., Speechley, M. & Vandervoort, A. A. Defining a fall and reasons for falling: comparisons among the views of seniors, health care providers, and the research literature. *The Gerontologist* **46,** 367–376 (2006).
32. Andres, R., Coppard, L., Gibson, M. & Kennedy, T. Kellogg International Work Group on the Prevention of Falls by the Elderly. *The prevention of falls in later life. Dan Med J* **34,** 1–24 (1987).
33. Lach, H. W. *et al.* Falls in the elderly: reliability of a classification system. *Journal of the American Geriatrics Society* **39,** 197–202 (1991).
34. Buchner, D. M. *et al.* Development of the common data base for the FICSIT trials. *Journal of the American Geriatrics Society* **41,** 297–308 (1993).
35. Means, K. M., Rodell, D. E., O'Sullivan, P. S. & Cranford, L. A. Rehabilitation of elderly fallers: pilot study of a low to moderate intensity exercise program. *Archives of physical medicine and rehabilitation* **77,** 1030–1036 (1996).
36. Berg, W. P., Alessio, H. M., Mills, E. M. & Tong, C. Circumstances and consequences of falls in independent community-dwelling older adults. *Age and ageing* **26,** 261–268 (1997).
37. Morris, J., Fries, B., Belleville-Taylor, P., *et al.* RAI Home-Care (RAI-HC) User's Manual, Canadian Version. *Washington, DC: interRAI* (2010).
38. Carter, N. D. *et al.* Community-based exercise program reduces risk factors for falls in 65-to 75-year-old women with osteoporosis: randomized controlled trial. *Cmaj* **167,** 997–1004 (2002).
39. Cesari, M. *et al.* Prevalence and risk factors for falls in an older community-dwelling population. *The Journals of Gerontology Series A: Biological Sciences and Medical Sciences* **57,** M722–M726 (2002).
40. Ambrose, A. F., Paul, G. & Hausdorff, J. M. Risk factors for falls among older adults: a review of the literature. *Maturitas* **75,** 51–61 (2013).

41. Bergen, G. Falls and fall injuries among adults aged 65 years—United States, 2014. *MMWR. Morbidity and mortality weekly report* **65** (2016).
42. Dunlop, D. D., Manheim, L. M., Sohn, M.-W., Liu, X. & Chang, R. W. Incidence of functional limitation in older adults: the impact of gender, race, and chronic conditions. *Archives of physical medicine and rehabilitation* **83,** 964–971 (2002).
43. Ganz, D. A., Bao, Y., Shekelle, P. G. & Rubenstein, L. Z. Will my patient fall? *Jama* **297,** 77–86 (2007).
44. Rubenstein, L. Z., Josephson, K. R. & Robbins, A. S. Falls in the nursing home. *Annals of internal medicine* **121,** 442–451 (1994).
45. Deandrea, S. *et al.* Risk factors for falls in community-dwelling older people:" a systematic review and meta-analysis". *Epidemiology*, 658–668 (2010).
46. Jensen, J. L., Brown, L. A. & Woollacott, M. H. Compensatory stepping: the biomechanics of a preferred response among older adults. *Experimental aging research* **27,** 361–376 (2001).
47. Verghese, J., Ambrose, A. F., Lipton, R. B. & Wang, C. Neurological gait abnormalities and risk of falls in older adults. *Journal of neurology* **257,** 392–398 (2010).
48. Salonen, L. & Kivelä, S.-L. Eye diseases and impaired vision as possible risk factors for recurrent falls in the aged: a systematic review. *Current gerontology and geriatrics research* **2012** (2012).
49. Freeman, E. E., Munoz, B., Rubin, G. & West, S. K. Visual field loss increases the risk of falls in older adults: the Salisbury eye evaluation. *Investigative ophthalmology & visual science* **48,** 4445–4450 (2007).
50. Yogev-Seligmann, G., Hausdorff, J. M. & Giladi, N. The role of executive function and attention in gait. *Movement disorders: official journal of the Movement Disorder Society* **23,** 329–342 (2008).
51. Hausdorff, J. M., Herman, T., Baltadjieva, R., Gurevich, T. & Giladi, N. Balance and gait in older adults with systemic hypertension. *American Journal of Cardiology* **91,** 643–645 (2003).
52. Sanders, N. A. *et al.* Atrial fibrillation: an independent risk factor for nonaccidental falls in older patients. *Pacing and clinical electrophysiology* **35,** 973–979 (2012).
53. Leipzig, R. M., Cumming, R. G. & Tinetti, M. E. Drugs and falls in older people: a systematic review and meta-analysis: I. Psychotropic drugs. *Journal of the American Geriatrics Society* **47,** 30–39 (1999).
54. Berlie, H. D. & Garwood, C. L. Diabetes medications related to an increased risk of falls and fall-related morbidity in the elderly. *Annals of pharmacotherapy* **44,** 712–717 (2010).
55. Kelly, K. D. *et al.* Medication use and falls in community-dwelling older persons. *Age and ageing* **32,** 503–509 (2003).
56. Ensrud, K. E. *et al.* Central nervous system–active medications and risk for falls in older women. *Journal of the American Geriatrics Society* **50,** 1629–1637 (2002).

57. Boswell, E. B. & Stoudemire, A. Major depression in the primary care setting. *The American journal of medicine* **101,** 3S–9S (1996).
58. Wang, Y.-C. *et al.* Depression as a predictor of falls amongst institutionalized elders. *Aging & mental health* **16,** 763–770 (2012).
59. Paleacu, D. *et al.* Effects of pharmacological therapy on gait and cognitive function in depressed patients. *Clinical neuropharmacology* **30,** 63–71 (2007).
60. Menant, J. C., Steele, J. R., Menz, H. B., Munro, B. J. & Lord, S. R. Optimizing footwear for older people at risk of falls (2008).
61. Kochera, A. Falls among older persons and the role of the home: an analysis of cost, incidence, and potential savings from home modification. *Issue Brief (Public Policy Institute (American Association of Retired Persons)),* 1 (2002).
62. Gill, T. M., Williams, C. S. & Tinetti, M. E. Environmental hazards and the risk of nonsyncopal falls in the homes of community-living older persons. *Medical care,* 1174–1183 (2000).
63. Cormier, G. *Analyse statique et dynamique de cartes de profondeurs: application au suivi des personnes à risque sur leur lieu de vie* PhD thesis (Rennes 1, 2015).
64. Stokes, J. & Lindsay, J. Major causes of death and hospitalization in Canadian seniors. *Chronic Diseases in Canada* **17,** 63–73 (1996).
65. Kannus, P. *et al.* Epidemiology of hip fractures. *Bone* **18,** S57–S63 (1996).
66. Howland, J. *et al.* Covariates of fear of falling and associated activity curtailment. *The Gerontologist* **38,** 549–555 (1998).
67. Fessel, K. D. & Nevitt, M. C. Correlates of fear of falling and activity limitation among persons with rheumatoid arthritis. *Arthritis & Rheumatism: Official Journal of the American College of Rheumatology* **10,** 222–228 (1997).
68. Lachman, M. E. *et al.* Fear of falling and activity restriction: the survey of activities and fear of falling in the elderly (SAFE). *The Journals of Gerontology Series B: Psychological Sciences and Social Sciences* **53,** P43–P50 (1998).
69. Friedman, S. M., Munoz, B., West, S. K., Rubin, G. S. & Fried, L. P. Falls and fear of falling: which comes first? A longitudinal prediction model suggests strategies for primary and secondary prevention. *Journal of the American Geriatrics Society* **50,** 1329–1335 (2002).
70. Vellas, B. J. *et al.* One-leg balance is an important predictor of injurious falls in older persons. *Journal of the American Geriatrics Society* **45,** 735–738 (1997).
71. Florence, C. S. *et al.* Medical costs of fatal and nonfatal falls in older adults. *Journal of the American Geriatrics Society* **66,** 693–698 (2018).
72. Alexander, B. H., Rivara, F. P. & Wolf, M. E. The cost and frequency of hospitalization for fall-related injuries in older adults. *American journal of public health* **82,** 1020–1023 (1992).

73. Alekna, V., Stukas, R., Tamulaitytė-Morozovienė, I., Šurkienė, G. & Tamulaitienė, M. Self-reported consequences and healthcare costs of falls among elderly women. *Medicina* **51,** 57–62 (2015).
74. Public Health England. *Falls: applying All Our Health* 2017. https://www.gov.uk/government/publications/falls-applying-all-our-health/falls-applying-all-our-health (2020).
75. *Le coût des chutes des personnes âgées estimé à 2 milliards d'euros pour les collectivités — Silver Economie* https://www.silvereco.fr/le-cout-des-chutes-des-personnes-agees-estime-a-2-milliards-deuros-pour-les-collectivites/3157058 (2020).
76. Shimada, H., Uchiyama, Y. & Kakurai, S. Specific effects of balance and gait exercises on physical function among the frail elderly. *Clinical rehabilitation* **17,** 472–479 (2003).
77. Gillespie, L. D. *et al.* Interventions for preventing falls in elderly people. *Cochrane database of systematic reviews* (2003).
78. Delbaere, K. *et al.* A multifactorial approach to understanding fall risk in older people. *Journal of the American Geriatrics Society* **58,** 1679–1685 (2010).
79. Camp, N. J. & Slattery, M. L. Classification tree analysis: a statistical tool to investigate risk factor interactions with an example for colon cancer (United States). *Cancer Causes & Control* **13,** 813–823 (2002).
80. Lord, S. R., Menz, H. B. & Tiedemann, A. A physiological profile approach to falls risk assessment and prevention. *Physical therapy* **83,** 237–252 (2003).
81. Le Deun, P. & Gentric, A. L'évaluation gériatrique standardisée: intérêt et modalités. *Médecine thérapeutique* **10,** 229–236 (2004).
82. Jones, D. M., Song, X. & Rockwood, K. Operationalizing a frailty index from a standardized comprehensive geriatric assessment. *Journal of the American Geriatrics Society* **52,** 1929–1933 (2004).
83. Maeda, N., Kato, J. & Shimada, T. Predicting the probability for fall incidence in stroke patients using the Berg Balance Scale. *Journal of International Medical Research* **37,** 697–704 (2009).
84. Domıénguez-Carrillo, L. G., Arellano-Aguilar, G. & Leos-Zierold, H. Unipedal stance time and fall risk in the elderly. *Cirugia y cirujanos* **75,** 107–112 (2007).
85. Cohen, H., HEATON, L. G., CONGDON, S. L. & JENKINS, H. A. Changes in sensory organization test scores with age. *Age and ageing* **25,** 39–44 (1996).
86. Demura, S.-i. & Yamada, T. Simple and easy assessment of falling risk in the elderly by functional reach test using elastic stick. *The Tohoku journal of experimental medicine* **213,** 105–111 (2007).
87. Raıche, M., Hébert, R., Prince, F. & Corriveau, H. Screening older adults at risk of falling with the Tinetti balance scale. *The Lancet* **356,** 1001–1002 (2000).

88. Steffen, T. M., Hacker, T. A. & Mollinger, L. Age-and gender-related test performance in community-dwelling elderly people: Six-Minute Walk Test, Berg Balance Scale, Timed Up & Go Test, and gait speeds. *Physical therapy* **82,** 128–137 (2002).
89. Lundin-Olsson, L., Nyberg, L., Gustafson, Y., *et al.* Stops walking when talking as a predictor of falls in elderly people. *Lancet* **349,** 617 (1997).
90. Duong, T. V., Bui, H. H., Phung, D. Q. & Venkatesh, S. *Activity recognition and abnormality detection with the switching hidden semi-markov model* in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* **1** (2005), 838–845.
91. Blanke, D. J. & Hageman, P. A. Comparison of gait of young men and elderly men. *Physical Therapy* **69,** 144–148 (1989).
92. Toulotte, C., Thevenon, A., Watelain, E. & Fabre, C. Identification of healthy elderly fallers and non-fallers by gait analysis under dual-task conditions. *Clinical rehabilitation* **20,** 269–276 (2006).
93. Dorociak, R. & Cuddeford, T. Determining 3-D system accuracy for the VICON 370 system. *Gait & Posture* **3,** 88 (1995).
94. Jiang, S., Zhang, B. & Wei, D. *The elderly fall risk assessment and prediction based on gait analysis* in *2011 IEEE 11th international conference on computer and information technology* (2011), 176–180.
95. Senden, R., Savelberg, H., Grimm, B., Heyligers, I. & Meijer, K. Accelerometry-based gait analysis, an additional objective approach to screen subjects at risk for falling. *Gait & posture* **36,** 296–300 (2012).
96. https://www.profilmedecin.fr/. *Chiffres clés : Gériatre | Profil Médecin* https://www.profilmedecin.fr/contenu/chiffres-cles-medecin-geriatre/ (2020).
97. Chen, J., Kwong, K., Chang, D., Luk, J. & Bajcsy, R. *Wearable sensors for reliable fall detection* in *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference* (2006), 3551–3554.
98. Lombardi, A., Ferri, M., Rescio, G., Grassi, M. & Malcovati, P. *Wearable wireless accelerometer with embedded fall-detection logic for multi-sensor ambient assisted living applications* in *SENSORS, 2009 IEEE* (2009), 1967–1970.
99. Zhang, Z., Conly, C. & Athitsos, V. *A survey on vision-based fall detection* in *Proceedings of the 8th ACM international conference on PErvasive technologies related to assistive environments* (2015), 1–7.
100. Bourke, A., O'brien, J. & Lyons, G. Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. *Gait & posture* **26,** 194–199 (2007).
101. Kangas, M., Konttila, A., Lindgren, P., Winblad, I. & Jämsä, T. Comparison of low-complexity fall detection algorithms for body attached accelerometers. *Gait & posture* **28,** 285–291 (2008).

102. Bagalà, F. *et al.* Evaluation of accelerometer-based fall detection algorithms on real-world falls. *PloS one* **7,** e37062 (2012).
103. Bourke, A. K. & Lyons, G. M. A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. *Medical engineering & physics* **30,** 84–90 (2008).
104. Zhang, D., Wang, H., Wang, Y. & Ma, J. *Anti-fall: A non-intrusive and real-time fall detector leveraging CSI from commodity WiFi devices* in *International Conference on Smart Homes and Health Telematics* (2015), 181–193.
105. Palipana, S., Rojas, D., Agrawal, P. & Pesch, D. FallDeFi: Ubiquitous fall detection using commodity Wi-Fi devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* **1,** 1–25 (2018).
106. Jiang, H., Cai, C., Ma, X., Yang, Y. & Liu, J. Smart home based on WiFi sensing: A survey. *IEEE Access* **6,** 13317–13325 (2018).
107. Yang, X., Xiong, F., Shao, Y. & Niu, Q. WmFall: WiFi-based multistage fall detection with channel state information. *International Journal of Distributed Sensor Networks* **14,** 1550147718805718 (2018).
108. Liu, L. *et al. Automatic fall detection based on Doppler radar motion signature* in *2011 5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops* (2011), 222–225.
109. Su, B. Y., Ho, K., Rantz, M. J. & Skubic, M. Doppler radar fall activity detection using the wavelet transform. *IEEE Transactions on Biomedical Engineering* **62,** 865–875 (2014).
110. Amin, M. G., Zhang, Y. D., Ahmad, F. & Ho, K. D. Radar signal processing for elderly fall detection: The future for in-home monitoring. *IEEE Signal Processing Magazine* **33,** 71–80 (2016).
111. Jokanovic, B., Amin, M. & Ahmad, F. *Radar fall motion detection using deep learning* in *2016 IEEE radar conference (RadarConf)* (2016), 1–6.
112. Wu, Q., Zhang, Y. D., Tao, W. & Amin, M. G. Radar-based fall detection based on Doppler time–frequency signatures for assisted living. *IET Radar, Sonar & Navigation* **9,** 164–172 (2015).
113. Jokanović, B. & Amin, M. Fall detection using deep learning in range-Doppler radars. *IEEE Transactions on Aerospace and Electronic Systems* **54,** 180–189 (2017).
114. Chaccour, K., Darazi, R., el Hassans, A. H. & Andres, E. *Smart carpet using differential piezoresistive pressure sensors for elderly fall detection* in *2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)* (2015), 225–229.
115. Aud, M. A. *et al.* Smart Carpet: Developing a sensor system to detect falls and summon assistance. *Journal of gerontological nursing* **36,** 8–12 (2012).
116. Muheidat, F. & Tyrer, H. W. *Can we make a carpet smart enough to detect falls?* in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (2016), 5356–5359.

117. Litvak, D., Zigel, Y. & Gannot, I. *Fall detection of elderly through floor vibrations and sound* in *2008 30th annual international conference of the IEEE engineering in medicine and biology society* (2008), 4632–4635.
118. Zigel, Y., Litvak, D. & Gannot, I. A method for automatic fall detection of elderly people using floor vibrations and sound—Proof of concept on human mimicking doll falls. *IEEE transactions on biomedical engineering* **56,** 2858–2867 (2009).
119. Doukas, C. & Maglogiannis, I. *Advanced patient or elder fall detection based on movement and sound data* in *2008 Second International Conference on Pervasive Computing Technologies for Healthcare* (2008), 103–107.
120. Popescu, M., Li, Y., Skubic, M. & Rantz, M. *An acoustic fall detector system that uses sound height information to reduce the false alarm rate* in *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (2008), 4628–4631.
121. Li, Y., Ho, K. & Popescu, M. A microphone array system for automatic fall detection. *IEEE Transactions on Biomedical Engineering* **59,** 1291–1301 (2012).
122. Tajima, T., Abe, T. & Kimura, H. Development of fall detection system using ultrasound sensors. *IJTSM* **131,** 45–52 (2011).
123. Dura-Bernal, S. *et al. Human action categorization using ultrasound micro-doppler signatures* in *International Workshop on Human Behavior Understanding* (2011), 18–28.
124. Zhang, C. & Tian, Y. RGB-D camera-based daily living activity recognition. *Journal of computer vision and image processing* **2,** 12 (2012).
125. Panahi, L. & Ghods, V. Human fall detection using machine vision techniques on RGB–D images. *Biomedical Signal Processing and Control* **44,** 146–153 (2018).
126. Abobakr, A., Hossny, M., Abdelkader, H. & Nahavandi, S. *Rgb-d fall detection via deep residual convolutional lstm networks* in *2018 Digital Image Computing: Techniques and Applications (DICTA)* (2018), 1–7.
127. Stone, E. E. & Skubic, M. Fall detection in homes of older adults using the Microsoft Kinect. *IEEE journal of biomedical and health informatics* **19,** 290–301 (2014).
128. Gasparrini, S., Cippitelli, E., Spinsante, S. & Gambi, E. A depth-based fall detection system using a Kinect® sensor. *Sensors* **14,** 2756–2775 (2014).
129. Mastorakis, G. & Makris, D. Fall detection system using Kinect's infrared sensor. *Journal of Real-Time Image Processing* **9,** 635–646 (2014).
130. Kawatsu, C., Li, J. & Chung, C.-J. in *Robot Intelligence Technology and Applications 2012* 623–630 (Springer, 2013).
131. Vadivelu, S., Ganesan, S., Murthy, O. R. & Dhall, A. *Thermal imaging based elderly fall detection* in *Asian Conference on Computer Vision* (2016), 541–553.

132. Hayashida, A., Moshnyaga, V. & Hashimoto, K. *The use of thermal ir array sensor for indoor fall detection* in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (2017), 594–599.
133. Rafferty, J., Synnott, J., Nugent, C., Morrison, G. & Tamburini, E. in *Ubiquitous Computing and Ambient Intelligence* 84–90 (Springer, 2016).
134. Hayashida, A., Moshnyaga, V. & Hashimoto, K. *New approach for indoor fall detection by infrared thermal array sensor* in *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)* (2017), 1410–1413.
135. Sixsmith, A., Johnson, N. & Whatmore, R. *Pyroelectric IR sensor arrays for fall detection in the older population* in *Journal de Physique IV (Proceedings)* **128** (2005), 153–160.
136. Gade, R. & Moeslund, T. B. Thermal cameras and applications: a survey. *Machine vision and applications* **25,** 245–262 (2014).
137. Nogas, J., Khan, S. S. & Mihailidis, A. Deepfall: Non-invasive fall detection with deep spatio-temporal convolutional autoencoders. *Journal of Healthcare Informatics Research* **4,** 50–70 (2020).
138. Riquelme, F., Espinoza, C., Rodenas, T., Minonzio, J.-G. & Taramasco, C. eHomeSeniors Dataset: An Infrared Thermal Sensor Dataset for Automatic Fall Detection Research. *Sensors* **19,** 4565 (2019).
139. Kido, S., Miyasaka, T., Tanaka, T., Shimizu, T. & Saga, T. *Fall detection in toilet rooms using thermal imaging sensors* in *2009 IEEE/SICE International Symposium on System Integration (SII)* (2009), 83–88.
140. Epstein, H. *Medical alarm bracelet* US Patent 3,805,427. Apr. 1974.
141. Salvo, P. *et al.* A wearable sensor for measuring sweat rate. *IEEE Sensors Journal* **10,** 1557–1558 (2010).
142. Aziz, O. & Robinovitch, S. N. An analysis of the accuracy of wearable sensors for classifying the causes of falls in humans. *IEEE transactions on neural systems and rehabilitation engineering* **19,** 670–676 (2011).
143. Ranhotigmage, C. *Human activities & posture recognition: innovative algorithm for highly accurate detection rate: a thesis submitted in fulfilment of the requirements for the degree of Master of Engineering in Electronics & Computer Systems Engineering at Massey University, Palmerston North, New Zealand* PhD thesis (Massey University, 2013).
144. Shany, T., Redmond, S. J., Narayanan, M. R. & Lovell, N. H. Sensors-based wearable systems for monitoring of human movement and falls. *IEEE Sensors Journal* **12,** 658–670 (2011).
145. Leonov, V. Thermoelectric energy harvesting of human body heat for wearable sensors. *IEEE Sensors Journal* **13,** 2284–2291 (2013).
146. Tamura, T., Maeda, Y., Sekine, M. & Yoshida, M. Wearable photoplethysmographic sensors—past and present. *Electronics* **3,** 282–302 (2014).

147. Zhang, T. *et al. Sound based heart rate monitoring for wearable systems* in *2010 International Conference on Body Sensor Networks* (2010), 139–143.
148. Li, Q. *et al. Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information* in *2009 Sixth International Workshop on Wearable and Implantable Body Sensor Networks* (2009), 138–143.
149. Dai, J., Bai, X., Yang, Z., Shen, Z. & Xuan, D. Mobile phone-based pervasive fall detection. *Personal and ubiquitous computing* **14,** 633–643 (2010).
150. Chuo, Y. *et al.* Mechanically flexible wireless multisensor platform for human physical activity and vitals monitoring. *IEEE transactions on biomedical circuits and systems* **4,** 281–294 (2010).
151. Erol, B., Amin, M. G. & Boashash, B. *Range-Doppler radar sensor fusion for fall detection* in *2017 IEEE Radar Conference (RadarConf)* (2017), 0819–0824.
152. Van Dorp, P. & Groen, F. Feature-based human motion parameter estimation with radar. *IET Radar, Sonar & Navigation* **2,** 135–145 (2008).
153. Boulic, R., Thalmann, N. M. & Thalmann, D. A global human walking model with real-time kinematic personification. *The visual computer* **6,** 344–358 (1990).
154. Kim, Y. & Ling, H. Human activity classification based on micro-Doppler signatures using a support vector machine. *IEEE Transactions on Geoscience and Remote Sensing* **47,** 1328–1337 (2009).
155. Hall, M. *et al.* The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter* **11,** 10–18 (2009).
156. Solbach, M. D. & Tsotsos, J. K. *Vision-based fallen person detection for the elderly* in *Proceedings of the IEEE International Conference on Computer Vision Workshops* (2017), 1433–1442.
157. Dubois, A. & Charpillet, F. *A gait analysis method based on a depth camera for fall prevention* in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (2014), 4515–4518.
158. Tao, L. *et al.* Home Activity Monitoring using Low Resolution Infrared Sensor. *arXiv preprint arXiv:1811.05416* (2018).
159. Orlov, L. M. Technology for aging in place. *Aging In Place Technology Watch* (2012).
160. Im, I., Hong, S. & Kang, M. S. An international comparison of technology adoption: Testing the UTAUT model. *Information & management* **48,** 1–8 (2011).
161. Arning, K. & Ziefle, M. *"get that camera out of my house!" conjoint measurement of preferences for video-based healthcare monitoring systems in private and public places* in *International Conference on Smart Homes and Health Telematics* (2015), 152–164.
162. Green, P. E. & Srinivasan, V. Conjoint analysis in consumer research: issues and outlook. *Journal of consumer research* **5,** 103–123 (1978).

163. Erdélyi, A., Barát, T., Valet, P., Winkler, T. & Rinner, B. *Adaptive cartooning for privacy protection in camera networks* in *2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (2014), 44–49.
164. Abadi, M. *et al. Deep learning with differential privacy* in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (2016), 308–318.
165. Phan, N., Wang, Y., Wu, X. & Dou, D. *Differential privacy preservation for deep auto-encoders: an application of human behavior prediction* in *Thirtieth AAAI Conference on Artificial Intelligence* (2016).
166. Hajebi, K. & Zelek, J. S. *Structure from infrared stereo images* in *Computer and Robot Vision, 2008. CRV'08. Canadian Conference on* (2008), 105–112.
167. *Thermal Imaging Market worth 4.6 billion by 2025, growing with a CAGR of 6.2* https://www.marketsandmarkets.com/PressReleases/thermal-imaging.asp (2020).
168. Lin, S.-S. Extending visible band computer vision techniques to infrared band images (2001).
169. Goubet, E., Katz, J. & Porikli, F. *Pedestrian tracking using thermal infrared imaging* in *Infrared technology and applications XXXII* **6206** (2006), 62062C.
170. Maiti, A. & Sivanesan, S. *Cloud controlled intrusion detection and burglary prevention stratagems in home automation systems* in *Future Internet Communications (BCFIC), 2012 2nd Baltic Congress on* (2012), 182–186.
171. Bertozzi, M., Binelli, E., Broggi, A. & Rose, M. *Stereo vision-based approaches for pedestrian detection* in *Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on* (2005), 16–16.
172. Suard, F., Rakotomamonjy, A., Bensrhair, A. & Broggi, A. *Pedestrian detection using infrared images and histograms of oriented gradients* in *Intelligent Vehicles Symposium, 2006 IEEE* (2006), 206–212.
173. Bertozzi, M. *et al.* Pedestrian detection by means of far-infrared stereo vision. *Computer vision and image understanding* **106,** 194–204 (2007).
174. Pittaluga, F., Zivkovic, A. & Koppal, S. J. *Sensor-level privacy for thermal cameras* in *2016 IEEE International Conference on Computational Photography (ICCP)* (2016), 1–12.
175. Zoetgnandé, Y., Fougeres, A.-J., Cormier, G. & Dillenseger, J.-L. *Robust low resolution thermal stereo camera calibration* in *11th International Conference on Machine Vision (ICMV 2018)* (Munich, 2018), 1–5.
176. Bertozzi, M., Broggi, A., Lasagni, A., Del Rose, M. & Rose, M. *Infrared stereo vision-based pedestrian detection* in *IEEE Proceedings. Intelligent Vehicles Symposium* (2005), 24–29. ISBN: 0-7803-8961-1.

177. Dhua, A., Cutu, F., Hammoud, R. & Kiselewich, S. J. *Triangulation based technique for efficient stereo computation in infrared images* in *IEEE Intelligent Vehicles Symposium, Proceedings* (2003), 673–678.
178. Guennebaud, G., Jacob, B., *et al. Eigen v3* http://eigen.tuxfamily.org. 2010.
179. Zhang, K., Zuo, W. & Zhang, L. *Learning a single convolutional super-resolution network for multiple degradations* in *IEEE Conference on Computer Vision and Pattern Recognition* **6** (2018).
180. Brown, D. Decentering distortion of lenses, Photogrammetric Eng (1966).
181. Zhang, Z. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence* **22,** 1330–1334 (2000).
182. Strecha, C., Von Hansen, W., Van Gool, L., Fua, P. & Thoennessen, U. *On benchmarking camera calibration and multi-view stereo for high resolution imagery* in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (2008), 1–8.
183. Ursine, W. *et al. Thermal/visible autonomous stereo visio system calibration methodology for non-controlled environments* in *11th International Conference on Quantitative Infrared Thermography* (2012), 1–10.
184. Kong, W., Zhang, D., Wang, X., Xian, Z. & Zhang, J. *Autonomous landing of an UAV with a ground-based actuated infrared stereo vision system* in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on* (2013), 2963–2970.
185. Harguess, J. & Strange, S. *Infrared stereo calibration for unmanned ground vehicle navigation* in *Unmanned Systems Technology XVI* **9084** (2014), 90840S.
186. Shibata, T., Tanaka, M. & Okutomi, M. Accurate joint geometric camera calibration of visible and far-infrared cameras. *Electronic Imaging* **2017,** 7–13 (2017).
187. St-Laurent, L., Mikhnevich, M., Bubel, A. & Prévost, D. Passive calibration board for alignment of VIS-NIR, SWIR and LWIR images. *Quantitative InfraRed Thermography Journal* **14,** 193–205 (2017).
188. Rankin, A. *et al. Unmanned ground vehicle perception using thermal infrared cameras* in *Unmanned Systems Technology XIII* **8045** (2011), 804503.
189. Ellmauthaler, A., da Silva, E. A., Pagliari, C. L., Gois, J. N. & Neves, S. R. *A novel iterative calibration approach for thermal infrared cameras* in *Image Processing (ICIP), 2013 20th IEEE International Conference on* (2013), 2182–2186.
190. Yang, R., Yang, W., Chen, Y. & Wu, X. Geometric calibration of IR camera using trinocular vision. *Journal of Lightwave technology* **29,** 3797–3803 (2011).
191. Gschwandtner, M., Kwitt, R., Uhl, A. & Pree, W. *Infrared camera calibration for dense depth map construction* in *Intelligent Vehicles Symposium (IV), 2011 IEEE* (2011), 857–862.
192. Bradski, G. & Kaehler, A. *Learning OpenCV: Computer vision with the OpenCV library* (" O'Reilly Media, Inc.", 2008).

193. Otsu, N. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics* **9,** 62–66 (1979).
194. Olivia, A. & Torralba, A. Building the gist of a scence: the role of global image feaures in recognition. *Progress in Brain Research* **155,** 23–36 (2006).
195. Bertozzi, M. *et al. A Pedestrian Detector Using Histograms of Oriented Gradients and a Support Vector Machine Classifier* in *IEEE Intelligent Transportation Systems Conference* (2007), 143–148.
196. Olmeda, D., De La Escalera, A. & Armingol, J. M. *Contrast invariant features for human detection in far infrared images* in *2012 IEEE Intelligent Vehicles Symposium* (2012), 117–122.
197. Zhang, F., Liu, S.-q., Wang, D.-b. & Guan, W. Aircraft recognition in infrared image using wavelet moment invariants. *Image and Vision Computing* **27,** 313–318 (2009).
198. Dai, X. & Khorram, S. A feature-based image registration algorithm using improved chain-code representation combined with invariant moments. *IEEE Transactions on Geoscience and Remote Sensing* **37,** 2351–2362 (1999).
199. Watanabe, T., Ito, S. & Yokoi, K. in *3rd Pacific Rim Symposium on Advances in Image and Video Technology* 37–47 (Springer Berlin Heidelberg, 2009).
200. Iwata, S. & Enokida, S. *Object Detection Based on Multiresolution CoHOG* in *International Symposium on Visual Computing* (2014), 427–437.
201. Liu, X. & Fujimura, K. Pedestrian Detection Using Stereo Night Vision. *IEEE Transactions on Vehicular Technology* **53,** 1657–1665 (2004).
202. Bertozzi, M. *et al.* Pedestrian detection by means of far-infrared stereo vision. *Computer Vision and Image Understanding* **106,** 194–204. ISSN: 1077-3142 (May 2007).
203. Harris, C. G., Stephens, M., *et al. A combined corner and edge detector.* in *Alvey vision conference* **15** (1988), 10–5244.
204. Shi, J. *et al. Good features to track* in *1994 Proceedings of IEEE conference on computer vision and pattern recognition* (1994), 593–600.
205. Lucas, B. D., Kanade, T., *et al.* An iterative image registration technique with an application to stereo vision (1981).
206. Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision* **60,** 91–110 (2004).
207. Bay, H., Tuytelaars, T. & Van Gool, L. *Surf: Speeded up robust features* in *European conference on computer vision* (2006), 404–417.
208. Calonder, M. *et al.* BRIEF: Computing a local binary descriptor very fast. *IEEE transactions on pattern analysis and machine intelligence* **34,** 1281–1298 (2011).
209. Rublee, E., Rabaud, V., Konolige, K. & Bradski, G. R. *ORB: An efficient alternative to SIFT or SURF.* in *ICCV* **11** (2011), 2.

210. Rosin, P. L. Measuring corner properties. *Computer Vision and Image Understanding* **73,** 291–307 (1999).
211. Alcantarilla, P. F., Bartoli, A. & Davison, A. J. *KAZE features* in *European Conference on Computer Vision* (2012), 214–227.
212. Alcantarilla, P. F. & Solutions, T. Fast explicit diffusion for accelerated features in nonlinear scale spaces. *IEEE Trans. Patt. Anal. Mach. Intell* **34,** 1281–1298 (2013).
213. Yang, X. & Cheng, K.-T. *LDB: An ultra-fast feature for scalable augmented reality on mobile devices* in *2012 IEEE international symposium on mixed and augmented reality (ISMAR)* (2012), 49–57.
214. Mair, E., Hager, G. D., Burschka, D., Suppa, M. & Hirzinger, G. *Adaptive and generic corner detection based on the accelerated segment test* in *European conference on Computer vision* (2010), 183–196.
215. Aljalbout, E., Golkov, V., Siddiqui, Y., Strobel, M. & Cremers, D. Clustering with deep learning: Taxonomy and new methods. *arXiv preprint arXiv:1801.07648* (2018).
216. Al-Qatf, M., Lasheng, Y., Al-Habib, M. & Al-Sabahi, K. Deep learning approach combining sparse autoencoder with SVM for network intrusion detection. *IEEE Access* **6,** 52843–52856 (2018).
217. Ohta, Y. & Kanade, T. Stereo by Intra- and Inter-Scanline Search Using Dynamic Programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **7,** 139–154 (1985).
218. Brown, M. Z., Burschka, D. & Hager, G. D. Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **25,** 993–1008 (2003).
219. Saxena, A., Ng, E. Y. K. & Lim, S. T. Imaging modalities to diagnose carotid artery stenosis: progress and prospect. *Biomedical engineering online* **18,** 66 (2019).
220. Yang, J., Yu, K., Gong, Y. & Huang, T. *Linear spatial pyramid matching using sparse coding for image classification* in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (2009), 1794–1801.
221. Baumberg, A. *Reliable feature matching across widely separated views* in *IEEE Conference on Computer Vision and Pattern Recognition* **1** (2000), 774–781.
222. Dey, N., Ashour, A. S. & Althoupety, A. S. in *Recent Advances in Applied Thermal Imaging for Industrial Applications* 87–117 (IGI Global, 2017).
223. Saxena, A., Raman, V. & Ng, E. Study on methods to extract high contrast image in active dynamic thermography. *Quantitative InfraRed Thermography Journal,* 1–17 (2019).
224. Saxena, A. *et al.* Infrared (IR) Thermography-based Quantitative Parameters to Predict the Risk of Post-operative Cancerous Breast Resection Flap Necrosis. *Infrared Physics & Technology,* 103063 (2019).

225. Reddy, B. S. & Chatterji, B. N. An FFT-based technique for translation, rotation, and scale-invariant image registration. *IEEE transactions on image processing* **5,** 1266–1271 (1996).
226. Hajebi, K. & Zelek, J. S. *Structure from infrared stereo images* in *5th Canadian Conference on Computer and Robot Vision (CRV)* (2008), 105–112. ISBN: 9780769531533.
227. Delon, J. & Rougé, B. Small baseline stereovision. *Journal of Mathematical Imaging and Vision* **28,** 209–223 (2007).
228. Maier-Hein, L. *et al.* Optical techniques for 3D surface reconstruction in computer-assisted laparoscopic surgery. *Medical image analysis* **17,** 974–996 (2013).
229. Haller, I. & Nedevschi, S. Design of interpolation functions for subpixel-accuracy stereo-vision systems. *IEEE Transactions on image processing* **21,** 889–898 (2011).
230. Pritchett, P. & Zisserman, A. *Wide baseline stereo matching* in *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)* (1998), 754–760.
231. Tian, Q. & Huhns, M. N. Algorithms for subpixel registration. *Computer Vision, Graphics, and Image Processing* **35,** 220–233 (1986).
232. Smith, J.-P. C. & Volegov, D. B. *Processing of digital motion images* US Patent 10,078,905. Sept. 2018.
233. Haller, I. & Nedevschi, S. Design of interpolation functions for subpixel-accuracy stereo-vision systems. *IEEE Transactions on image processing* **21,** 889–898 (2012).
234. Miclea, V.-C., Vancea, C.-C. & Nedevschi, S. *New sub-pixel interpolation functions for accurate real-time stereo-matching algorithms* in *Intelligent Computer Communication and Processing (ICCP), 2015 IEEE International Conference on* (2015), 173–178.
235. Shi, C. *et al.* High-Accuracy Stereo Matching Based on Adaptive Ground Control Points. *IEEE Transactions on Image Processing* **24,** 1412–1423 (2015).
236. Stone, H. S., Orchard, M. T., Chang, E. C. & Martucci, S. A. A fast direct Fourier-based algorithm for subpixel registration of images. *IEEE Transactions on Geoscience and Remote Sensing* **39,** 2235–2243 (2001).
237. Takita, K., Aoki, T., Sasaki, Y., Higuchi, T. & Kobayashi, K. High-Accuracy Subpixel Image Registration Based on Phase-Only Correlation. *IEICE Transaction on Fundamentals of Electronics, Communications and Computer Sciences* **E86-A,** 1925–1934 (2003).
238. Kovesi, P. Phase congruency: A low-level image invariant. *Psychological Research-Psychologische Forschung* **64,** 136–148 (2000).
239. Kovesi, P. *et al.* Image features from phase congruency. *Videre: Journal of computer vision research* **1,** 1–26 (1999).
240. Kovesi, P. *Phase Congruency Detects Corners and Edges* in *Digital Image Computing: Techniques and Applications 2003* **1** (2003), 309–318.
241. Foroosh, H., Zerubia, J. B. & Berthod, M. Extension of phase correlation to subpixel registration. *IEEE Transactions on Image Processing* **11,** 188–200 (2002).

242. Nagashima, S., Aoki, T., Higuchi, T. & Kobayashi, K. *A Subpixel Image Matching Technique Using Phase-Only Correlation* in *2006 International Symposium on Intelligent Signal Processing and Communications, ISPACS'06* (IEEE, Dec. 2006), 701–704.
243. Abdi, H. Singular value decomposition (SVD) and generalized singular value decomposition. *Encyclopedia of measurement and statistics,* 907–912 (2007).
244. Ma, N., Sun, P. F., Men, Y. B., Men, C. G. & Li, X. A Subpixel Matching Method for Stereovision of Narrow Baseline Remotely Sensed Imagery. *Mathematical Problems in Engineering* **2017,** 1–14 (2017).
245. Wu, Z., Fuller, N., Theriault, D. & Betke, M. *A Thermal Infrared Video Benchmark for Visual Analysis* in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2014), 201–208.
246. Zoetgnande, Y. *Dataset low-resolution thermal images* https://www.dropbox.com/sh/b6is62skda8vmun/AAD6-pQpcthQO560DPjogZdNa?dl=0. Accessed: 2019-07-10. 2019.
247. Kaehler, A. & Bradski, G. *Learning OpenCV 3: computer vision in C++ with the OpenCV library* (" O'Reilly Media, Inc.", 2016).
248. Szeliski, R. in, 87–94 (Springer Science & Business Media, 2010).
249. Hajebi, K. & Zelek, J. S. *Dense surface from infrared stereo* in *2007 IEEE Workshop on Applications of Computer Vision (WACV)* (2007), 21–21.
250. Wolff, K. *et al. Point Cloud Noise and Outlier Removal for Image-Based 3D Reconstruction* in *2016 Fourth International Conference on 3D Vision (3DV)* (Oct. 2016), 118–127.
251. Gottlieb, D. & Shu, C.-W. On the Gibbs phenomenon and its resolution. *SIAM review* **39,** 644–668 (1997).
252. Irani, M. & Peleg, S. Improving resolution by image registration. *CVGIP: Graphical models and image processing* **53,** 231–239 (1991).
253. Stark, H. & Oskoui, P. High-resolution image recovery from image-plane arrays, using convex projections. *JOSA A* **6,** 1715–1726 (1989).
254. Freeman, W. T., Jones, T. R. & Pasztor, E. C. Example-based super-resolution. *IEEE Computer graphics and Applications* **22,** 56–65 (2002).
255. Yang, J., Wright, J., Huang, T. S. & Ma, Y. Image super-resolution via sparse representation. *IEEE transactions on image processing* **19,** 2861–2873 (2010).
256. Baker, S. & Kanade, T. Limits on super-resolution and how to break them. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24,** 1167–1183 (2002).
257. Schultz, R. R. & Stevenson, R. L. A Bayesian approach to image expansion for improved definition. *IEEE Transactions on Image Processing* **3,** 233–242 (1994).
258. Irani, M. & Peleg, S. *Super resolution from image sequences* in [*1990*] *Proceedings. 10th International Conference on Pattern Recognition* **2** (1990), 115–120.
259. Nasrollahi, K. & Moeslund, T. B. Super-resolution: A comprehensive survey. *Machine Vision and Applications* **25,** 1423–1468. ISSN: 14321769. http://link.springer.com/10.1007/s00138-014-0623-4 (Aug. 2014).

260. Hou, H. & Andrews, H. Cubic splines for image interpolation and digital filtering. *IEEE Transactions on acoustics, speech, and signal processing* **26,** 508–517 (1978).
261. Lehmann, T. M., Gonner, C. & Spitzer, K. Addendum: B-spline interpolation in medical image processing. *IEEE Transactions on Medical Imaging* **20,** 660–665 (2001).
262. Keys, R. Cubic convolution interpolation for digital image processing. *IEEE transactions on acoustics, speech, and signal processing* **29,** 1153–1160 (1981).
263. De Natale, F., Desoli, G. & Giusto, D. Adaptive least-squares bilinear interpolation (ALSBI): a new approach to image-data compression. *Electronics Letters* **29,** 1638–1640 (1993).
264. Lee, S.-W. & Paik, J. K. *Image interpolation using adaptive fast B-spline filtering* in *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing* **5** (1993), 177–180.
265. Jensen, K. & Anastassiou, D. Subpixel edge localization and the interpolation of still images. *IEEE transactions on Image Processing* **4,** 285–295 (1995).
266. Han, J.-K. & Baek, S.-U. Parametric cubic convolution scaler for enlargement and reduction of image. *IEEE transactions on Consumer Electronics* **46,** 247–256 (2000).
267. Unser, M., Aldroubi, A., Eden, M., *et al.* Fast B-spline transforms for continuous image representation and interpolation. *IEEE Transactions on pattern analysis and machine intelligence* **13,** 277–285 (1991).
268. Wang, Q. & Ward, R. K. A new orientation-adaptive interpolation method. *IEEE Transactions on Image Processing* **16,** 889–900 (2007).
269. Yang, S., Kim, Y. & Jeong, J. Fine edge-preserving technique for display devices. *IEEE Transactions on Consumer Electronics* **54,** 1761–1769 (2008).
270. Hong, K. P., Paik, J. K., Kim, H. J. & Lee, C. H. An edge-preserving image interpolation system for a digital camcorder. *IEEE Transactions on Consumer Electronics* **42,** 279–284 (1996).
271. Chen, M.-J., Huang, C.-H. & Lee, W.-L. A fast edge-oriented algorithm for image interpolation. *Image and Vision Computing* **23,** 791–798 (2005).
272. Sun, J., Xu, Z. & Shum, H.-Y. *Image super-resolution using gradient profile prior* in *2008 IEEE Conference on Computer Vision and Pattern Recognition* (2008), 1–8.
273. Kim, K. I. & Kwon, Y. Single-image super-resolution using sparse regression and natural image prior. *IEEE transactions on pattern analysis and machine intelligence* **32,** 1127–1133 (2010).
274. Efrat, N., Glasner, D., Apartsin, A., Nadler, B. & Levin, A. *Accurate blur models vs. image priors in single image super-resolution* in *Proceedings of the IEEE International Conference on Computer Vision* (2013), 2832–2839.

275. Baker, S. & Kanade, T. *Hallucinating faces* in *Proceedings Fourth IEEE international conference on automatic face and gesture recognition (Cat. No. PR00580)* (2000), 83–88.
276. Datsenko, D. & Elad, M. Example-based single document image super-resolution: a global MAP approach with outlier rejection. *Multidimensional Systems and Signal Processing* **18,** 103–121 (2007).
277. Capel, D. & Zisserman, A. *Super-resolution from multiple views using learnt image models* in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001* **2** (2001), II–II.
278. Liu, J., Qiao, J., Wang, X., *et al. Face hallucination based on independent component analysis* in *2008 IEEE International Symposium on Circuits and Systems* (2008), 3242–3245.
279. Liang, Y., Lai, J.-H., Xie, X. & Liu, W. *Face hallucination under an image decomposition perspective* in *2010 20th International Conference on Pattern Recognition* (2010), 2158–2161.
280. Hui, Z. & Lam, K.-M. *An efficient local-structure-based face-hallucination method* in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2012), 1265–1268.
281. Park, S. W. & Savvides, M. *Breaking the limitation of manifold analysis for super-resolution of facial images* in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07* **1** (2007), I–573.
282. Kumar, B. V. & Aravind, R. *Face hallucination using OLPP and kernel ridge regression* in *2008 15th IEEE International Conference on Image Processing* (2008), 353–356.
283. Dong, C., Loy, C. C., He, K. & Tang, X. Image Super-Resolution Using Deep Convolutional Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **38,** 295–307. ISSN: 0162-8828. http://ieeexplore.ieee.org/document/7115171/ (Feb. 2016).
284. Kim, J., Lee, J. K. & Lee, K. M. Accurate Image Super-Resolution Using Very Deep Convolutional Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **38,** 295–307. ISSN: 01628828. http://arxiv.org/abs/1511.04587 (Nov. 2015).
285. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
286. Zhang, K., Zuo, W., Chen, Y., Meng, D. & Zhang, L. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing* **26,** 3142–3155 (2017).
287. Zhang, K., Zuo, W., Gu, S. & Zhang, L. *Learning deep CNN denoiser prior for image restoration* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), 3929–3938.

288. Dong, C., Loy, C. C. & Tang, X. *Accelerating the super-resolution convolutional neural network* in *European conference on computer vision* (2016), 391–407.
289. Shi, W. *et al. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), 1874–1883.
290. Lim, B., Son, S., Kim, H., Nah, S. & Mu Lee, K. *Enhanced deep residual networks for single image super-resolution* in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (2017), 136–144.
291. He, K., Zhang, X., Ren, S. & Sun, J. *Deep residual learning for image recognition* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), 770–778.
292. Ahn, N., Kang, B. & Sohn, K.-A. *Fast, accurate, and lightweight super-resolution with cascading residual network* in *Proceedings of the European Conference on Computer Vision (ECCV)* (2018), 252–268.
293. Jiao, J., Tu, W.-C., He, S. & Lau, R. W. *Formresnet: Formatted residual learning for image restoration* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2017), 38–46.
294. Fan, Y. *et al. Balanced two-stage residual networks for image super-resolution* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2017), 161–168.
295. Mao, X., Shen, C. & Yang, Y.-B. *Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections* in *Advances in neural information processing systems* (2016), 2802–2810.
296. Ronneberger, O., Fischer, P. & Brox, T. *U-net: Convolutional networks for biomedical image segmentation* in *International Conference on Medical image computing and computer-assisted intervention* (2015), 234–241.
297. Kim, J., Kwon Lee, J. & Mu Lee, K. *Deeply-recursive convolutional network for image super-resolution* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), 1637–1645.
298. Tai, Y., Yang, J. & Liu, X. *Image super-resolution via deep recursive residual network* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), 3147–3155.
299. Tai, Y., Yang, J., Liu, X. & Xu, C. *Memnet: A persistent memory network for image restoration* in *Proceedings of the IEEE international conference on computer vision* (2017), 4539–4547.
300. Li, Z. *et al. Feedback network for image super-resolution* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), 3867–3876.
301. Wang, Z., Liu, D., Yang, J., Han, W. & Huang, T. *Deep networks for image super-resolution with sparse prior* in *Proceedings of the IEEE international conference on computer vision* (2015), 370–378.

302. Lai, W. S., Huang, J. B., Ahuja, N. & Yang, M. H. *Deep Laplacian pyramid networks for fast and accurate super-resolution* in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* **2017-Janua** (Oct. 2017), 5835–5843. ISBN: 9781538604571. http://arxiv.org/abs/1710.01992.
303. Tong, T., Li, G., Liu, X. & Gao, Q. *Image super-resolution using dense skip connections* in *Proceedings of the IEEE International Conference on Computer Vision* (2017), 4799–4807.
304. Haris, M., Shakhnarovich, G. & Ukita, N. *Deep back-projection networks for super-resolution* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), 1664–1673.
305. Ren, H., El-Khamy, M. & Lee, J. *Image super resolution based on fusing multiple convolution neural networks* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2017), 54–61.
306. Hu, Y., Gao, X., Li, J., Huang, Y. & Wang, H. Single image super-resolution via cascaded multi-scale cross network. *arXiv preprint arXiv:1802.08808* (2018).
307. Hui, Z., Wang, X. & Gao, X. *Fast and accurate single image super-resolution via information distillation network* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), 723–731.
308. Qiu, Y., Wang, R., Tao, D. & Cheng, J. *Embedded block residual network: A recursive restoration model for single-image super-resolution* in *Proceedings of the IEEE International Conference on Computer Vision* (2019), 4180–4189.
309. Choi, J.-S. & Kim, M. *A deep convolutional neural network with selection units for super-resolution* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2017), 154–160.
310. Zhang, Y. *et al. Image super-resolution using very deep residual channel attention networks* in *Proceedings of the European Conference on Computer Vision (ECCV)* (2018), 286–301.
311. Anwar, S. & Barnes, N. Densely residual laplacian super-resolution. *arXiv preprint arXiv:1906.12021* (2019).
312. Ledig, C. *et al. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network.* in *CVPR* **2** (2017), 4.
313. Sajjadi, M. S., Scholkopf, B. & Hirsch, M. *Enhancenet: Single image super-resolution through automated texture synthesis* in *Proceedings of the IEEE International Conference on Computer Vision* (2017), 4491–4500.
314. Wang, X. *et al. Esrgan: Enhanced super-resolution generative adversarial networks* in *Proceedings of the European Conference on Computer Vision (ECCV)* (2018), 0–0.
315. Jolicoeur-Martineau, A. The relativistic discriminator: a key element missing from standard GAN. *arXiv preprint arXiv:1807.00734* (2018).

316. Han, T. Y., Kim, Y. J. & Song, B. C. *Convolutional neural network-based infrared image super resolution under low light environment* in *Signal Processing Conference (EUSIPCO), 2017 25th European* (2017), 803–807.
317. He, Z. *et al.* Cascaded Deep Networks with Multiple Receptive Fields for Infrared Image Super-Resolution. *IEEE Transactions on Circuits and Systems for Video Technology*, 1–1. ISSN: 10518215. https://ieeexplore.ieee.org/document/8432397/ (2018).
318. Dong, W., Zhang, L., Shi, G. & Li, X. Non locally Centralized Sparse Representation for Image Restoration. *IEEE Transactions on Image Processing* **22,** 1620–1630. ISSN: 1057-7149. http://ieeexplore.ieee.org/document/6392274/ (Apr. 2013).
319. Riegler, G., Schulter, S., Ruther, M. & Bischof, H. *Conditioned Regression Models for Non-blind Single Image Super-Resolution* in *2015 IEEE International Conference on Computer Vision (ICCV)* (IEEE, Dec. 2015), 522–530. ISBN: 978-1-4673-8391-2. http://ieeexplore.ieee.org/document/7410424/.
320. Boracchi, G. & Foi, A. Modeling the Performance of Image Restoration From Motion Blur. *IEEE Transactions on Image Processing* **21,** 3502–3517. ISSN: 1057-7149. http://ieeexplore.ieee.org/document/6175123/ (Aug. 2012).
321. Sharifi, M., Fathy, M. & Mahmoudi, M. T. *A classified and comparative study of edge detection algorithms* in *Proceedings - International Conference on Information Technology: Coding and Computing, ITCC 2002* (2002), 117–120. ISBN: 0769515061. http://www.es.ele.tue.nl/~heco/courses/PlatformDesign2008/WiCa-assignment/01000371.pdf.
322. Gao, W., Zhang, X., Yang, L. & Liu, H. *An improved Sobel edge detection* in *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on* **5** (2010), 67–71.
323. Lipkin, B. S. *Picture Processing and Psychopictorics*. 535. ISBN: 0323146856. https://books.google.fr/books?hl=en&lr=&id=vp-w_pC9JBAC&oi=fnd&pg=PA75&dq=edge+extraction+prewitt&ots=szE6-ooCI4&sig=D2kYix_ohBGESfq7-TNletqvNbI#v=onepage&q=edge%20extraction%20prewitt&f=false (Elsevier Science, 1970).
324. Robinson, G. S. Color Edge Detection. *Optical Engineering* **16,** 165479. ISSN: 0091-3286. http://opticalengineering.spiedigitallibrary.org/article.aspx?doi=10.1117/12.7972120 (Oct. 1977).
325. Van Vliet, L. J., Young, I. T. & Beckers, G. L. A nonlinear laplace operator as edge detector in noisy images. *Computer Vision, Graphics, and Image Processing* **45,** 167–195. ISSN: 0734-189X. https://www.sciencedirect.com/science/article/pii/0734189X8990131X (Feb. 1989).
326. Tong, T., Li, G., Liu, X. & Gao, Q. *Image Super-Resolution Using Dense Skip Connections* in *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), 4809–4817. ISBN: 978-1-5386-1032-9. http://openaccess.thecvf.com/

content_ICCV_2017/papers/Tong_Image_Super-Resolution_Using_ICCV_2017_paper.pdf%20http://ieeexplore.ieee.org/document/8237776/.

327. Tai, Y., Yang, J., Liu, X. & Xu, C. *Memnet: A persistent memory network for image restoration* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), 4539–4547.
328. Shi, W. *et al. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), 1874–1883.
329. Sattar, F., Floreby, L., Salomonsson, G. & Lovstrom, B. Image enhancement based on a nonlinear multiscale method. *IEEE Transactions on Image Processing* **6,** 888–895. ISSN: 10577149. http://ieeexplore.ieee.org/document/585239/ (June 1997).
330. Adlakha, D., Adlakha, D. & Tanwar, R. Analytical Comparison between Sobel and Prewitt Edge Detection Techniques. *International Journal of Scientific & Engineering Research* (2016).
331. Fischler, M. A. & Bolles, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* **24,** 381–395 (1981).
332. Max Roser, C. A. & Ritchie, H. Human Height. *Our World in Data.* https://ourworldindata.org/hu height (2013).
333. Foroughi, H., Rezvanian, A. & Paziraee, A. *Robust fall detection using human shape and multi-class support vector machine* in *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing* (2008), 413–420.
334. Yang, L., Ren, Y. & Zhang, W. 3D depth image analysis for indoor fall detection of elderly people. *Digital Communications and Networks* **2,** 24–34 (2016).
335. Rougier, C., Auvinet, E., Rousseau, J., Mignotte, M. & Meunier, J. *Fall detection from depth map video sequences* in *International conference on smart homes and health telematics* (2011), 121–128.
336. Kwolek, B. & Kepski, M. Fuzzy inference-based fall detection using kinect and body-worn accelerometer. *Applied Soft Computing* **40,** 305–318 (2016).
337. Halima, I., Laferté, J.-M., Cormier, G., Fougère, A.-J. & Dillenseger, J.-L. Depth and thermal information fusion for head tracking using particle filter in a fall detection context. *Integrated Computer-Aided Engineering*, 1–14 (2020).
338. Romei, A. & Ruggieri, S. A multidisciplinary survey on discrimination analysis. *The Knowledge Engineering Review* **29,** 582–638 (2014).
339. Sixsmith, A. & Johnson, N. A smart sensor to detect the falls of the elderly. *IEEE Pervasive computing* **3,** 42–47 (2004).
340. Wong, W. K., Lim, H. L., Loo, C. K. & Lim, W. S. *Home alone faint detection surveillance system using thermal camera* in *2010 Second International Conference on Computer Research and Development* (2010), 747–751.

341. Han, J. & Bhanu, B. *Human activity recognition in thermal infrared imagery* in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops* (2005), 17–17.
342. Bhanu, B. & Han, J. *Kinematic-based human motion analysis in infrared sequences* in *Sixth IEEE Workshop on Applications of Computer Vision, 2002.(WACV 2002). Proceedings.* (2002), 208–212.
343. Pears, N. & Liang, B. *Ground plane segmentation for mobile robot visual navigation* in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)* **3** (2001), 1513–1518.
344. Zhou, J. & Li, B. *Robust ground plane detection with normalized homography in monocular sequences from a robot platform* in *2006 International Conference on Image Processing* (2006), 3017–3020.
345. Liang, B., Pears, N. & Chen, Z. *Affine height landscapes for monocular mobile robot obstacle avoidance* in *Proceedings of Intelligent Autonomous Systems* **8** (2004), 863–872.
346. Lin, C.-H., Jiang, S.-Y., Pu, Y.-J. & Song, K.-T. *Robust ground plane detection for obstacle avoidance of mobile robots using a monocular camera* in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2010), 3706–3711.
347. Low, T. & Manzanera, A. *Ground-plane classification for robot navigation: Combining multiple cues toward a visual-based learning system* in *2010 11th International Conference on Control Automation Robotics & Vision* (2010), 994–999.
348. Cormier, G., Laferté, J.-M., Carrault, G., Dillenseger, J.-L. & Gauthier, V. *Suivi d'individus dans des cartes de profondeurs par champ de Markov, dans un contexte de détection de chute* in *Gretsi 2015* (2015).
349. Zhang, Z., Liu, W., Metsis, V. & Athitsos, V. *A viewpoint-independent statistical method for fall detection* in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)* (2012), 3626–3630.
350. Civera, J., Davison, A. J. & Montiel, J. M. Inverse depth parametrization for monocular SLAM. *IEEE transactions on robotics* **24,** 932–945 (2008).
351. Mur-Artal, R. & Tardós, J. D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics* **33,** 1255–1262 (2017).
352. Nogas, J., Khan, S. S. & Mihailidis, A. DeepFall–Non-invasive Fall Detection with Deep Spatio-Temporal Convolutional Autoencoders. *arXiv preprint arXiv:1809.00977* (2018).
353. Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* **16,** 321–357 (2002).

354. Elkan, C. *The foundations of cost-sensitive learning* in *International joint conference on artificial intelligence* **17** (2001), 973–978.
355. Chawla, N. V., Lazarevic, A., Hall, L. O. & Bowyer, K. W. *SMOTEBoost: Improving prediction of the minority class in boosting* in *European conference on principles of data mining and knowledge discovery* (2003), 107–119.
356. Tukey, J. W. *Exploratory data analysis* (Reading, Mass., 1977).
357. Vishwanathan, S. & Murty, M. N. *SSVM: a simple SVM algorithm* in *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)* **3** (2002), 2393–2398.
358. Frigo, M. & Johnson, S. G. *FFTW: An adaptive software architecture for the FFT* in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)* **3** (1998), 1381–1384.
359. Guennebaud, G., Jacob, B., *et al.* Eigen. *URl: http://eigen. tuxfamily. org* (2010).
360. Garreta, R. & Moncecchi, G. *Learning scikit-learn: machine learning in python* (Packt Publishing Ltd, 2013).
361. Abadi, M. *et al. Tensorflow: A system for large-scale machine learning* in *12th Symposium on Operating Systems Design and Implementation* (2016), 265–283.
362. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
363. Dietterich, T. G. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation* **10,** 1895–1923 (1998).
364. Huang, H., Xu, H., Wang, X. & Silamu, W. Maximum F1-score discriminative training criterion for automatic mispronunciation detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **23,** 787–797 (2015).
365. Shorten, C. & Khoshgoftaar, T. M. A survey on image data augmentation for deep learning. *Journal of Big Data* **6,** 60 (2019).
366. Perez, L. & Wang, J. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621* (2017).
367. Roitberg, A., Somani, N., Perzylo, A., Rickert, M. & Knoll, A. *Multimodal human activity recognition for industrial manufacturing processes in robotic workcells* in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction* (2015), 259–266.
368. Meißner, C., Meixensberger, J., Pretschner, A. & Neumuth, T. Sensor-based surgical activity recognition in unconstrained environments. *Minimally Invasive Therapy & Allied Technologies* **23,** 198–205 (2014).
369. Robertson, N. & Reid, I. A general method for human activity recognition in video. *Computer Vision and Image Understanding* **104,** 232–248 (2006).
370. Luštrek, M. & Kaluža, B. Fall detection and activity recognition with machine learning. *Informatica* **33** (2009).

371. Aggarwal, J. K. & Ryoo, M. S. Human activity analysis: A review. *ACM Computing Surveys (CSUR)* **43,** 1–43 (2011).
372. Hu, Y. *et al. Action detection in complex scenes with spatial and temporal ambiguities* in *2009 IEEE 12th International Conference on Computer Vision* (2009), 128–135.
373. Qian, H., Mao, Y., Xiang, W. & Wang, Z. Recognition of human activities using SVM multi-class classifier. *Pattern Recognition Letters* **31,** 100–111 (2010).
374. Johansson, G. Visual motion perception. *Scientific American* **232,** 76–89 (1975).
375. Sheikh, Y., Sheikh, M. & Shah, M. *Exploring the space of a human action* in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1* **1** (2005), 144–149.
376. Yilmaz, A. & Shah, M. *Recognizing human actions in videos acquired by uncalibrated moving cameras* in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1* **1** (2005), 150–157.
377. Chomat, O. & Crowley, J. L. *Probabilistic recognition of activity using local appearance* in *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)* **2** (1999), 104–109.
378. Gorelick, L., Blank, M., Shechtman, E., Irani, M. & Basri, R. Actions as space-time shapes. *IEEE transactions on pattern analysis and machine intelligence* **29,** 2247–2253 (2007).
379. Laptev, I. On space-time interest points. *International journal of computer vision* **64,** 107–123 (2005).
380. Dollár, P., Rabaud, V., Cottrell, G. & Belongie, S. *Behavior recognition via sparse spatio-temporal features* in *2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance* (2005), 65–72.
381. Holte, M. B., Moeslund, T. B., Nikolaidis, N. & Pitas, I. *3D human action recognition for multi-view camera systems* in *2011 International conference on 3D imaging, modeling, processing, visualization and transmission* (2011), 342–349.
382. Klaser, A., Marszałek, M. & Schmid, C. *A spatio-temporal descriptor based on 3d-gradients* in (2008).
383. Laptev, I., Marszalek, M., Schmid, C. & Rozenfeld, B. *Learning realistic human actions from movies* in *2008 IEEE Conference on Computer Vision and Pattern Recognition* (2008), 1–8.
384. Yu, T.-H., Kim, T.-K. & Cipolla, R. *Real-time Action Recognition by Spatiotemporal Semantic and Structural Forests.* in *BMVC* **2** (2010), 6.
385. Wang, H. & Schmid, C. *Action recognition with improved trajectories* in *Proceedings of the IEEE international conference on computer vision* (2013), 3551–3558.
386. Dalal, N., Triggs, B. & Schmid, C. *Human detection using oriented histograms of flow and appearance* in *European conference on computer vision* (2006), 428–441.

387. Arandjelović, R. & Zisserman, A. *Three things everyone should know to improve object retrieval* in *2012 IEEE Conference on Computer Vision and Pattern Recognition* (2012), 2911–2918.
388. Yacoob, Y. & Black, M. J. Parameterized modeling and recognition of activities. *Computer Vision and Image Understanding* **73,** 232–247 (1999).
389. Veeraraghavan, A., Chellappa, R. & Roy-Chowdhury, A. K. *The function space of an activity* in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)* **1** (2006), 959–968.
390. Yamato, J., Ohya, J. & Ishii, K. *Recognizing human action in time-sequential images using hidden markov model.* in *CVPR* **92** (1992), 379–385.
391. Oliver, N. M., Rosario, B. & Pentland, A. P. A Bayesian computer vision system for modeling human interactions. *IEEE transactions on pattern analysis and machine intelligence* **22,** 831–843 (2000).
392. Brand, M., Oliver, N. & Pentland, A. *Coupled hidden Markov models for complex action recognition* in *Proceedings of IEEE computer society conference on computer vision and pattern recognition* (1997), 994–999.
393. Oliver, N., Horvitz, E. & Garg, A. *Layered representations for human activity recognition* in *Proceedings. Fourth IEEE International Conference on Multimodal Interfaces* (2002), 3–8.
394. Shi, Y., Huang, Y., Minnen, D., Bobick, A. & Essa, I. *Propagation networks for recognition of partially ordered sequential action* in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* **2** (2004), II–II.
395. Ivanov, Y. A. & Bobick, A. F. Recognition of visual activities and interactions by stochastic parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22,** 852–872 (2000).
396. Allen, J. F. Maintaining knowledge about temporal intervals. *Communications of the ACM* **26,** 832–843 (1983).
397. Allen, J. F. & Ferguson, G. Actions and events in interval temporal logic. *Journal of logic and computation* **4,** 531–579 (1994).
398. Nevatia, R., Hobbs, J. & Bolles, B. *An ontology for video event representation* in *2004 Conference on Computer Vision and Pattern Recognition Workshop* (2004), 119–119.
399. Ghanem, N., DeMenthon, D., Doermann, D. & Davis, L. *Representation and recognition of events in surveillance video using petri nets* in *2004 Conference on Computer Vision and Pattern Recognition Workshop* (2004), 112–112.
400. Siskind, J. M. Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic. *Journal of artificial intelligence research* **15,** 31–90 (2001).
401. Taylor, G. W., Fergus, R., LeCun, Y. & Bregler, C. *Convolutional learning of spatio-temporal features* in *European conference on computer vision* (2010), 140–153.

402. Ji, S., Xu, W., Yang, M. & Yu, K. 3D convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence* **35,** 221–231 (2012).
403. Wang, L., Qiao, Y. & Tang, X. *Action recognition with trajectory-pooled deep-convolutional descriptors* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), 4305–4314.
404. Simonyan, K. & Zisserman, A. *Two-stream convolutional networks for action recognition in videos* in *Advances in neural information processing systems* (2014), 568–576.
405. Crammer, K. & Singer, Y. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of machine learning research* **2,** 265–292 (2001).
406. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural computation* **9,** 1735–1780 (1997).
407. Donahue, J. *et al. Long-term recurrent convolutional networks for visual recognition and description* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), 2625–2634.
408. Wang, L., Xiong, Y., Wang, Z. & Qiao, Y. Towards good practices for very deep two-stream convnets. *arXiv preprint arXiv:1507.02159* (2015).
409. Tran, D., Bourdev, L., Fergus, R., Torresani, L. & Paluri, M. *Learning spatiotemporal features with 3d convolutional networks* in *Proceedings of the IEEE international conference on computer vision* (2015), 4489–4497.
410. Feichtenhofer, C., Pinz, A. & Zisserman, A. *Convolutional two-stream network fusion for video action recognition* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), 1933–1941.
411. Yue-Hei Ng, J. *et al. Beyond short snippets: Deep networks for video classification* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), 4694–4702.
412. Wang, L. *et al. Temporal segment networks: Towards good practices for deep action recognition* in *European conference on computer vision* (2016), 20–36.
413. Bilen, H., Fernando, B., Gavves, E., Vedaldi, A. & Gould, S. *Dynamic image networks for action recognition* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), 3034–3042.
414. Carreira, J. & Zisserman, A. *Quo vadis, action recognition? a new model and the kinetics dataset* in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), 6299–6308.
415. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. *Rethinking the inception architecture for computer vision* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), 2818–2826.
416. Zach, C., Pock, T. & Bischof, H. *A duality based approach for realtime tv-l 1 optical flow* in *Joint pattern recognition symposium* (2007), 214–223.

417. Wang, L., Xiong, Y., Lin, D. & Van Gool, L. *Untrimmednets for weakly supervised action recognition and detection* in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (2017), 4325–4334.
418. Wu, C.-Y. *et al. Compressed video action recognition* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), 6026–6035.
419. Le Gall, D. MPEG: A video compression standard for multimedia applications. *Communications of the ACM* **34,** 46–58 (1991).
420. Piergiovanni, A., Angelova, A., Toshev, A. & Ryoo, M. S. *Evolving space-time neural architectures for videos* in *Proceedings of the IEEE international conference on computer vision* (2019), 1793–1802.
421. Piergiovanni, A. & Ryoo, M. *Temporal gaussian mixture layer for videos* in *International Conference on Machine Learning* (2019), 5152–5161.
422. Ryoo, M. S., Piergiovanni, A., Tan, M. & Angelova, A. Assemblenet: Searching for multi-stream neural connectivity in video architectures. *arXiv preprint arXiv:1905.13209* (2019).
423. Feichtenhofer, C., Fan, H., Malik, J. & He, K. *Slowfast networks for video recognition* in *Proceedings of the IEEE international conference on computer vision* (2019), 6202–6211.
424. Wang, L., Koniusz, P. & Huynh, D. Q. *Hallucinating idt descriptors and i3d optical flow features for action recognition with cnns* in *Proceedings of the IEEE International Conference on Computer Vision* (2019), 8698–8708.
425. Tran, D. *et al. A closer look at spatiotemporal convolutions for action recognition* in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (2018), 6450–6459.
426. Wang, Y., Long, M., Wang, J. & Yu, P. S. *Spatiotemporal pyramid network for video action recognition* in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (2017), 1529–1538.
427. Goodfellow, I. *et al. Generative adversarial nets* in *Advances in neural information processing systems* (2014), 2672–2680.
428. Hertzmann, A., Jacobs, C. E., Oliver, N., Curless, B. & Salesin, D. H. *Image analogies* in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), 327–340.
429. Isola, P., Zhu, J.-Y., Zhou, T. & Efros, A. A. *Image-to-image translation with conditional adversarial networks* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), 1125–1134.
430. Radford, A., Metz, L. & Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015).
431. Chen, T., Zhu, J.-Y., Shamir, A. & Hu, S.-M. Motion-aware gradient domain video composition. *IEEE Transactions on Image Processing* **22,** 2532–2544 (2013).

432. Wexler, Y., Shechtman, E. & Irani, M. *Space-time video completion* in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* **1** (2004), I–I.
433. Vondrick, C., Pirsiavash, H. & Torralba, A. *Generating videos with scene dynamics* in *Advances in neural information processing systems* (2016), 613–621.
434. Tulyakov, S., Liu, M.-Y., Yang, X. & Kautz, J. *Mocogan: Decomposing motion and content for video generation* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), 1526–1535.
435. Wang, T.-C. *et al.* Video-to-video synthesis. *arXiv preprint arXiv:1808.06601* (2018).
436. Muandet, K., Balduzzi, D. & Schölkopf, B. *Domain generalization via invariant feature representation* in *International Conference on Machine Learning* (2013), 10–18.
437. Ganin, Y. *et al.* Domain-adversarial training of neural networks. *The Journal of Machine Learning Research* **17,** 2096–2030 (2016).
438. Li, H., Jialin Pan, S., Wang, S. & Kot, A. C. *Domain generalization with adversarial feature learning* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), 5400–5409.
439. Everingham, M., Van Gool, L., Williams, C. K., Winn, J. & Zisserman, A. The pascal visual object classes (voc) challenge. *International journal of computer vision* **88,** 303–338 (2010).
440. Russell, B. C., Torralba, A., Murphy, K. P. & Freeman, W. T. LabelMe: a database and web-based tool for image annotation. *International journal of computer vision* **77,** 157–173 (2008).
441. Fei-Fei, L., Fergus, R. & Perona, P. *Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories* in *2004 conference on computer vision and pattern recognition workshop* (2004), 178–178.
442. Choi, M. J., Lim, J. J., Torralba, A. & Willsky, A. S. *Exploiting hierarchical context on a large database of object categories* in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2010), 129–136.
443. Ben-David, S. *et al.* A theory of learning from different domains. *Machine learning* **79,** 151–175 (2010).
444. Kifer, D., Ben-David, S. & Gehrke, J. *Detecting change in data streams* in *VLDB* **4** (2004), 180–191.
445. Ben-David, S., Blitzer, J., Crammer, K. & Pereira, F. *Analysis of representations for domain adaptation* in *Advances in neural information processing systems* (2007), 137–144.
446. Zhao, H. *et al. Adversarial multiple source domain adaptation* in *Advances in neural information processing systems* (2018), 8559–8570.

447. Khosla, A., Zhou, T., Malisiewicz, T., Efros, A. A. & Torralba, A. *Undoing the damage of dataset bias* in *European Conference on Computer Vision* (2012), 158–171.
448. Li, D., Yang, Y., Song, Y.-Z. & Hospedales, T. M. *Deeper, broader and artier domain generalization* in *Proceedings of the IEEE international conference on computer vision* (2017), 5542–5550.
449. Shankar, S. *et al.* Generalizing across domains via cross-gradient training. *arXiv preprint arXiv:1804.10745* (2018).
450. Volpi, R. *et al. Generalizing to unseen domains via adversarial data augmentation* in *Advances in neural information processing systems* (2018), 5334–5344.
451. Li, D., Yang, Y., Song, Y.-Z. & Hospedales, T. M. Learning to generalize: Meta-learning for domain generalization. *arXiv preprint arXiv:1710.03463* (2017).
452. Palmero, C. *et al.* Multi-modal rgb–depth–thermal human body segmentation. *International Journal of Computer Vision* **118,** 217–239 (2016).
453. Hwang, S., Park, J., Kim, N., Choi, Y. & So Kweon, I. *Multispectral pedestrian detection: Benchmark dataset and baseline* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), 1037–1045.
454. González, A. *et al.* Pedestrian detection at day/night time with visible and FIR cameras: A comparison. *Sensors* **16,** 820 (2016).
455. Davis, J. W. & Keck, M. A. *A two-stage template approach to person detection in thermal imagery* in *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05)-Volume 1* **1** (2005), 364–369.
456. Torabi, A., Massé, G. & Bilodeau, G.-A. An iterative integrated framework for thermal–visible image registration, sensor fusion, and people tracking for video surveillance applications. *Computer Vision and Image Understanding* **116,** 210–221 (2012).
457. Hadfield, S., Bowden, R. & Lebeda, K. The visual object tracking VOT2016 challenge results. *Lecture Notes in Computer Science* **9914,** 777–823 (2016).
458. Kristan, M. *et al. The visual object tracking vot2017 challenge results* in *Proceedings of the IEEE international conference on computer vision workshops* (2017), 1949–1972.
459. Wu, Y., Lim, J. & Yang, M.-H. *Online object tracking: A benchmark* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2013), 2411–2418.
460. Portmann, J., Lynen, S., Chli, M. & Siegwart, R. *People detection and tracking from aerial thermal views* in *2014 IEEE international conference on robotics and automation (ICRA)* (2014), 1794–1800.
461. Gade, R., Jorgensen, A. & Moeslund, T. B. *Long-term occupancy analysis using graph-based optimisation in thermal imagery* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2013), 3698–3705.
462. Gao, C. *et al.* Infar dataset: Infrared action recognition at different times. *Neurocomputing* **212,** 36–47 (2016).

463. Paszke, A. *et al. Pytorch: An imperative style, high-performance deep learning library* in *Advances in neural information processing systems* (2019), 8026–8037.
464. Pérez, J. S., Meinhardt-Llopis, E. & Facciolo, G. TV-L1 optical flow estimation. *Image Processing On Line* **2013,** 137–150 (2013).
465. Maaten, L. v. d. & Hinton, G. Visualizing data using t-SNE. *Journal of machine learning research* **9,** 2579–2605 (2008).
466. Chen, L.-C., Papandreou, G., Schroff, F. & Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587* (2017).
467. Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y. & Girshick, R. *Detectron2* https://github.com/facebookresearch/detectron2. 2019.

UNIVERSITÉ DE RENNES 1

Titre : Détection des chutes et reconnaissance d'activité à l'aide de l'imagerie thermique stéréo basse résolution

**Mot clés :** Détection de chute ; Reconnaissance d'activité ; Apprentissage profond ; Généralisation de domaine ; Thermique

**Résumé :** De nos jours, il est important de trouver des solutions pour détecter et prévenir les chutes des personnes âgées. Nous avons proposé un dispositif bas-coût à base d'une paire de capteurs thermiques. La contrepartie de ces capteurs bas-coût est leur faible résolution (80x60 pixels), la faible fréquence de rafraîchissement, le bruit et des effets de halo. Nous avons donc proposé quelques approches pour contourner ces inconvénients. Tout d'abord, nous avons proposé une nouvelle méthode de calibration avec une grille adaptée à l'image thermique et une méthodologie assurant la robustesse de l'estimation des paramètres malgré la faible résolution. Ensuite, pour la vision 3D, nous avons proposé une méthode de mise en correspondance stéréo avec une précision sous-pixels (appelée ST pour Subpixel Thermal) composée : 1) d'une méthode robuste d'extraction des caractéristiques basée sur la congruence de phase, 2) d'une mise en correspondance de ces caractéristiques au pixel près, et 3) d'une mise correspondance raffinée en précision sous-pixel basée sur la corrélation de phase locale. Nous avons également proposé une méthode de super-résolution appelée Edge Focused Thermal Super-Resolution (EFTS) qui contient un module d'extraction de contours amenant le réseau de neurones artificiels de se concentrer sur les contours des objets dans les images. Par la suite, pour la détection des chutes, nous avons proposé une nouvelle méthode (TSFD pour Thermal Stereo Fall Détection) basée sur les correspondances stéréo mais sans calibration et un apprentissage de points au sol. Enfin, pour la surveillance des activités des personnes âgées, nous avons exploré de nombreuses approches basées sur l'apprentissage profond pour classer des activités avec une quantité limitée de données d'apprentissage.

Title: Fall detection and activity recognition using stereo low-resolution thermal imaging

**Keywords:** Fall detection; Activity recognition; Deep learning; Domain generalization; Thermal

**Abstract:** Nowadays, it is essential to find solutions to detect and prevent the falls of seniors. We proposed a low-cost device based on a pair of thermal sensors. The counterpart of these low-cost sensors is their low resolution (80x60 pixels), low refresh rate, noise, and halo effects. We proposed some approaches to bypass these drawbacks. First, we proposed a calibration method with a grid adapted to the thermal image and a framework ensuring the robustness of the parameters estimation despite the low resolution. Then, for 3D vision, we proposed a threefold sub-pixel stereo matching framework (called ST for Subpixel Thermal): 1) robust features extraction method based on phase congruency, 2) matching of these features in pixel precision, and 3) refined matching in sub-pixel accuracy based on local phase correlation. We also proposed a super-resolution method called Edge Focused Thermal Super-resolution (EFTS), which includes an edge extraction module enforcing the neural networks to focus on the edge in images. After that, for fall detection, we proposed a new method (called TSFD for Thermal Stereo Fall Detection) based on stereo point matching but without calibration and the classification of matches as on the ground or not on the ground. Finally, we explored many approaches to learn activities from a limited amount of data for seniors activity monitoring.