Tackling pedestrian detection in large scenes with multiple views and representations

Nicola Pellicanò

NNT : 2018SACLS608

Thèse de doctorat

Tackling pedestrian detection in large scenes with multiple views and representations

Thèse de doctorat de l'Université Paris-Saclay
Préparée à l'Université Paris-Sud

École doctorale n°**580**
Sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat : Traitement du Signal et des Images

Thèse présentée et soutenue à Gif-sur-Yvette, le 21 Décembre 2018, par

Nicola Pellicanò

Composition du Jury :

Isabelle Bloch Professeur des Universités, Télécom ParisTech (LTCI)	Président
Thierry Denœux Professeur des Universités, Université de Technologie de Compiègne	Rapporteur
Jean-Luc Dugelay Professeur des Universités, EURECOM	Rapporteur
Alexandre Alahi Professeur assistant, École Polytechnique Fédérale de Lausanne	Examinateur
Pascal Monasse Professeur, IMAGINE, École des Ponts ParisTech	Examinateur
Sylvie Le Hégarat-Mascle Professeur des Universités, Université Paris-Saclay (SATIE)	Directrice de thèse
Emanuel Aldea Maître de Conférences, Université Paris-Saclay (SATIE)	Co-encadrant

Abstract

Pedestrian detection and tracking have become important fields in Computer Vision research, due to their implications for many applications, e.g. surveillance, autonomous cars, robotics. Pedestrian detection in high density crowds is a natural extension of such research body, and has a growing interest since large scale events (e.g. concerts, sport events, public ceremonies) are, nowadays, critical scenarios from a safety point of view. The ability to track each pedestrian independently in a dense crowd has multiple applications: study of human social behavior under high densities; detection of anomalies (e.g. a pedestrian exhibits dynamics different from the rest of the crowd); large event infrastructure planning (e.g. study of bottleneck accesses and exits from the event area). On the other hand, high density crowds introduce novel problems to the detection task. First, clutter and occlusion problems are taken to the extreme, so that only heads are visible, and they are not easily separable from the moving background. Second, heads are usually small (they have a diameter of typically less than ten pixels) and with little or no textures. This comes out from two independent constraints, the need of one camera to have a field of view as high as possible (in order to cover a larger crowd area), and the need of anonymization, i.e. the pedestrians must be not identifiable because of privacy concerns.

In this work we develop a complete framework in order to handle the pedestrian detection and tracking problems under the presence of the novel difficulties that they introduce, by using multiple cameras, in order to implicitly handle the high occlusion issues.

As a first contribution, we propose a robust method for camera pose estimation in surveillance environments. We handle problems as high distances between cameras, large perspective variations, and scarcity of matching information, by exploiting an entire video stream to perform the calibration, in such a way that it exhibits fast convergence to a good solution. Moreover, we are concerned not only with a global fitness of the solution, but also with reaching low local errors, which is sought for when dealing with small objects like the pedestrian heads.

As a second contribution, we propose an unsupervised multiple camera detection method which exploits the visual consistency of pixels between multiple views in order to estimate the presence of a pedestrian. After a fully automatic metric registration of the scene, one is capable of jointly estimating the presence of a pedestrian and its height, allowing for the projection of detections on a common ground plane, and thus allowing for 3D tracking, which can be much more robust with respect to typical problems specific to image space based tracking like crossing or dead tracks.

In the third part, we study different methods in order to perform supervised pedestrian detection on single views. Specifically, we aim to build a dense pedestrian segmentation of the scene starting from spatially imprecise labeling of data, i.e. heads centers instead of full head contours, since their extraction is unfeasible in a dense crowd. Most notably, deep architectures for semantic segmentation are studied and adapted to the problem of small head detection in cluttered environments.

As last but not least contribution, we propose a novel framework in order to perform efficient information fusion in 2D spaces. The final aim is to perform multiple sensor fusion (supervised detectors on each view, and an unsupervised detector on multiple views) at ground plane level, that is, thus, our discernment frame. Since the space complexity of such discernment frame is very large, we propose an efficient compound hypothesis representation which has been shown to be invariant to the scale of the search space. Through such representation, we are capable of

defining efficient basic operators and combination rules of Belief Function Theory. Furthermore, we propose a complementary graph based description of the relationships between compound hypotheses (i.e. intersections and inclusion), in order to perform efficient algorithms for, e.g. high level decision making.

Finally, we demonstrate our information fusion approach both at a spatial level, i.e. between detectors of different natures, and at a temporal level, by performing evidential tracking of pedestrians on real large scale scenes in sparse and dense conditions.

Résumé

La détection et le suivi de piétons sont devenus des thèmes phares en recherche en Vision Artificielle, car ils sont impliqués dans de nombreuses applications, comme la surveillance, les voitures autonomes, ou la robotique. La détection de piétons dans des foules très denses est une extension naturelle de ce domaine de recherche, et l'intérêt croissant pour ce problème est lié aux évènements de grande envergure (concerts, évènements sportifs, cérémonies publiques) qui sont, de nos jours, des scenarios à risque d'un point de vue de la sûreté publique. Le suivi individuel des piétons dans une foule dense permettra : l'étude du comportement social humain à des densités élevées, la détection des anomalies (par exemple, un piéton qui montre une dynamique de mouvement différente du reste de la foule), la conception des infrastructures (par exemple, pour limiter les embouteillages à l'entrée et à la sortie de la zone dédiée à un évènement). Par ailleurs, les foules très denses soulèvent des problèmes inédits pour la tâche de détection. Les problèmes d'occultation deviennent prépondérants avec des individus dont seules les têtes sont visibles, tout en n'étant pas facilement séparables de l'arrière-plan. Par ailleurs, de par le fait que les caméras ont le champ de vision le plus grand possible pour couvrir au mieux la foule, et la contrainte d'« anonymization » des individus (devant être non identifiables pour des raisons de respect de la vie privée), les têtes sont généralement très petites (diamètre de l'ordre de dix pixels, voire inférieur) et non texturées.

Dans ce manuscrit nous présentons un système complet pour traiter les problèmes de détection et de suivi en présence des difficultés spécifiques à ce contexte. Ce système utilise plusieurs caméras, pour gérer les problèmes de forte occultation.

Comme première contribution, nous proposons une méthode robuste pour l'estimation de la position relative entre plusieurs caméras dans le cas des environnements requérant une surveillance. Ces environnements soulèvent des problèmes comme la grande distance entre les caméras, le fort changement de perspective, et la pénurie d'information en commun. Nous avons alors proposé d' exploiter le flot vidéo pour effectuer la calibration, avec l'objectif d'obtenir une convergence rapide vers une solution globale de bonne qualité. Nous avons montré que la solution proposée permettait également de réduire le niveau des erreurs locales, ce qui est un atout fondamental en vue de la mise en correspondance des objets petits, comme les têtes des piétons.

Comme deuxième contribution, nous proposons une mèthode non supervisée pour la détection des piétons avec plusieurs caméras, qui exploite la consistance visuelle des pixels à partir des différents points de vue. Suite à un recalage métrique de la scéne qui est entièrement automatisé, notre approche estime conjointement la présence d'un piéton et sa hauteur, ce qui nous permet d'effectuer la projection de l'ensemble des détections sur le plan du sol, et donc de passer à un suivi 3D, qui est plus robuste face à des problèmes typiques de suivi dans l'espace de l'image, comme le croisement et la disparition des trajectoires. Dans une troisième partie, nous revenons sur la détection supervisée des piétons dans chaque caméra indépendamment en vue de l'améliorer. L'objectif est alors d'effectuer la segmentation des piétons dans la scène en partant d'une labélisation imprécise (spatialement) des données d'apprentissage, par exemple centres des têtes à la place des contours, car leur extraction précise est impossible dans des foules denses. En particulier, nous nous sommes intéressés aux architectures de réseaux profonds (deep learning) pour la segmentation sémantique and nous en avons proposé une adaptation au problème de détection de petites têtes dans des environnements difficiles.

Comme dernière contribution, nous proposons un cadre formel original pour une fusion de

données efficace dans des espaces 2D. L'objectif est d'effectuer la fusion entre différents capteurs (détecteurs supervisés en chaque caméra et détecteur non supervisé en multi-vues) sur le plan du sol, qui représente notre cadre de discernement. Selon une approche naïve, la complexité de ce cadre de discernement est liée aux dimensions et à la résolution spatiales de la région à surveiller, soit trop grande pour être envisageable dans le cadre de la théorie des fonctions de croyance. Pour travailler sous ce cadre qui permet de modéliser à la fois l'incertitude et l'imprécision de nos détections, nous avons proposé une représentation efficace des hypothèses composées (disjonctions d'hypothèses élémentaires) qui est invariante au changement de résolution de l'espace de recherche. Avec cette représentation, nous sommes capables de définir des opérateurs de base et des règles de combinaison efficaces pour combiner les fonctions de croyance. En plus de la représentation des hypothèses elles-mêmes, nous avons également proposé une nouvelle représentation des relations entre les hypothèses (notamment intersection et inclusion) sous forme de graphe. Cette dernière nous a alors permis de proposer des versions efficaces d'algorithmes pour, par exemple, la prise de décision.

Enfin, notre approche de fusion de données a été évaluée à la fois au niveau spatial, c'est à dire en combinant des détecteurs de nature différente, et au niveau temporel, en faisant du suivi évidentiel de piétons sur de scènes à grande échelle dans des conditions de densité variable (faible et élevée).

Introduction

Context and social issues

The rise of industrialization, the intensive urbanization, which have led to the formation of megalopolises, have revolutionized our society. Such trend is predicted to persist, since the world population is projected to grow from 7 to 9 billion by 2050, and the developed world will be urbanized at a degree of 86% by the same year. The increase of the size of metropolitan areas has the effect of an increased requirement of public transports and well designed infrastructures (see Figure 1). Dealing with the risks and difficulties associated with such problems is one of the key challenges for the upcoming future. All the risks associated with transportation and urban planning have a common originating factor: the behavior of human crowds. The ability to study the dynamics and predict the behavior of humans forming crowds is a priority for any of the challenges introduced by urbanization.

One of the most critical problems is represented by panic stampedes, which are a major concern during mass events, and many lives are lost every year due to such issue. As an example, we cite the stampede at Mecca, Saudi Arabia in 2006, where 363 people died. The authors of [65] have studied the videos of such specific case, and have discovered that a change in the flow of the crowd (stop-and-go waves) was observable as early as 30 minutes before the accident. They have also noticed that, as the crowd density increased, few minutes before the tragedy, the flow started to be irregular, with people displacing into all possible directions. The pedestrians were moved by the crowd, and they could not stop, and the individuals who fell down were not able to get on their feet anymore. Such observations teach a fundamental lesson: a video analysis of the crowd behavior could have detected the risk of stampedes in advance, making possible the implementation of corrective measures in order to prevent the accident from happening.

a) b)

Figure 1: High density contexts: a) Paris marathon starting block b) CCTV image from one of the monitoring systems in Mecca. High density events become ubiquitous, but they always involve a high risk of instabilities.

Beyond these exceptional tragic events, the correct design of infrastructures is an every day concern, since it has a big impact on the quality of living in terms of delays, stress and discomfort. In terms of planning, it is common practice to use simulations as a support in order to model the

pedestrian social interactions. However, as the crowd density increases, simulation models start to be inadequate for behavior analysis.

When recalling the interest of advancing this field, crowds can be studied for three different objectives.

1. **Prevention**. Use of real data estimates in order to validate and calibrate the crowd simulations, to assess the quality of the infrastructure and for altering the design of the areas which have to sustain a high crowd flow.

2. **Monitoring**. Use of real data estimates in order to enable operational decision making on crowds during a major event. Detection of abnormal behavior of lone pedestrians in the crowds also falls in this category, and it has huge security implications.

3. **Prediction**. Use of real data estimates in order to forecast the future behavior of the crowd. The detection of risk factors in crowds which indicates a possible upcoming accident falls in this category, since predictive models have to be constructed in order to anticipate human reactions.

For estimating the number of people attending an event, one may use imprecise information from previous related events, or additional sources such as the local Global System of Mobile Communications (GSM) usage. However, such information is so unreliable that the event planning is usually performed by largely overestimating the number of attendants, thus tending to be overcautious in the urban design, thus not fully exploiting the available capacity.

Due to the unsuitability of simulations for dense crowds, and to the unreliability of alternative sources of information, video analysis through computer vision can provide a powerful support to the solution of the aforementioned challenges. There is a compelling need of real crowd data, and all the process from data acquisition to pedestrian trajectory extraction can be supported by computer vision. The extraction of human trajectories has been done usually by manual intervention, or automatically in controlled settings. Recent advancements in computer vision have allowed to build always faster and more reliable systems for automatic video analysis. However, for dense crowds, the problem remains still difficult to solve. Designing a system which scales at high crowd densities would undoubtedly represent a valuable contribution to all the fields which actively study the problem.

Several novel difficulties arise when applying computer vision algorithms to high-density crowd analysis. The heads (and, occasionally, the shoulders) are the only visible body parts. Thus, a pedestrian detector cannot rely on any clue other than the head presence in order to distinguish the target. A related problem to this is that strong occlusions are frequent and persistent. Not only body parts are occluded, but also the head of a pedestrian may not be visible from a single point of observation. Moreover, the background is not static. The background is the crowd itself, thus making difficult to separate the foreground. Moreover, with such background, clutter can be a problem as worse as occlusion for harming the ability to detect some targets. As highlighted in [92], even if significant advancements have been done in computer vision for the analysis of non-crowded scenes, new methods have to be proposed in order to cope with high density, since such works rely on priors that are violated in dense crowds.

Thesis objective

The main objective of this thesis is to propose a complete framework for pedestrian detection and tracking in high-density crowds, from calibration to pedestrian track estimation. The work aims to exploit multiple camera fusion in order to handle robustly the novel problems introduced by high crowd densities.

The aim of our work is to overcome the limitations of state-of-the-art approaches which fail in different ways when dealing with dense crowds. First, we realize that single camera pedestrian detection is insufficient at high-densities, since only the visible heads are detectable, and thus can

lead to an important underestimation of the real crowd density. For such reason we believe that multiple view algorithms are essential for an exhaustive analysis. Second, a large body of research has been conducted on multiple view pedestrian tracking, but higher densities introduce novel difficulties which make the current methods fail due to the high association ambiguity.

By exploiting the potential of smart camera networks it is possible to handle implicitly the problem of frequent occlusion, since with multiple cameras the probability that one person is detected by at least a subset of the network increases. At the same time, an object could be visible from one camera but background clutter could be too heavy, while multiple cameras which are placed far and tilted enough with respect to each other can see the same head from totally different perspectives, and some of them could perform easier the detection task. Such complementary contribution of the cameras comes with a cost, which is represented by setting all the system in correspondence on a shared reference world. The more the cameras can have a different viewpoint of the scene (which is potentially beneficial for the detection task), the more the perspective change makes the world registration difficult. Thus, the interest of the work extends from registration to detection and tracking.

The main parts of the thesis can be summarized as follows:

1. **Calibration (Part I)**. We study the fundamental problem of calibrating two cameras. We highlight that in urban environments the placement of the cameras, the scarcity and ambiguity of information, can make the state-of-the-art approaches of relative camera pose estimation unusable. For such reason we exploit an entire video stream of the dynamics of the urban scene prior to the event (in low density conditions), for a robust iterative method which exhibits fast convergence to a globally good solution. We then highlight how local low errors of the estimated pose are critical for the crowd analysis task, and we show how our method is suited in order to enforce the solution to be good in all the region of interest.

2. **Multiple camera detection (Part II)**. We tackle the problem of multiple camera based pedestrian detection by performing low level information fusion. We propose an unsupervised detection method which exploits the visual consistency of the pixels in multiple views in order to estimate the pedestrian occupation. Such work can complement any supervised learning based detector because it can solve problems of difficult detection induced by cluttering, while being robust to head occlusion in some of the cameras. In order to perform such work, it is necessary to perform a metric registration of the entire scene (relate image distances to metric distances), thus a fully automated method, which uses as an input the relative pose of the cameras, is proposed. The output of such step is the joint estimation of the pedestrian occupancy in the image space and in the 3D world (by inferring the height with respect to the ground plane of each pixel), thus making possible to track pedestrians on a common ground plane, where distances between heads reflect the real ground plane separation in the crowd.

3. **Data fusion on the ground plane (Part III)**. We study the generic problem of performing data fusion on a 2D space under the Belief Function framework. We identify major limitations when handling discernment frames with large complexity, which make the state-of-the-art representations of 2D compound hypothesis not suitable for large scale data fusion. We propose an efficient compound hypothesis representation which is scale invariant and hashable. Such representation can be used for defining efficient basic operators and combination rules. We also propose a complementary representation which encodes the high level relationship between compound hypothesis (intersection and inclusion relations), in order to perform efficient decision making and to provide a compact baseline for the proposition of efficient fundamental operations of the theory. Such work provides an extensive framework for spatial and temporal fusion of detectors in the ground plane. We demonstrate our information fusion approach for temporal fusion by performing evidential tracking of pedestrians, while demonstrating that our fusion framework scales for large scenes both in sparse and dense crowds.

4. **Supervised pedestrian detection (Part IV)**. We study pedestrian detection in single views as a semantic segmentation problem. We highlight the need for an appearance based single-view detector as the foundation for a multiple view system, where supervised inference can be performed in each view, providing an essential source of information for further high level fusion. We underline the difficulties for single view detection in such scenario, such as lack of high-quality, high diversity annotated data, imprecise and not exhaustive labeling of the heads, lack of texture on the heads. We adapt state-of-the-art deep architectures in order to solve the semantic segmentation problem under such conditions, and we demonstrate the performance of the method in extreme conditions.

   We finally combine the information provided by the multiple camera pedestrian detection and the supervised estimators by using our information fusion framework, thus performing ground plane data fusion between all such sources.

Contents

List of Figures

List of Tables

Glossary

AOI - *Area Of Interest*

BA - *Bundle Adjustment*

BBA - *Basic Belief Assignment*

BFT - *Belief Function Theory*

CNN - *Convolutional Neural Network*

CPU - *Central Processing Unit*

CRF - *Conditional Random Field*

CUDA - *Compute Unified Device Architecture*

DAG - *Directed Acyclic Graph*

EM - *Expectation Maximization*

FMT - *Fast Möbius Transform*

FP - *False Positive*

FOV - *Field Of View*

GNSS - *Global Navigation Satellite System*

GPS - *Global Positioning System*

GPU - *Graphics Processing Unit*

GSM - *Global System for Mobile Communications*

GSSF - *Generalized Simple Support Function*

HOG - *Histogram of Oriented Gradients*

IA - *Interval Analysis*

IMU - *Inertial Measurement Unit*

IoU - *Intersection over Union*

ISSF - *Inverse Simple Support Function*

KDE - *Kernel Density Estimation*

LBP - *Loopy Belief Propagation*

LFE - *Local Feature Extraction*

LS - *Least Squares*

mAP - *mean Average Precision*

MLESAC - *Maximum Likelihood Estimation SAmple Consensus*

MRF - *Markov Random Field*

ORSA - *Optimized Random Sampling Algorithm*

PROSAC - *PROgressive SAmple Consensus*

RANSAC - *RANdom SAmple Consensus*

ReLU - *Rectified Linear Unit*

RMSE - *Root Mean Square Error*

RNN - *Recurrent Neural Network*

SIFT - *Scale Invariant Feature Transform*

SIMD - *Single Instruction Multiple Data*

SfM - *Structure from Motion*

SLAM - *Simultaneous Localization and Mapping*

SM - *Streaming Multiprocessor*

SSF - *Simple Support Function*

SURF - *Speeded Up Robust Feature*

SVD - *Singular Value Decomposition*

SVM - *Support Vector Machine*

TP - *True Positive*

Part I

Wide baseline pose estimation

Chapter 1

Camera pose estimation: an overview

Contents

The relative pose estimation between cameras is a crucial preliminary step of any algorithm exploiting a smart camera network, where neighboring cameras have an overlapping field of view. In this chapter we introduce the geometry of the problem as well as the classical steps for camera pose estimation.

1.1 Pinhole camera model

The pinhole camera model gives the basic foundation for projective geometry, by mapping the real world to the space of the image. Let the camera center **C** be the center of the world reference system. The axis Z is called *principal axis*, and the plane $Z = f$ is the *image plane*. The f parameter is called *focal distance*. The principal axis intersects the image plane in the principal point **p**.

The transformation between world and image plane coordinates quickly follows from the use of similar triangles [62]:

$$\mathbf{X} = (X, Y, Z)^T \longmapsto \mathbf{x} = (fX/Z, fY/Z)^T,$$

where the third coordinate of **x** is ignored because it always corresponds to f.

The transformation above assumes that the principal point $p = (p_x, p_y)$ is the origin of the image plane reference system. Since such assumption does not hold in general, the mapping is always expressed by using the principal point as a constant offset [62]:

$$\mathbf{X} = (X, Y, Z)^T \longmapsto \mathbf{x} = (fX/Z + p_x, fY/Z + p_x)^T.$$

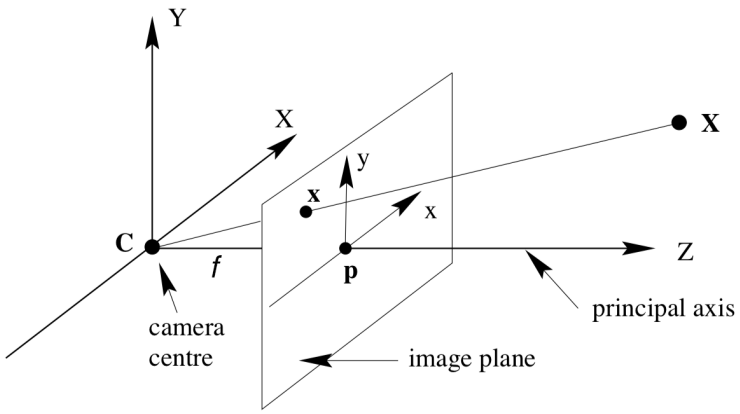


Figure 1.1: Pinhole camera model. The principal axis of the camera passes through the camera center **C** and through the image plane at the principal point **p**. The distance between **C** and **p** along the principal axis is the *focal distance f*. Image taken from [62].

The above transformation translates in homogeneous coordinates to the *camera calibration matrix* (intrinsic calibration) K:

$$\mathrm{K} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix},$$

which allows to express the transformation as the product:

$$(x, y, 1)^{\mathrm{T}} \sim \mathrm{K}[\mathrm{I} \mid \mathbf{0}](\mathrm{X}, \mathrm{Y}, \mathrm{Z}, 1)^{\mathrm{T}},$$

where I is the 3×3 identity matrix, and $\mathbf{0}$ is a 3×1 vector of zeroes.

In practice, the definition of K that we have presented introduces some simplifications which fail for some real cameras. It assumes that the image coordinates have the same scale in both axis directions. This is not true in general, since it is possible to have non-squared pixels. Thus, a scale factor has to be introduced for each dimension, which is applied to the focal distance f. Scaling the f in both dimensions results into the parameters f_x and f_y, which are the focal length in pixels along x and y respectively. The value f_y / f_x is the *aspect ratio* of the image. An additional parameter, the *skew s*, is usually added to the calibration matrix. Even if such term is usually zero for normal matrices, it accounts for particular situations where the camera axis are skewed in such a way that they are not perpendicular anymore. The general calibration matrix K is then expressed as:

$$\mathrm{K} = \begin{bmatrix} f_x & s & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix}.$$

1.2 Epipolar geometry

The epipolar geometry [62] is a projective geometry between two views, depending only on the camera intrinsic parameters and camera pose. Consider a point **X** in the 3D-space, and its projection in the two views, **x** and **x'**. The two projections, the 3D point, and the camera centers are coplanar. The plane built from such points is the *epipolar plane* π. The line connecting the centers of the two camera is the *baseline*, and the intersection of the baseline with the image planes defines two points called *epipoles* **e** and **e'** (see Figure 1.2).

The epipolar geometry solves the following problem: given **x** in the first view, what is the point locus in the second view where **x'** can lie in? First, we project in the 3D-space a ray passing through the center of the camera and **x**. The ambiguity in the depth perception provides an infinite number of 3D points placed along the ray as possible correspondences with the projected point **x**. The

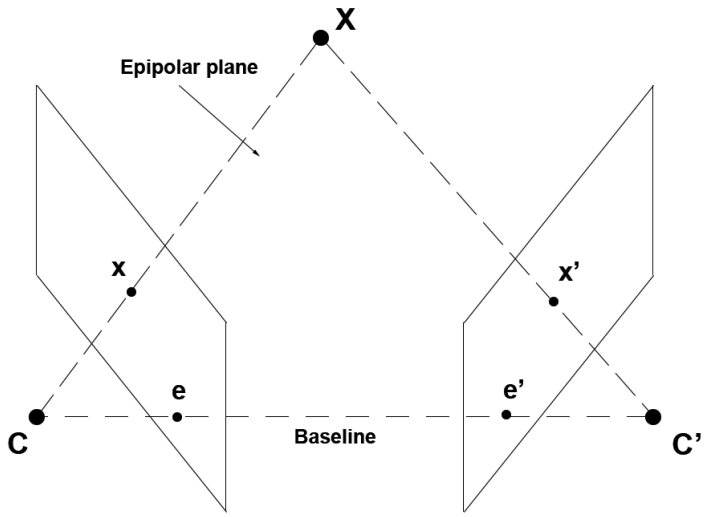


Figure 1.2: Epipolar plane built on points **C**,**C'** and **X**.

point **x'** can lie in the back-projection of the ray starting from **x** in the second image plane. It corresponds to a line **l'**, called *epipolar line*, which can be also seen as the intersection of the epipolar plane with the second image plane. All the possible epipolar lines intersect at the epipole. So the epipolar constraint leads to a point-to-line transfer relationship (see Figure 1.3).

The algebraic representation of the epipolar geometry is the **fundamental matrix F** [62]. The expression of the fundamental matrix can be derived geometrically, starting from the point-to-line correspondence. The mapping from an image point to an epipolar line can be subdivided into two steps. First, **x** is mapped to some point **x'** in the other image. Second, since the candidate **x'** will lie for sure along an epipolar line, **l'** is built as the straight line passing through **x'** and the epipole **e'**. Let us consider a generic plane π_H different from any possible epipolar plane (it does not pass through the camera centers). The point **x** in the image plane of the first camera will project a ray in the 3D space which will intersect the plane at the point **X**. Then we re-project **X** in the second image plane, obtaining the point **x'**. We have just defined a homography H_π with reference plane π_H. H_π represents the 2D homography matrix which maps the first image to the second via any plane π.
Having **x'** and **e'**, the epipolar line can be written as [62]:

$$l' = e' \times x' = \left[e'\right]_\times x' = \left[e'\right]_\times H_\pi x = Fx$$

The matrix:

$$F = \left[e'\right]_\times H_\pi$$

is the fundamental matrix between the two views. The expression $\left[e'\right]_\times$ is the skew-symmetric matrix generated by e'. F has rank 2, because $\left[e'\right]_\times$ has rank 2 and H_π has rank 3.

The expression of F can be obtained also analytically from the camera matrices [62]:

$$F = K'^{-T} [t]_\times R K^{-1}$$

The term:

$$E = [t]_\times R$$

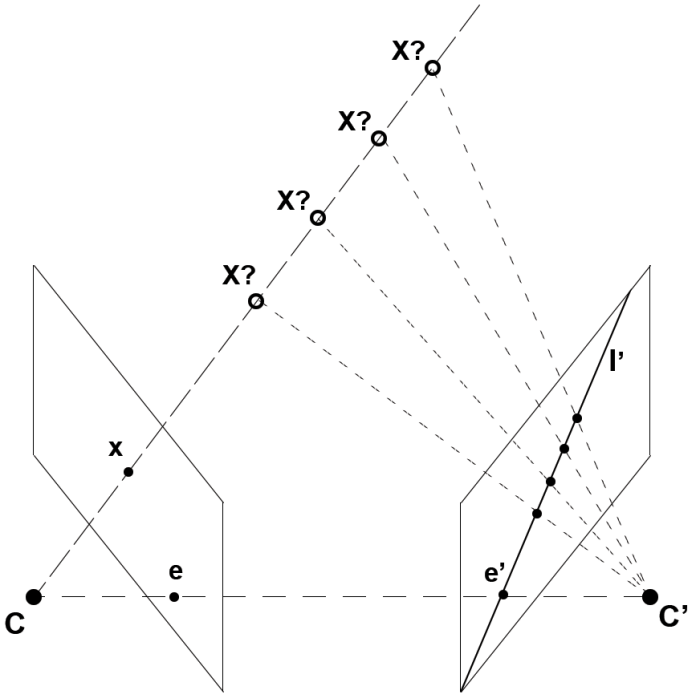


Figure 1.3: Geometric derivation of epipolar line **l'** induced but the point **x**.

is called **essential matrix**. The R and t terms are the relative rotation matrix and translation vector of the second view with respect to the reference system of the first one. This formula provides a mapping between the fundamental matrix representation and the relative camera pose parameters (up to a scale). For a given essential matrix E, multiple solutions for R and t (four in general) are admitted. However, only one of those corresponds to a valid configuration (one in which each 3D point is in front of both cameras, thus a single observation is sufficient to identify it).

The final expression of the geometric constraint imposed by epipolar geometry is the following [62]:

$$x'^{\mathrm{T}} \mathrm{F} x = 0, \quad \forall x \leftrightarrow x' \tag{1.1}$$

The expression derives from the fact that both x' belongs to the epipolar line $l' = \mathrm{F}x$ and x belongs to the epipolar line $l = \mathrm{F}^{\mathrm{T}} x'$.

1.3 Robust estimation of the epipolar geometry

1.3.1 Robust estimation of fundamental matrix

Equation (1.1) implies that F can be estimated from image correspondences. The *normalized 8-point algorithm* [61] can be used to provide a unique solution from at least 8-point pairs. The term *normalized* derives from the fact that a normalization step is performed on the points before looking for a solution (e.g. by performing isotropic scaling of points), for numerical stability.

The epipolar constraint at Equation (1.1) can be transformed into a linear system of equations:

$$\begin{bmatrix} x_1x_2 & y_1x_2 & x_2 & x_1y_2 & y_1y_2 & y_2 & x_1 & y_1 & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \end{bmatrix} = 0$$

The problem is now an Homogeneous Least Square problem, which can be solved by the use of Singular Value Decomposition (SVD). The fundamental matrix F_r, solution of the homogeneous system, will not respect the property of having rank-2. In order to well-condition the matrix, the final matrix F will be the rank-2 one minimizing the the Frobenius norm $\| \mathrm{F} - \mathrm{F}_r \|$ [61]. Given:

$$\mathrm{F}_r = \mathrm{USV}^{\mathrm{T}} = \mathrm{U}diag(\sigma_1, \sigma_2, \sigma_3)\mathrm{V}^{\mathrm{T}}, \quad \sigma_1 \geq \sigma_2 \geq \sigma_3$$

We can obtain the closest rank-2 F as:

$$\mathrm{F} = \mathrm{U}\tilde{\mathrm{S}}\mathrm{V}^{\mathrm{T}} = \mathrm{U}diag(\sigma_1, \sigma_2, 0)\mathrm{V}^{\mathrm{T}}$$

An alternative *7-point algorithm* exists, satisfying directly the rank 2 constraints, and providing from 1 to 3 solutions as output.

In real word scenarios, the correspondences used for the derivation of F are noisy, since they are extracted by using some statistical feature descriptor (such as SIFT [100], SURF [8], or more recent CNN based descriptors [182]). Moreover, due to the presence of outlier pairs, the set of correspondences should be fed as input of a robust estimation schema. A robust estimator outputs not only the solution, but also the point matches which are *inliers* for the estimated solution (the ones which have consensus on the estimated geometry).

1.3.2 RANSAC

The *RANdom SAmple Consensus* algorithm [41] is a robust iterative algorithm for general parameter estimation from a set of observations, able to cope with a significant number of outliers. In the case of geometric constraint estimation the observations are represented by pairs of matched features.

The robust estimator needs the definition of a distance metric between the observations (the matches) and the model (the matrix parameters). In the case of epipolar geometry estimation, the distance function is defined as follows. Given a point match (p_1, p_2) which should satisfy $p_2^{\mathrm{T}}\mathrm{F}p_1 = 0$, we use the symmetric geometric distance between a point and the correspondent epipolar line:

$$d = \frac{1}{2}\left(\mid p_1 \cdot l_1 \mid + \mid p_2 \cdot l_2 \mid \right)$$

The main steps of the procedure are described in Algorithm 1.

Algorithm 1: RANSAC algorithm

Data: Set of observed data.
Result: Estimated parameters of the model; set of inliers.

1. Randomly select a subset containing the minimum number of points required to determine the parameters of the model.
2. Test for degeneracy of selected sample: if the sample is degenerate, go to step 1.
3. Compute model parameters starting from the subset's points.
4. Classify all the points in the original set as either inliers or outliers.
   A point is an inlier if its distance d to the estimated model is lower than a threshold T_ϵ.
5. If the number of inliers is the maximum obtained so far:
   - store the estimated parameters as the actual best solution;
   - re-estimate the expected number N of iterations to perform from this point.
6. If the iteration number is less than N, go to step 1.
7. Recompute the parameters of the model based on all the inliers.

The main disadvantage of the algorithm is that it contains a critical parameter, the threshold T_ϵ, which is specific for the model, and usually is set on the basis on experimental evaluations. The number of iterations N is automatically computed, and must be high enough to ensure a probability p of choosing a random sample containing only inliers (e.g. $p = 0.99$). Let w represent the probability that a data point is an inlier. It follows that, given n the size of an extracted subset, w^n is the probability that all the points of the sample are inliers, and $1 - w^n$ is the probability that at least one point is an outlier. It follows the relationship [41]:

$$1 - p = \left(1 - w^n\right)^{\mathrm{N}}$$

The expected number of samples N is then estimated as [41]:

$$\mathrm{N} = \lceil \frac{log(1-p)}{log(1-w^n)} \rceil$$

The value of N must be re-estimated each time that a new possible best-fit solution is found because the expected probability of a data point of being an inlier w is taken exactly as the actual inliers percentage for the current solution.

Beside the standard RANSAC version, various adaptations exist (e.g. MLESAC [158], PROSAC [20], etc.), but the underlying idea of exploring the inlier consensus is always present.

1.3.3 ORSA

In [112] the authors propose an *a-contrario* model which defines a rigidity detection criterion to be used together with an Optimized Random Sampling Algorithm (ORSA), in order to outperform other methods (e.g. RANSAC) in term of robustness. Moreover, it doesn't need for an explicit threshold definition, while being robust to the presence of high percentage of outliers (on the contrary, another robust method, Least Median of Squares, does not require a threshold but works only for inlier percentages above 50%).

The symmetric F−rigidity of a set of n correspondences $\mathrm{S} = \{p_{i,1}, p_{i,2}\}_{i=1\ldots n}$, can be defined as [112]:

$$\alpha_{\mathrm{F}}(\mathrm{S}) = \max_{(p_{i,1},p_{i,2})\in \mathrm{S}} \max\left(\frac{2\mathrm{D}_2}{\mathrm{A}_2} \mid p_{i,1} \cdot l_{i,1} \mid, \frac{2\mathrm{D}_1}{\mathrm{A}_1} \mid p_{i,2} \cdot l_{i,2} \mid\right)$$

where A_1,D_1,A_2 and D_2 are the area and diameter of first and second image, respectively. The set S is said α-rigid if there exists a fundamental matrix such that α is the highest bound of rigidity.

The *a-contrario* definition of ϵ-meaningfulness is then used in order to reduce the parameter space (size of the set S, rigidity threshold) to a unique parameter, which is the expected number of false alarms. In the given scenario the meaningfulness of a set is given by the expected number of sets with same size and at least same rigidity, given a uniform distribution of points. The authors of [112] quantitatively define meaningfulness in presence of outliers. A set $S' \subseteq S$ of k among n matches is ϵ-meaningful as soon as it is α-rigid with [112]:

$$\epsilon(\alpha, n, k) \coloneqq 3\,(n-7)\binom{n}{k}\binom{k}{7}\alpha^{k-7} \leq \epsilon$$

Given the previous definitions, the ORSA algorithm is detailed in Algorithm 2.

Algorithm 2: ORSA algorithm

Data: Set of observed data S; N

Result: Estimated parameters of the model; set of inliers.

1. Perform random sampling of the correspondences set S for N iterations, until an absolutely meaningful rigid set U is found ($\epsilon(U) < 1$) .
2. Set $\widetilde{\epsilon} = \epsilon(U)$.
3. Set $N_{opt} = \frac{N}{10}$.
4. Find a random set of 7 points $T \subseteq U$.
5. For each fundamental matrix F computed from T:
   - Find the most meaningful rigid set $\widetilde{S} = \widetilde{S}(F)$ associated to F;
   - If $\epsilon(\widetilde{S}) < \widetilde{\epsilon}$ set $\widetilde{\epsilon} = \epsilon(\widetilde{S})$ and $U = \widetilde{S}$.
6. If the number of trials is less than N_{opt}, go to step 4.

1.4 Non linear refinement of the solution

Once a robust solution is computed, a non-linear optimization step can be performed on the set of inliers. The objective is to obtain a solution which is a global minimizer of the sum of square inliers residuals.
The problem is in the form:

$$\min_x \| g(x) \|^2$$

where the function $g(x)$, in our case, defines a geometric error distance, the x vector defines the parameters to be tuned, which depend on the F entries in our case.

1.4.1 Re-parametrization of the solution

The simplest approach that we could consider is to use every element of the fundamental matrix as parameter. So, if we define:

$$F = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

the parameters' vector will be:

$$x = \begin{bmatrix} a & b & c & d & e & f & g & h & i \end{bmatrix}$$

However this approach is not suitable, because we are not exploiting the fact that the fundamental matrix is a rank-2 matrix. What would happen using this parametrization is that we start from a rank-2 matrix, but we give to the algorithm the possibility to explore full rank solutions, and with high probability we will have in output a matrix which is not of rank 2. We want to limit the degree of freedom of the nonlinear algorithm because transforming the matrix in a rank-2 one at the end (as we do in the first part) causes a loss of precision that could be even more consistent that the gain achieved with the refinement.

Enforcing rank-2 constraint in the parametrization causes the use of a parameter vector of length 8. One row is the linear combination of the other two. For example, if the dependent row is the first one, the following conditions hold [185]:

$$\exists \lambda_1, \lambda_2 \quad s.t. \quad \vec{r}_1 + \lambda_1 \vec{r}_2 + \lambda_2 \vec{r}_3 = 0 \tag{1.2}$$

$$\nexists \lambda \quad s.t. \vec{r}_2 + \lambda \vec{r}_3 = 0 \tag{1.3}$$

where $\vec{r}_i$ is the i^{th} row. The condition in Equation (1.3) cannot be expressed inside the parametrization, so, using only Equation (1.2), we parametrize the matrix as a matrix of rank strictly less than 3. If the dependent row is the first one, F can be written as:

$$\mathrm{F} = \begin{bmatrix} -\lambda_1 a - \lambda_2 d & -\lambda_1 b - \lambda_2 e & -\lambda_1 c - \lambda_2 f \\ a & b & c \\ d & e & f \end{bmatrix}$$

$$p_8 = \begin{bmatrix} a & b & c & d & e & f & \lambda_1 & \lambda_2 \end{bmatrix}$$

p_8 is the parameters' vector, which now has 8 entries.
Consider the submatrix Q_i obtained by removing the i^{th} row from F:

$$\mathrm{Q}_i = \begin{bmatrix} a_i & b_i & c_i \\ d_i & e_i & f_i \end{bmatrix}$$

The row index to remove will be the one which maximizes:

$$argmax_i \left(\mathrm{Q}_i^{\mathrm{T}} \mathrm{Q}_i \right), \quad i = 1...3$$

This step is essential since if we choose e.g. row 1, we may be in a degenerate configuration for which the parametrization is invalid:

$$\mathrm{F} = \begin{bmatrix} \vec{r}_1 \\ \vec{r}_2 \\ \alpha \vec{r}_2 \end{bmatrix} \quad \mathrm{F} = \begin{bmatrix} \vec{r}_1 \\ \vec{0}_{1\times 3} \\ \vec{r}_3 \end{bmatrix} \quad \mathrm{F} = \begin{bmatrix} \vec{r}_1 \\ \vec{r}_2 \\ \vec{0}_{1\times 3} \end{bmatrix}$$

As suggested in [185] we can perform an even smarter parametrization: we enforce the constraint both for rows and for columns. We have to choose a j_0 independent column and a i_0 independent row. We have a total of 9 different parametrizations corresponding to different choices of i_0 and j_0. For example, the matrix F in the case $i_0 = j_0 = 1$ will be expressed as:

$$\mathrm{F} = \begin{bmatrix} \overline{\lambda}_1 (\lambda_1 a + \lambda_2 b) + \overline{\lambda}_2 (\lambda_1 c + \lambda_2 d) & -\overline{\lambda}_1 a - \overline{\lambda}_2 c & -\overline{\lambda}_1 b - \overline{\lambda}_2 d \\ -\lambda_1 a - \lambda_2 b & a & b \\ -\lambda_1 c - \lambda_2 d & c & d \end{bmatrix}$$

$$p_8 = \begin{bmatrix} a & b & c & d & \lambda_1 & \lambda_2 & \overline{\lambda}_1 & \overline{\lambda}_2 \end{bmatrix}$$

Note that, in the previous equation we keep the same notation as in [185], and that all the parameters are independent. The parametrization's length is again 8, but this time we gain a big

advantage: we can extract both the epipoles directly from the parametrization [185]. The new criteria for the best map choice is the maximization of the rank of the 9×8 Jacobian matrix defined as [185]:

$$\mathrm{J} = \frac{\partial f_{i_0,j_0}}{\partial p_8}$$

The function f is simply the parametrization of F reshaped in a 1×9 vector. Such task is equivalent to the maximization of the norm of the 1×9 vector v whose elements are the determinants of the 8×8 submatrices of J. Since different maps lead just to a different order of the entries of the vector v, we can obtain an expression of the norm of v which is independent from the values of i_0 and j_0:

$$\| v \| = (ad - bc)^2 \sqrt{\left(1 + \lambda_1^2 + \lambda_2^2\right)\left(1 + \overline{\lambda}_1^2 + \overline{\lambda}_2^2\right)}$$

So the best map can be chosen without explicitly calculating the Jacobian.

The length of the parameter vector can be further reduced by one, since the fundamental matrix is defined up to a scale. We can use a simple criterion to remove one of the elements a, b, c, d, e.g. remove the biggest number in absolute value [185]. We also divide the three remaining values by the deleted one. If, for example, we decide to remove the d entry, the final 7-entries parameter vector will be:

$$p_7 = \begin{bmatrix} \frac{a}{d} & \frac{b}{d} & \frac{c}{d} & \lambda_1 & \lambda_2 & \overline{\lambda}_1 & \overline{\lambda}_2 \end{bmatrix}$$

1.4.2 Cost function

We need to define a cost function for the non-linear optimization algorithm. Using:

$$g_i = p_{i,2}^{\mathrm{T}} \mathrm{F} p_{i,1}$$

would lead to a bad estimation, because the cost function is scale dependent, and the variance of each g_i is not the same. In general we want to minimize the geometric distance between the point pair under analysis and the closest correspondence which satisfy exactly the epipolar constraint. As shown in [62], trying to minimize exactly this distance needs an estimation of the reprojection of a 3D point in the image space, and involves estimating not F directly, but first the camera projection matrices which give a best fit, and then obtaining F from them. The problem in this form has $7 + 3n$ degrees of freedom, where n is the number of 3D points.
A number of cost functions which approximate the reprojection distance are proposed in literature. The **Sampson distance** for the epipolar constraint is defined as [62]:

$$g_i = \frac{p_{i,2}^{\mathrm{T}} \mathrm{F} p_{i,1}}{\sqrt{\left(\mathrm{F} p_{i,1}\right)_1^2 + \left(\mathrm{F} p_{i,1}\right)_2^2 + \left(\mathrm{F}^{\mathrm{T}} p_{i,2}\right)_1^2 + \left(\mathrm{F}^{\mathrm{T}} p_{i,2}\right)_2^2}}$$

It gives a first order approximation of the geometric distance. Please note that it is undefined for a point pair containing the two epipoles, since the denominator would be zero.

1.4.3 *Levenberg-Marquardt* algorithm

The *Levenberg-Marquardt* algorithm is an iterative method for solving the non-linear least squares problem. The algorithm takes as an input a starting point parameter estimate, which should be theoretically close to the global minimum, in case of non-convex optimization.

The parameter update policy of the algorithm can be thought as a combination of Gradient Descent and Gauss-Newton methods. Let us consider the function to minimize:

$$\mathrm{G}(x) = \frac{1}{2} \sum_{i=1}^{m} g_i(x)^2$$

where m is the number of observations (inlier matches in our problem), g_i is the cost function (e.g. Sampson distance), and x is the parameter estimate.

Given the current estimate β, if we denote as $J_i(\beta)$ the Jacobian of $g_i(\beta)$, the direction of parameter variation δ is given by:

$$\left(\mathbf{J}^{\mathrm{T}}\mathbf{J} + \lambda\,\mathrm{diag}(\mathbf{J}^{\mathrm{T}}\mathbf{J})\right)\delta = -\mathbf{J}^{\mathrm{T}}\mathbf{g}(\beta)$$

where λ is called *damping parameter*.

1.4.4 Uncertainty of the estimation

The uncertainty of the estimation is represented by a covariance matrix in the nine entries of the fundamental matrix. We call $\mathbf{J}_f$ the Jacobian of the error at the last iteration of the non-linear algorithm. The associated covariance matrix will be:

$$\Lambda_{p_7} = \frac{\mathrm{S}}{n-p}\left(\mathbf{J}_f^{\mathrm{T}}\mathbf{J}_f\right)^{-1}$$

where S is the final norm of the residuals, n is the number of points matches used, p is the size of the parametrization ($p = 7$ in our case). Λ_{p_7} is the 7×7 covariance matrix of the 7 parameters of the fundamental matrix. We are now interested into obtaining the 9×9 covariance matrix for the 9 entries of F. We can obtain it by using the following transformation [23]:

$$\Lambda_{\mathrm{F}} = \frac{\partial f_{i_0,j_0}}{\partial p_7}\Lambda_{p_7}\frac{\partial f_{i_0,j_0}}{\partial p_7}^{\mathrm{T}}$$

Here we need the Jacobian of the parametrization with respect to the vector p_7. Having the expression of the 9×8 Jacobian with respect to p_8:

$$\frac{\partial f_{i_0,j_0}}{\partial p_7} = \frac{\partial f_{i_0,j_0}}{\partial p_8}\frac{\partial p_8}{\partial p_7}$$

This is equivalent to computing the 9×8 Jacobian using the normalized entries e.g. $a/d, b/d, c/d$, and at the end removing the column corresponding to the index of the entry which has been removed when passing from p_8 to p_7.

1.5 Conclusion

In this chapter we have introduced the reference camera model, as well as the geometric relationship between pairs of camera models, in terms of the fundamental matrix. Moreover, we have outlined the standard approach for robust estimation and non linear optimization of such constraint. This preliminary chapter lays the foundation of the estimations which will be used further for inferring pose on difficult scenes through video streams. As it will be further discussed, the estimation robustness of any of the different estimators outlined depends heavily on the quantity and quality of the available correspondence set. In the following chapter we will detail the problematic which may arise in surveillance scenarios, which can be a consequence of a constrained camera placement in large scale scenes. Moreover, a new methodology to overcome such problems will be proposed, allowing to perform robust camera calibration in the wild without manual intervention.

Chapter 2

Robust wide baseline estimation from video

Contents

2.1 Introduction

The calibration of a camera network with minimal requirements of human intervention (use of calibration objects, guidance of the pose estimation process) has long represented a major field of research in computer vision and photogrammetry, with novel contributions and surveys appearing regularly [7, 13, 14, 32, 58, 126, 130, 135]. Recently, the increased focus on safety and surveillance applications has underlined the importance of smart camera networks (the reader may refer to [107, 138] for a more detailed taxonomy of the major challenges raised by smart cameras). The self calibration part is critical for monitoring projects, for multiple reasons. In order to be able to

(a) (b) (c)

Figure 2.1: Sample frames acquired from the three cameras. (a) Camera 1, (b) Camera 2, (c) Camera 3. Two large featureless regions can be seen on the bottom-right and top-left of the square

project image elements from one camera to another in the case of cameras with overlapping fields of view, a relative pose estimation is mandatory and may either help locate an existing element of interest in a different view, or if the calibration is accurate enough, it may help identify elements of interest from raw data (i.e. disambiguate using the second view a person who is strongly occluded in the initial view).

Irrespective of the number of cameras deployed, the pose estimation between a pair of cameras is the foundation of any camera network calibration. Existing relative pose estimation algorithms are, for the vast majority, as seen in Chapter I.1, based on matching interest points among the two views and then on applying a robust optimization algorithm in order to determine the unknown pose parameterization [62, 106, 112, 147]. Besides being used in surveillance, these approaches stem from and benefit to various domains ranging from aerial imaging to Structure from Motion (SfM) for virtual reality. However, for large scale camera networks in urban environments, some specific scene characteristics complicate or dismiss altogether the use of existing approaches. As introduced in the end of Chapter I.1, due to physical positioning constraints, wide baselines with significant perspective change may be imposed. Even when ignoring positioning constraints, it is beneficial to cope robustly with significant pose variations in order to minimize the number of cameras required for covering a specific area. Another problem is raised by the actual image content; for outdoor surveillance, the scenes are often homogeneous (open spaces) for the most part, or featuring repetitive patterns (human shapes, building facades), and this hampers the use of fully automatic calibration algorithms. Finally, calibration solutions which require significant human intervention, by using calibration objects for example, are time and resource consuming, and in certain situations they are impracticable due to the size of the scene or due to access constraints.

As an example consider the surveillance scenario of Figure 2.1. The same location (Regents Park Mosque, London) is recorded from three different views. The scale of the scene makes it impracticable to use helper objects for calibration, since it would be non trivial to cover all the image space. Moreover the area could not be fully accessible (e.g. for security reasons), so an automatic calibration is required. A common surveillance task (that will be explored in later chapters) is the analysis of a dense crowd interacting in the region of interest. There are two calibration options:

- **Online calibration**. Calibrate the cameras during the event. There is a rich amount of information, due to the presence of several people. However, calibrating on a dense crowd can sensibly harm the quality of the correspondences used for the estimation (clutter, occlusions, and especially ambiguities when matching body parts).

- **Offline calibration**. Calibrate the cameras before the start of the event. The scene is free from cluttery areas, but the information may be poor or inexistent in some areas of the space. Such featureless regions are more frequent in areas of interest for crowd analysis, since pedestrian spaces are usually free of static obstacles which may be used as calibration means.

Our study is focused on offline calibration solutions which try to enrich the original amount of information conveyed by the image pixels.

2.2 Related works

Since the pose estimation requires a set of correct matches, the outlier rejection is a prerequisite step which is usually performed using a RANSAC-based approach [106, 147]. A large number of matching observations with a significant ratio of inliers is a positive indicator for, but does not implicitly guarantee, a high-quality pose estimation, as the distribution of matches over the image space is also involved. Wide baseline setups in urban areas exhibit at the same time a low number of matches, a low ratio of inliers as well as a skewed distribution due to large uniform zones (ground, roofs, facades etc). As a result, an uneven distribution leads to a pose estimation which is correct only in covered areas, although the solution is consistent with the observations.

2.2.1 Guided matching

In order to address these problems, guided matching strategies aim to expand the well-constrained area by encouraging a progressive inclusion of new matches [118].

The basic idea of guided matching is the following. Given the estimated F and a set of inliers, one can calculate the covariance of the solution. Such covariance represents the uncertainty of the epipolar line drawn for a specific point. Thus, one can find new matching points by restricting the search area from the whole image to a small band around the epipolar line. Moreover, further relaxed matching scores can be used (admitting lower quality matches). Such new matches can be included in the initial inlier set, and the F may be re-estimated by non linear optimization among such larger set. Thus, guided matching aims to increase the consensus by solving ambiguities via admitting some uncertainty of the current solution.

However, in difficult scenes the potential elements to include are sparse and distant, and guided matching may easily include outliers and drive the pose estimation towards an inadequate solution. More elaborate strategies may relax the quality of matches in addition to guiding the search spatially [153], but this favors the inclusion of incorrect correspondences. Correct matches tend to form clusters with specific motions, and previous works proposed explicit geometrical checks for guaranteeing a consistent transformation of the inlier point set [52, 95, 153], based on local planarity or local contour invariance. More recently, data-driven strategies for selecting consistent observations have been proposed; for example in [173] the authors rely on a one-class SVM to select a reliable candidate inlier set, and in [96] a motion model based on bilateral functions is used. However, all these approaches which rely on higher level perceptual information in order to validate the inlier set coherent motion are not effective in complex urban environments with scarce candidates, abrupt and frequent depth variations of the scene and inconsistent edge detections due to significant viewpoint changes (see for example Figure 2.1).

An interesting correlation between the pose estimation errors and the number of matches, albeit empirically validated, has been discussed in [98]. This justifies all the more the fact that below a certain level of conveniently distributed inlier information, guided matching will not be able to recover a globally fit solution.

Observed limitations of guided matching

The standard guided matching approach has been tested on the *Regent's Park* sequence (see Figure 2.1). An incremental strategy has been followed. Typical guided matching approaches, which aim to enforce uniform distribution of matches, subdivide the reference image in buckets, and set a maximum number of matches per bucket (in order to avoid excessive clustering). An incremental approach is then used: New matches are searched inside the most covered buckets and in their neighbors. Then the new matrix is estimated from these new points, and the process continues.

Such approach allows us to look for new points around the area where the solution is more confident, in such a way to avoid sudden variations of the estimation at following steps, and to get a smooth and incremental refinement.

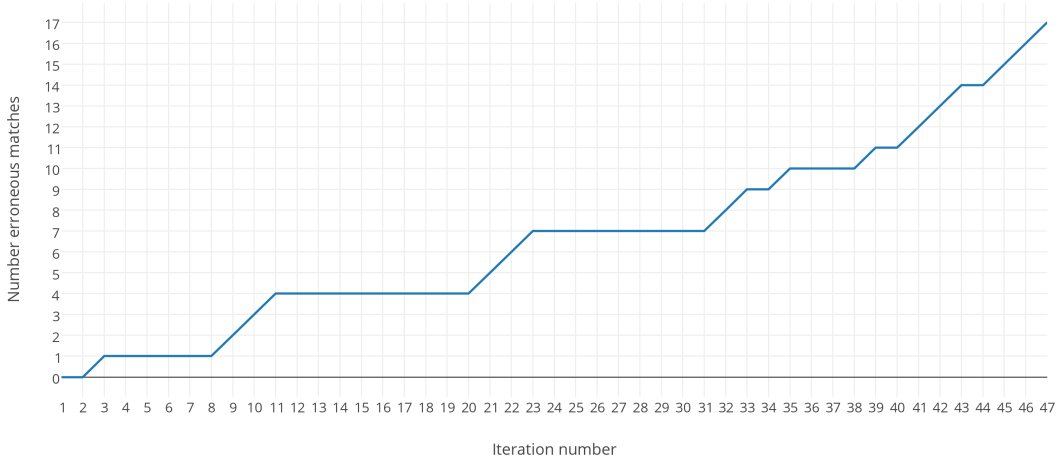


Figure 2.2: Guided matching results for the *Regent's Park* dataset. A new match is added at each iteration. The curve steps up each time a new match is erroneous with respect to the given ground truth.

Figure 2.2 provides an analysis of the outcome of guided matching. Each iteration corresponds to the addition of a new match. The number of erroneous matches increases each time that such match is wrong in terms of the ground truth epipolar geometry. The given dataset exhibit some clustering of inliers points in a specific area of the image (see Figure 2.6 in order to get an idea of the inliers distribution). The first iterations add points close to such clustered area, so the ratio of erroneous matches is relatively low. Limitations of guided matching arise when moving away from such area. Many areas of the image, which are further explored, lack of salient points, and thus low quality matches are forced to be added. In the last iterations, almost all new matches are in reality wrong. Such issue may have a large negative impact for the initial estimation, especially if the initial set of inliers has a small size.

2.2.2 Externally guided pose estimation

The impact of the challenges raised when facing wide baseline calibration may be mitigated by the use of independent sources of information. One promising avenue is the use of a prior pose hypothesis relying on GPS devices, which provides the approximate locations, coupled with IMUs, which provide the orientations. M-estimators are well adapted for guiding the pose search based on prior information [50], and for real-time applications RANSAC based strategies are also widely used i.e.[46, 82].

A second strategy which has gained popularity recently relies on the additional creation of a cartography of the surveyed environment using SLAM [6, 49, 127]. While this technique is the only way to register cameras with non-overlapping fields of view (using visual information), it can also help in wide baseline scenarios as the pose estimation is reduced to two localization tasks within the cartography.

The externally guided techniques overcome the difficulties of the purely vision based pose estimation, at a cost. For prior pose hypotheses, the cameras must be fitted with additional devices, and also the systems must be accurately calibrated offline in order to align the sensor and camera reference systems. When using a cartography, the mapping procedure may be cumbersome and is valid as long as the scene does not change significantly. In addition, any dynamic parts of the scene contribute only to the outlier observations, and also access to the scene for mapping is not always possible due to various types of restrictions. Finally, externally guided procedures cannot be appended once the dataset has already been acquired - the ideal solution would just rely on the actual video data.

2.2.3 Leveraging temporal information

The exploitation of the video stream seems a promising solution (the temporal synchronization of the cameras being convenient, but not a strict requirement). A naive approach, as pointed out by [132], is to extend image-based to video-based registration by temporal accumulation of matches. An alternative strategy identifies corresponding trajectories of salient objects [16] in order to populate the match set. Despite the richness of video information, the exploitation of video sequences does not address implicitly all the problems previously raised. Although the number of total matches does increase, in scenes with homogeneous dynamic objects such as crowded areas the inlier ratio may actually decrease. Another limitation of straightforward video accumulation is that the new matches are clustered around moving objects, and the pose estimation may get constrained locally very strongly, which in turn may remove sparse correct matches and deteriorate the solution.

Moreover, in [16], each candidate estimation is performed on a set of matches extracted from a single trajectory (or a pair of them). The authors request non-trivial trajectories to be present, which are trajectories able to cover a large enough part of the image space, and which do not belong to a degenerate configuration (planar trajectory). However, in large scale scenes a representative set of non-trivial trajectories which span most of the image space is often not available; each trajectory is likely to cover a small fraction of the total area, and to be degenerate, when the dynamics of the scene are mostly produced by people walking on the ground plane.

In [132] the authors estimate the geometric constraint by accumulating matches from a fixed number of dynamic texture image pairs. A limitation of this approach (and of the trajectory-based one), is that only dynamic parts of the scene are considered. If a scene contains large static parts (e.g. buildings, see Figure 2.1) the estimation will not be globally correct. Moreover, the method is unfeasible, in terms of memory requirements, when applied to high resolution images.

Recent efforts aimed at pose estimation from video use motion barcodes of lines [77]. The authors sample points on the image borders and connect any pair of them in order to build a set of candidate epipolar lines. Then, lines are matched by their motion barcodes, computed from background subtraction, and a RANSAC estimation is performed given the line matches. Beside the need to explore a large search space, the method may fail when people move in a straight line in the scene, due to the extraction of a quasi-degenerate pencil of candidates. Moreover, when applied to real datasets as PETS 2009 [40], the method in [77] as well as other algorithms are benchmarked against the provided ground truth calibration. However, such ground truth may itself present (as we will discuss in Section 2.8.2 for PETS 2009) local errors resulting into a performance bias of the evaluation.

2.3 Camera pose estimation in difficult scenes

2.3.1 Motivation

In the previous section we have highlighted the main limitations of guided matching methods and of temporal approaches. When dealing with difficult scenes like the ones presented, we believe that guided matching is still an efficient solution, provided that it can exploit the richness of information from not only a single image pair, but from an entire video stream. In our approach we aim to overcome compromises (e.g. admitting lower quality matches) by efficiently exploiting the temporal information.

Moreover, we stress the fact that an high quality ground truth (e.g. with at least a one-pixel precision in all the region of interest) is critical for global assessment of the proposed estimation, especially in fields where high precision calibration is critical (see Section 2.8.3). Thus, a rigorous ground truth extraction procedure needs to be stated in order to evaluate possible local errors of the solution, which may appear good in general, but may exhibit inadequate performance in some areas of the region of interest.

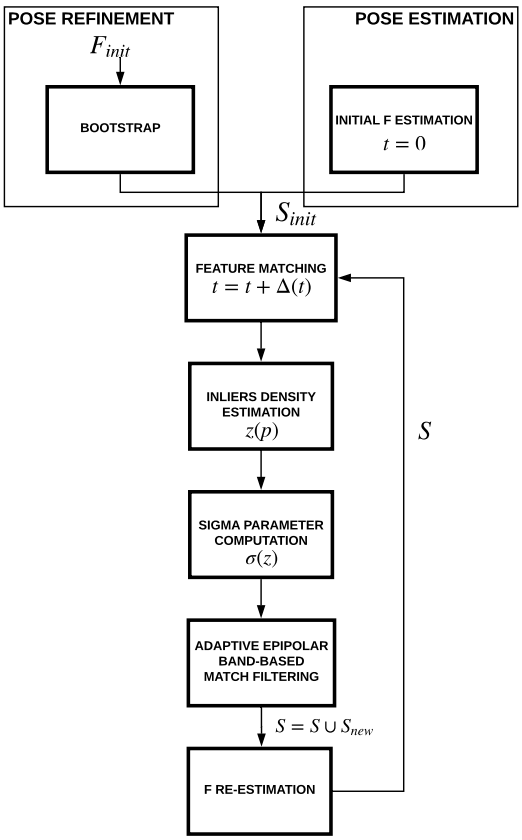


Figure 2.3: Overview of our algorithm, which may be executed either for a generic pose estimation (Section 2.4) or for the refinement of an existing prior pose (Section 2.6)

2.3.2 Overview of the proposed approach

We consider a pair of calibrated, synchronized cameras, with overlapping fields of view. Intrinsic calibration is performed on each camera independently using the Tsai's algorithm [161].

In our approach we exploit the richness of information provided by an existing video sequence, in contrast with relying on a single image pair. In fact, we have noticed that in such wide baseline scenarios with large scale regions of interest, it is common that at any given moment only some image locations provide correspondences, increasing the risk of obtaining locally optimal epipolar geometry estimations. As a result, the quality of an estimation based on feature matching may differ a lot for different time instants, rendering the image based estimation algorithms unreliable (this point is underlined by Figure 2.8 in the results Section).

On the contrary, our method starts from an image-to-image initial estimation, and refines it by acquiring new information in the following frames. At each iteration, the epipolar constraint estimated at the previous step is used to guide the acquisition of new matches between the current frames, through the use of an epipolar band. This new set of matches is combined with the set of inliers identified at the previous step, and a new robust estimation is performed on the new set.

A common practice for match selection is to extract globally distinctive matches which satisfy specific quality-related metrics (such as the 2NN heurisic proposed in [100]), as well as to enforce a symmetry check which validates pairs only with the best match candidate for both left and right feature points. Given a feature point in the first frame, and a set of candidate features in the second frame, a match with a point of the candidate set is extracted only if it is by far the most distinctive among the others. Thus, a filtering procedure is applied, by taking into account only the quality of the candidate matches.

In contrast to this approach, in our selection stage we first extract matches inside the band region, and, only afterwards, we look for candidate matches which are distinctive inside such search region. On the other hand, filtering at an early stage could remove high quality candidate matches,

and it could consequently harm the overall goodness of the output correspondences. However, in our matching strategy, we always encourage the choice of filtered matches only if their quality is as high as without the filtering step. This procedure is very effective in providing a much larger number of good quality matches, which is critical both because in a wide baseline scene globally distinctive high quality matches are scarce, and because the algorithm is capable to converge faster towards a robust solution.

Moreover, differently from a standard guided matching approach, we do not use only the uncertainty of the estimation of the fundamental matrix to compute the band size, but we adjust the band based on the inlier distribution in the image. This approach has two advantages: it guarantees a faster convergence of the solution, encouraging the matching in parts deficient in inliers, while discouraging the inclusion of conflicting matches in areas rich in information.

The illustration of all the proposed steps is supported by a ground truth that we have manually created from the testing scenes. The ground truth consists in manual matches uniformly extracted across all the common field of view, in order to test as fairly and comprehensively as possible the quality of the solution.

Our method, which allows us to automatically recover the relative pose between two cameras in an iterative way in the time dimension, has shown during our experiments to reach a quasi-monotonic decreasing of the geometric error with respect to the number of iterations, while strongly improving the robustness of the estimation, even with different choices of the robust estimator employed.

The main functionalities of our algorithm are presented in Figure 2.3 and will be detailed in the following sections.

2.4 Integrating temporal information from synchronized video streams

2.4.1 Temporal sampling

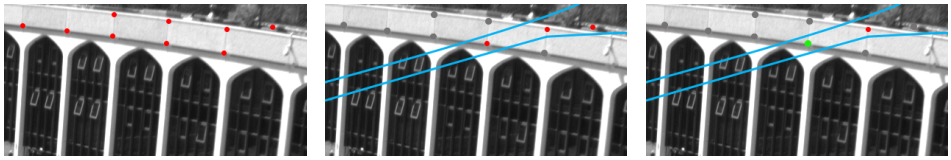
An important parameter of our process is the stream sampling period $\triangle_t$. Since we want to exploit the dynamic behavior of the objects in the scene, $\triangle_t$ should be large enough in order to allow for a significant displacement of the dynamic objects, and to avoid new information being mostly redundant. This constraint is in opposition with a tracking-based approach which needs small inter-frame difference in order to work efficiently. On the other hand, setting a too high $\triangle_t$ would just cause a slower convergence in time.

2.4.2 Matching strategy

Given the two frames at the current instant, the objective is to extract a new set of matches S_{new} that will add new information to the current set of inliers S, which represents the output of the previous iteration. The SIFT descriptor [100] is employed in the feature extraction and matching stages. We extract an initial set of candidate matches M_{init}. Each element of the M_{init} set consists of an array m of the best k candidate matches involving a specific point p in the first frame. The array is ordered in ascending order on the basis of the SIFT descriptor's distance score.

Let us consider that we have to face the presence of repetitive structures, such as the elements on building facades or people with body parts being very similar looking at small scales. Thus, it is common for a point in the first image to relate strongly to multiple points in the second image. Of course such matches would not pass the 2NN heuristic proposed in [100], because descriptor distances would be very similar. However, if we first restrict the search space using an epipolar band, provided by the approximate fundamental matrix F computed at the previous iteration, we could find that there is only one possible match which is coherent with the geometry. In such case, that match should be considered a valid candidate because it is distinctive within the area of interest.

For this reason we invert the order of filtering stages which is typical of guided matching approaches: instead of getting global distinctive matches and then checking them against the epipo-



(a) (b) (c)

Figure 2.4: Matching strategy. (a) Initial match candidates (in red), (b) Band filtered candidates (in red), (c) Final match after 2NN-band heuristic (in green).

lar bands, we first perform the band filtering and then we isolate the distinctive matches. Given $m = [p'_1, p'_2, ..., p'_k]$, we can compute the epipolar bands in both views for each pair (p, p'_i), as a function of the uncertainty of the estimation and of the point location. The normalized epipolar line in the second image is defined as $\hat{l} = \mathrm{F}p / \| \mathrm{F}p \|$. The epipolar band is an envelope around the epipolar line which depends on the epiline covariance [186][151]:

$$\Sigma_l = \mathrm{J_F} \Sigma_\mathrm{F} \mathrm{J_F^T} + \sigma^2 \mathrm{J}_p \mathrm{J}_p^\mathrm{T}. \tag{2.1}$$

We assume that the point p is independent from F, since it has not been used in the estimation procedure. The first term encodes the uncertainty of the nine F parameters, while the second one encodes the uncertainty of the position of point p in the image. The standard deviation σ represents the isotropic uncertainty in both image directions.

The conic which gives the mathematical representation of the epipolar band can be retrieved as [62]:

$$\mathrm{C} = \hat{l}\hat{l}^\mathrm{T} - \kappa^2 \Sigma_l, \tag{2.2}$$

where κ^2 is chosen by solving $\mathscr{F}_2^{-1}(\kappa^2) = \lambda$, with λ the confidence level parameter, commonly set to 95%, and $\mathscr{F}_2$ the cumulative χ^2_2 distribution.

If p or p'_i are not contained in one of the corresponding epipolar bands, then p'_i is removed from m. We call the new filtered vector $m_{\mathrm{B}and} = [\tilde{p}'_1, \tilde{p}'_2, ..., \tilde{p}'_{k'}]$, where $k' \leq k$. In order to retain only high quality matches, the following constraint must hold:

$$\tilde{p}'_1 = p'_1, \tag{2.3}$$

if the match with best score is not contained in the epipolar band, we discard the entire current set of candidate matches, and continue. This constraint avoids the inclusion in the final set of matches with a poor absolute score. In other words, the inversion of the filtering and heuristic stages has an impact only on the choice of the second best match for score comparison, while it encourages the same matching quality as the standard approach.

We are now able to perform the 2NN heuristic on $m_{\mathrm{B}and}$:

$$\frac{d(p, \tilde{p}'_1)}{d(p, \tilde{p}'_2)} < \tau, \tag{2.4}$$

where d is the SIFT distance measure, and τ is a threshold usually set in the range 0.6-0.8.

Together with the test in Equation (2.4), we perform also a symmetry check, as proposed by some authors, in order to improve considerably the quality of the matching process. This consists in applying the same procedure in the opposite sense, from the second to the first frame. If $\tilde{p}'_1$ is the best match for p, and p is the best match for $\tilde{p}'_1$, the symmetry check is respected. If both tests are passed, then the match $(p, \tilde{p}'_1)$ is added to the set S_{new}, which contains all the matches discovered at the current iteration. Figure 2.4 depicts an example of the proposed matching strategy.

2.4.3 Fundamental matrix re-estimation

Once the matching stage has been completed, the set S_{new} containing the new matches may be added to the inlier set S obtained from the previous estimation. All these matches can be used as input of a robust estimation algorithm, in order to obtain F for the current iteration.

Our approach is independent from the specific algorithm employed at this stage, and we will demonstrate in Section 4.4.4 its use with the ORSA [112] framework. The resulting F is then refined using the Levenberg-Marquardt algorithm, and the 9×9 parameter covariance matrix is evaluated as in [186].

2.5 Density-based uncertainty estimation: the σ parameter

We exploit the parameter σ in Equation (2.1) in order to be able to deal with large errors in the epipolar constraint. If the epipolar line is correct, the σ value represents the error in the matching process which leads to a small deviation from the epipolar line. On the other hand, when the epipolar line is shifted because of an estimation error in some part of the image, σ can represent the error due to the bad localization of the line.

The underlying idea is that in areas of the image which lack inliers, there is a high risk that the current estimation is biased with respect to the optimal one. Our approach consists into varying smoothly the value of σ as a function of the inlier density, which reflects how well constrained locally the solution was at the previous iteration. When σ is small, the first term of Equation (2.1) is predominant, and the shape of the epipolar band will likely follow a hyperbola; when σ is high, the second term of Equation (2.1) dominates the first, and the epipolar band will be likely enclosed by two straight lines. Possible outliers included in the process are taken into account by using a robust estimation technique at every iteration.

Starting from a binary model for the σ function [120], we investigate the possibility to have a smoother transition function. Since the estimation involves a single parameter with a monotonic relationship with density, in this work we focus on an ad-hoc function, even if the framework could benefit from well founded statistical estimation approaches, or even belief function works [176], [75].

2.5.1 The binary density model

In our preliminary work [120], we defined the notion of well-constrained regions by using a fundamental concept introduced in the field of data clustering with noisy data [37]. In [37], a point q is considered as a *core point* if, given two parameters ϵ and $MinPts$, $|N_\epsilon(q)| \geq MinPts$, where $N_\epsilon(q)$ is the set of points at a distance lower than ϵ from q. The following definition of a *directly density-reachable* point p, given ϵ and $MinPts$, has been exploited

1. $p \in N_\epsilon(q)$
2. q is a core point

Given the inlier set S, a new point p belongs to a clustered region if one of the two conditions holds:

1. p is a core point of the set $S \cup p$
2. p is *directly density-reachable* by at least one core point q, $q \in S$

Such condition provides a binary check whether the local area of interest is well constrained or not, and it has been used in order to set a low sigma σ_L if it is satisfied, or a high sigma σ_H otherwise. However, as a step function, such decision rule lacks continuity at different density levels, treating regions at medium densities as badly constrained as empty regions.

2.5.2 A continuous density-uncertainty dependency

In our formulation, we propose to define σ as a continuous sigmoid function which spans between σ_H and σ_L (Figure 2.5).

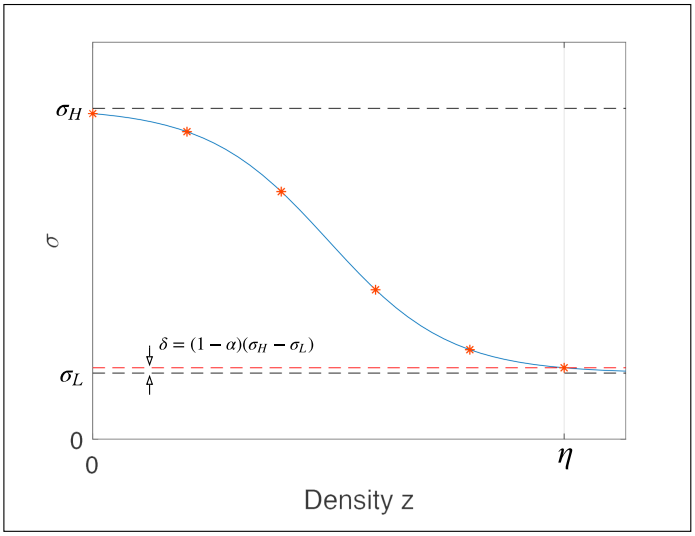


Figure 2.5: Sigmoid function which models in our algorithm the impact of the local observation density on the local uncertainty. The stars along the function represent the sampling locations which would be used by a histogram kernel density estimator with $n = 5$

While σ_L can be always set to $\sigma_L = 1$, as for the classic guided matching refinement methods, σ_H is a free parameter, which depends on the reliability of the initial solution, reflected by the scarcity of matches. Let us define as η the target density at which we have an α degree of confidence in the solution:

$$\sigma(\eta) = \alpha\sigma_L + (1-\alpha)\sigma_H \tag{2.5}$$

The use of α is due to the fact that the sigmoid reaches the bounding uncertainties σ_H and σ_L at $-\infty$ and $+\infty$. Thus, the degree of confidence α allows us to control the small disparity $\delta = (1-\alpha)(\sigma_H - \sigma_L)$ which is present at 0 and η densities between the reached σ value and the target uncertainties σ_H and σ_L, respectively (see Figure 2.5). The choice of α is not critical, and in all our experiments we set $\alpha = 0.99$.

We can then express the sigmoid as a function of the density z in the following way:

$$\sigma(z) = \sigma_L + \frac{\sigma_H - \sigma_L}{1 + e^{-b(z-\eta/2)}} \tag{2.6}$$

where the implicit steepness b has the form:

$$b = \frac{2}{\eta}\log\left(\frac{1-\alpha}{\alpha}\right) \tag{2.7}$$

We propose to evaluate the density z at each point $\mathbf{p}$ of the image using a Kernel Density Estimation (KDE) in the two-dimensional space:

$$z(p) = \frac{1}{h^2}\sum_{i=1}^{N} K\left(\frac{\mathbf{p} - \mathbf{p}_i}{h}\right) \tag{2.8}$$

Note that in Equation 2.8, differently from the classical KDE formulation, we do not normalize the density by the total number N of inliers. This is justified by the fact that N varies at each iteration of the algorithm, and thus this would require a continuous rescaling of the target density parameter η, without any change in the σ estimation output.

The choice of the kernel is not critical for our application, and a simple function as the histogram kernel:

$$K_H(\mathbf{u}) = \frac{1}{\pi} 1_{\|\mathbf{u}\| \le 1} \tag{2.9}$$

has shown good performance in our experiments, while more complex kernels as Epanechnikov:

$$K_e(\mathbf{u}) = \frac{2}{\pi}\left(1 - \|\mathbf{u}\|^2\right) 1_{\|\mathbf{u}\| \le 1} \tag{2.10}$$

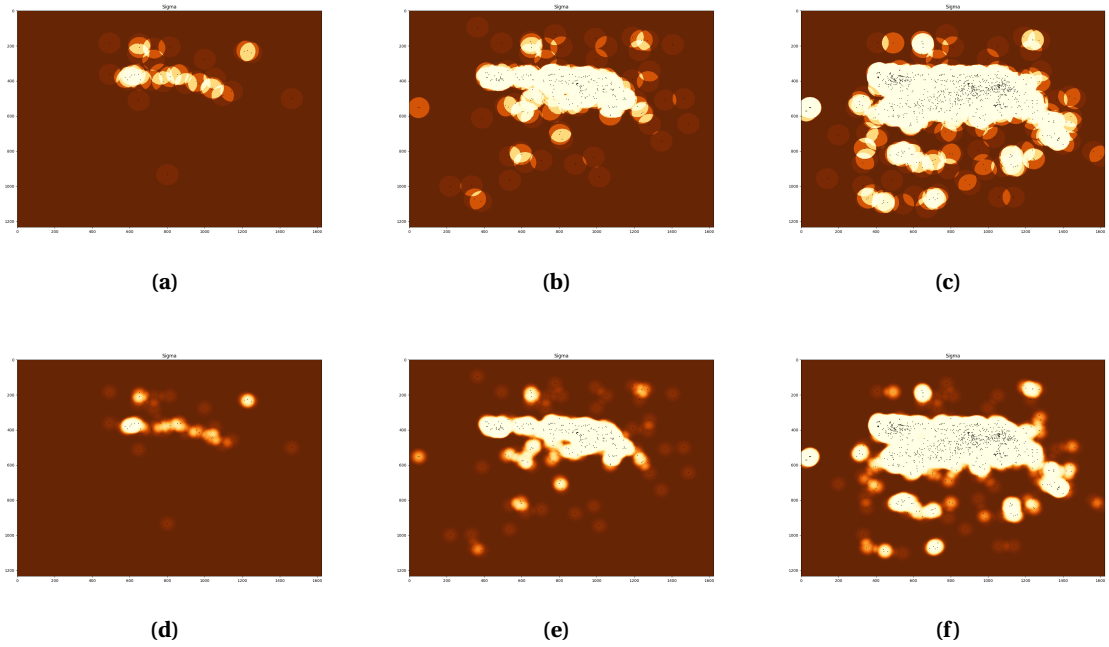


Figure 2.6: Sigmoid $\sigma(z)$ evaluation in the image space, with histogram kernel: (a) Iteration 0, (b) Iteration 5 (c) Iteration 30. The lighter the color, the lower σ value. As the method converges towards a robust solution, the well-constrained region grows in size. Smoother $\sigma(z)$ estimation can be performed with an Epanechnikov's kernel (d)(e)(f), but the higher computational does not correspond to substantial improvement in the result. The images refer to the *Regents Park* dataset, with camera 2 as the reference (Figure 6.1a)

do not introduce a significant advantage, while being more computationally costly (the kernels are normalized for the 2-D scenario occurring in our case). Figure 2.6 shows the gradual expansion of the well-constrained areas in the image space when using the histogram kernel (Figures 6.9a-6.9c) and the Epanechnikov kernel (Figures 6.9d-2.6f).

The target density η is a user defined quantity depending on the ideal interest point density for a specific type of scene. However, one may reason rather in terms of the expected number of corners n at a relevant spatial scale, while the actual numerical value of η involves a specific KDE function as well as the local relative corner layout. In our framework, we propose the following interpretation of the target density η with respect to the expected number of points via a given kernel K. The η target density may be represented as the density evaluated with n points at distance $h/2$ from the target:

$$\eta = \frac{n}{h^2} \mathrm{K}(\mathbf{v}) \tag{2.11}$$

with $\mathbf{v}$ being any vector such that $\|\mathbf{v}\| = 0.5$. This reasonable assumption allows us to relate the target density to the target number of points via the kernel. A critical parameter for the density estimation task is the bandwidth h, which identifies the radius of interest around a point. As we will show in the results, while the estimation task is sensible to the bandwidth, the actual error of the algorithm after convergence remains stable even for large variations of h.

2.6 Refining an existing pose estimation

In this section, we consider an adaptation of our algorithm which allows for data-driven refinement of an existing pose. Indeed, numerous existing datasets provide extrinsic calibrations, acquired with different techniques and characterized by various degrees of accuracy.

The main interest of the refinement procedure is that, as video data is analyzed, our algorithm may be used in order to refine the original estimation, which may lack precision in some specific areas of the image space. Moreover, pose refinement may be needed when the camera positions

might have changed slightly prior to an acquisition due to mechanical factors or due to internal behavior (e.g. pan-tilt-zoom cameras), but a reasonable prior pose is known. In robotic vision, the pose refinement is often applied to stereo rigs, but our setting is not suitable for continuous refinement in which the pose is time dependent (in this case Kalman filtering is the method of choice [27, 58, 115]). Our algorithm is suited for the accurate update of a stereo rig pose which is fixed but possibly different slightly from a reference value. Existing algorithms such as [97] rely on bucketing heuristics in order to enforce spatial uniformity of the observations, while devices which refine the stereo pose upon initialization such as the ZED camera from Stereolabs [148] run proprietary code.

The refinement procedure is similar to the estimation presented in Section 2.4, except the requirement of a *bootstrap period* at the beginning of the refinement process. The bootstrap period consists in building an initial set of matches by performing the acquisition and band filtering for several frames (setting inlier density $z(p) = 0$ for the entire period), by using the initial pose F_{init}. The period ends when a target number of matches is reached; we set this number to be proportional to the number m of raw matches acquired from the first frame pair of the sequence (we heuristically set this number to be $5m$, independently from the dataset). Please note that the bootstrap is different from a blind accumulation because it exploits via the band filtering the F_{init} that we intend to refine. The initial set of matches S_{init} will provide an approximate representation of the initial solution. The use of the bootstrap procedure follows from the fact that the convergence properties of our approach are related to the the growing percentage of matches which "vote" for a specific solution, thus the bootstrap period encourages a smooth convergence from F_{init} during the initial steps of the refinement. Viewed from another angle, this means that without any other information related to F_{init}, the bootstrap creates the support set which is needed in order to compute σ adaptively across the image space.

One may argue that in the context of pose refinement, a constant $\sigma = \sigma_{\mathrm{L}}$ would suffice. However, it is still advisable to use a variable σ parameter since the error introduced by the prior (e.g. the error on the tilting angle of a motorized surveillance camera) may be large enough in order to be impossible to sample correct observations; at the same time, the convergence should benefit from the adaptive σ in order to "follow" the pose variation as fast as possible.

2.7 Ground truth extraction

In order to perform a rigorous evaluation of the algorithm performance for real world scenes of relevant size, we propose the construction of a manual ground truth which allows to characterize the quality of the solution by performing a local analysis across the whole scene. The main motivator for such ground truth extraction comes from the observation that defining the error only at a global level may hide local high error regions, which may be harmful when using the estimation for tasks such as detection, tracking or depth estimation.

Our methodology for building this accurate ground truth data is the following (the outline is provided in Figure 2.7). We define an uniform grid of buckets which provides a partition of the ref-

1. Manual extraction of ground truth matches S_{gt}.
2. Robust RANSAC ($th = 1$) estimation of F matrix from the S_{gt}.
3. **if** inliers percentage $\geq \alpha$: stop.
4. Manual S_{gt} matches location refinement, from F matrix.
5. Go to step 2.

Figure 2.7: Outline of the ground truth extraction strategy

erence image, and we extract matches manually and uniformly inside the buckets belonging to the overlapping field of view. Since the human annotator may not find enough correspondences in a specific bucket (due to the presence of textureless regions), multiple image pairs may be exploited. In general the annotator should avoid local planar degeneracies inside a bucket. This means, one should try to select matches where the points are not co-planar, because the fundamental matrix could degenerate to well match points on a certain plane, but it could provide gross errors going further away from it. In such case the ground truth could not be able to detect degeneracy problems. However, even if it is encouraged, this property is not explicitly enforced, since, in general, it is not applicable everywhere (e.g. in parts of the image representing a building facade). Anyway, it still represents a valuable guideline, especially when extracting features from both the ground plane and pedestrians. In order to enforce a uniform distribution of ground truth points, to each bucket we assign a number M of matches, which is weighted by the portion of the bucket which belongs to the common field of view. Such extraction is essential in order to evaluate estimation errors even in regions where an automated process (followed by a manual validation) would not be able to identify meaningful and not degenerated interest points.

At the end of the uniform match extraction step, the measurement noise may be too high due to human impreciseness, and occasional gross annotation errors may also occur. Thus, the procedure is followed by the robust estimation of a fundamental matrix from the current set of matches, which is then used to refine the position of the generating matches, i.e. the human annotator is shown the annotations presenting high residuals in order to adjust them if necessary. The process is repeated iteratively, until we obtain a set of matches with half-pixel precision, which is at the same time large enough in order to guarantee a comprehensive evaluation of a candidate pose.

The error metrics we employ are the RMSE and the Max symmetric geometric error [62] on the ground truth. The use of the Max Error is the strictest possible metric, and is necessary for revealing localized errors, which would be mitigated by RMSE. Due to the stochastic nature of our estimation process, all the presented results are evaluated over 300 realizations of each test.

2.8 Results

We demonstrate the performance of our algorithm on three different datasets: *Regent's Park*, *PETS 2009* [40] and a laparoscope in-vivo procedure video provided by the Hamlyn Centre, Imperial College London [113]. The relevant information about the data content will be provided below.

Regarding the main parameters, we set for all the tests $\sigma_L = 1$, which is a common choice in guided matching covariance propagation methods [118]. The scale of σ_H depends on the matcher ease to associate features from the views, which is mainly reflected by the inlier set size, and by the inlier percentage of a robust estimation for a single frame (i.e. a small inlier set suggests an unstable estimation, and the value of σ_H should be set high enough in order to allow for a wider exploration). At the same time, small variations of the σ_H value have a negligible influence on the convergence behavior and on the final error. We set $\sigma_H = 5$ for all tests (with the exception of PETS 2009, see Section 2.8.2). The k parameter has shown to have little influence on the final results if chosen in a range of 2-5 (results with $k = 3$ are presented). We use as robust fundamental matrix estimator the ORSA [112] a-contrario framework, which exhibits good robustness without the need to set a sensitive threshold. Please note however that, while the robust method chosen has an influence on the final RMSE achieved, it has no effect on the actual convergence behavior of our approach, thus other methods based on the popular RANSAC [20, 131] may also be employed.

2.8.1 Pose estimation - Regent's Park dataset

The first part of our experiments is focused on estimating the relative pose in a realistic urban setup exhibiting typical challenges for this context.

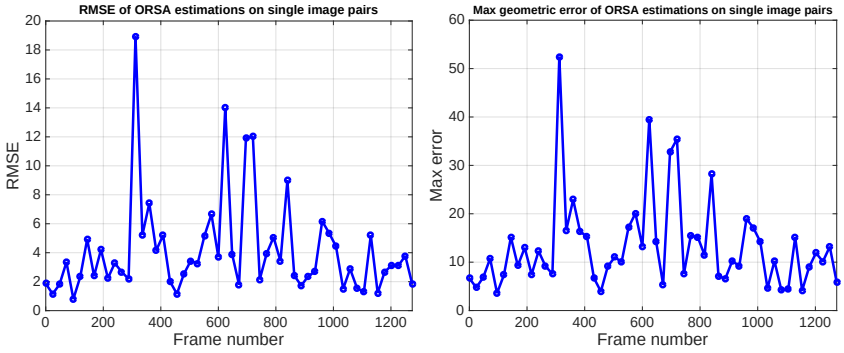


Figure 2.8: RMSE and Max geometric error by applying ORSA on each frame pair independently. Large variations in the result demonstrate the unreliability of estimation with still images in such setup. Streams from cameras 1 and 2 are used.

Experimental setup

We test our method on synchronized sequences recorded at Regent's Park Mosque, London. The camera network consists of three cameras installed on the roof (see Figure 2.1), labeled from 1 to 3. The analysis region is the rectangular shaped inner courtyard (the *sahn*), surrounded traditionally by arcades and other repetitive structures on all sides. The video streams capture the dynamic behavior of people who are free to move in the area. The grayscale video is recorded at 8 fps, with a 1624×1234 resolution. The stream is sampled each 3 seconds (i.e. $\Delta_t = 24$ frames).

Experimental results

We start by highlighting in Figure 2.8 the estimation errors obtained independently on single pairs of images extracted from the streams of cameras 1 and 2, with the ORSA estimator. For difficult scenes, the quality of the estimation is highly dependent on how the instantaneous configuration of the dynamic objects in the scene constrains the fundamental matrix, with large areas which may be left uncovered. In this specific case, the best achievable estimation has a maximum error of almost 4 pixels, which leaves room for a consistent improvement. Yet, the main underlying issue is that a single frame based estimation would provide a result of arbitrary quality. We evaluated at this stage the method in [153], which aims to extract matches iteratively from an image pair by enforcing spatial uniformity. This method fails to converge towards an acceptable solution (i.e. RMSE=245 for the first frame which was used for evaluation in Figure 2.11) as it does not cope with such a wide baseline correlated to a strong depth variation of the scene.

Section 2.3.2 underlined the importance of encouraging an uniform inlier distribution, and of accounting for the local inlier coverage in the estimation uncertainty. The two images in Figure 2.1 show a typical unbalanced inlier configuration which promotes high errors locally, and underline the importance of using a video sequence in the case of wide baseline cameras and large scale scenarios. Figure 2.9 shows the inlier matches which are maintained after running an estimation of the fundamental matrix between frames at $t = 74$ of cameras 2 and 3. We note the presence of a large region lacking correspondences on the bottom right of camera 2, where no feature matches can be acquired. As a result, that area could not be considered as reliable for guiding the geometry estimation during the subsequent iteration. Then, Figure 2.10b shows the spatial distribution of the symmetric geometric error on the left image. For each bucket of the image, we highlight the average error of the estimation with respect to the matches drawn from the ground truth points at that location. While approaching the area lacking inliers, we note the presence of high errors, which makes the single image pair approach unadapted for fitting the entire image space. While the overall RMSE=1.8 which is obtained from this estimation does not fully underline this major limitation, the Max geometric error equal to 7.53 reflects more accurately the local problems of the solution. This example also explains the significant variation, among different frames from

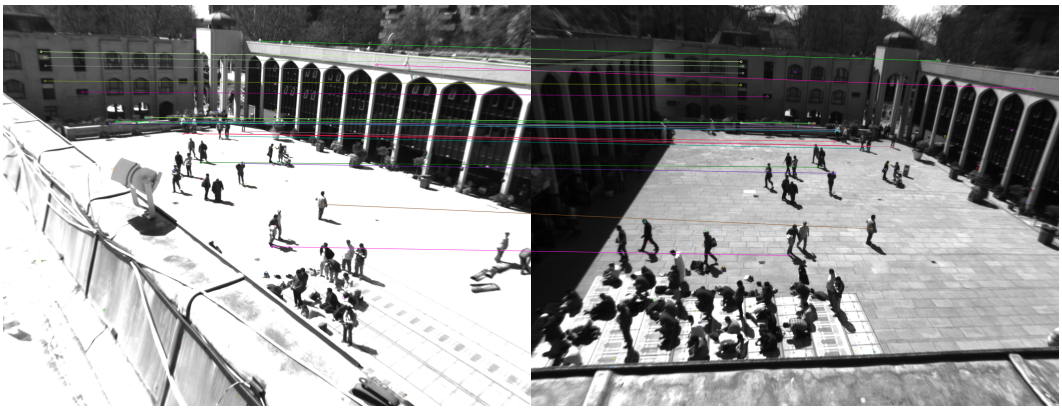


Figure 2.9: Sample pair of frames ($t = 74$) exhibiting an unbalanced inlier coverage (it is advisable to zoom in the electronic version for inspecting the inlier matches).

the same video, in the quality of the estimation which depends significantly on how the dynamic elements are disposed spatially. Finally, Figure 2.10c shows an example of the error distribution resulting from the proposed approach. The image shows a significant decrease of the error in areas which were challenging for single image pair methods, but also a reduction of the error on a global scale. The overall RMSE for this example is 0.46, while the Maximum geometric error is 1.23.

Next, we show our estimation results for cameras 1-2, presenting them against the results obtained by performing robust estimation on a set of matches accumulated naively from frame pairs (we call this strategy *All-matches*). Figure 2.11 shows the RMSE and Max geometric errors at different iterations of the algorithms. Our method is able to reduce the RMSE from 1.75 to 0.66, and to decrease consistently the Max error from 6.5 to 2.2 pixels. We note the robustness of our strategy, with the error following a monotonic decreasing trend after a few iterations. Conversely, *All-matches* presents large oscillations in time, which implies that getting more points from the video stream will not improve definitely the batch estimation result, introducing thus a frame window size choice problem. Our method also shows a smoother and faster convergence with respect to our previous work [120] which sets the adaptive σ parameter by using a binary decision threshold on inlier clustering (final RMSE 0.75 compared to 0.66 for the current algorithm).

The explanation of the behavior of the *All-matches* approach comes from the analysis of the inlier ratios estimated at each iteration (Figure 2.12). From the *All-matches* curve, we note that the inlier percentage obtained by accumulating matches drops monotonically. Thus the benefit of adding new points is negated by a lowering ratio of good matches, which implies the existence of a trade-off. On the other hand, our approach is based on a strict rejection procedure depending on the current inlier configuration. Subsequently, the inlier ratio follows the opposite trend, since being increasingly confident in the current solution, and using lower σ values will improve the probability of including only inliers as new matches. Such trend explains the robust convergence of our approach.

Figure 2.13 demonstrates the benefits of adapting the σ parameter of the covariance of the epipolar band to the actual spatial distribution of inlier matches in the image. It follows that by setting a $\sigma = \sigma_L = 1$, as in [118], we cannot add new information which is able to correct gross local errors in the estimation, leading to a much slower convergence which is never able to achieve performance, in terms of error, comparable to our strategy.

An important trait of an iterative pose estimation algorithm is its behavior in case of an adverse initialization. In Figure 2.14 we show the RMSE and Max geometric error evolution for the 1-2 pair when the most unfavorable initialization is selected (frame 312 in Figure 2.8). The algorithm is still able to recover and to decrease the RMSE from 18.7 to 0.78 and the Max error from 52 to 4.1 pixels. This result demonstrates that the algorithm is able to converge to a stable, low error solution, regardless of the starting point.

Then, we compare our adaptive σ solution with the use of a fixed $\sigma = \sigma_H$ for the band filtering

σ	RMSE	Max Error	Inliers ratio
σ_H	0.778	4.195	0.950
$\sigma(z), h = 30$	0.780	4.086	0.964
$\sigma(z), h = 60$	0.785	4.100	0.974
$\sigma(z), h = 100$	0.799	4.395	0.983

Table 2.1: RMSE, Max geometric error and inliers ratio on the worst initialization of camera pair 1-2 (*Regents Park* dataset) with different choices of the σ function and of the cross point density η ($n = 5$ is fixed for each selection of h). By using the sigmoid, the algorithm is capable of achieving comparable errors (less than 0.1 pixel difference) as an aggressive $\sigma = \sigma_H$ solution, at an higher inlier percentage (more than 3% difference).

$\eta = const$	RMSE	Max Error	Inliers ratio
$n = 1, h = 2.836$	0.838	4.389	0.974
$n = 5, h = 60$	0.785	4.100	0.974
$n = 14, h \approx 100$	0.775	4.012	0.973
$n = 56, h \approx 200$	0.779	4.135	0.976

Table 2.2: RMSE, Max geometric error and inliers ratio on the worst initialization of camera pair 1-2 (*Regents Park* dataset) at constant cross point density η and different choices of the bandwidth h.

step. Such an approach is more aggressive in the way it tries to add as many matches as possible by relaxing more the epipolar constraint. Although this strategy is able to achieve low errors occasionally, it does not trust the current solution locally more or less depending on the observations; this results in lower inlier ratios and a worse convergence stability. In Figure 2.15 we compare the inlier ratios when we use the sigmoid function or the $\sigma = \sigma_H$, and it is clear how the use of the sigmoid is able to promote a stronger, smoother increase, especially noticeable at the last iterations. Table 6.4 summarizes how the sigmoid approach is capable to achieve low errors which are comparable with an aggressive solution, guaranteeing at the same time an inlier ratio up to 0.98.

Table 6.4 also shows the effect of the choice of the cross point density η in the performance of the algorithm. The parameter η is expressed as the density of a desired number n of points in a h bandwidth. At a constant value of n, the higher the bandwidth h, the lower will be η. A lower η means that one gets confident sooner about the solution. This behavior is explained by the numbers in Table 6.4: higher values of η/lower values of h show the smallest errors, while lower values of η/higher values of h present the best inlier ratios. Therefore, the η parameter represents how aggressive the algorithm is in terms of adding new points. However, as it may be noticed from the same table, different choices of η do not have an important impact on the convergence and on the overall goodness of the final solution, which is a desirable property when consistent results with effortless parameter tuning are needed.

Table 2.2 shows the effects of the choice of the bandwidth parameter h, when η is kept constant. Varying the bandwidth entails different choices of the n parameter, which, being an integer number of points, tunes the resolution at which the sigmoid function is sampled. A specific value of n involves, when using a histogram kernel, sampling the same sigmoid curve $(n+1)$ times in the $[0, \eta]$ density interval, so higher the bandwidth, higher will be the sampling resolution. The first row of Table 2.2 corresponds to a binary selection of the σ value, equivalent to the one introduced in [120]. A significant error reduction is obtained by moving away from the binary representation of the inlier density. The table shows that increasing the resolution of the sigmoid has a benefit on the error levels, while maintaining stable the inlier ratio. However the error does not decrease monotonically as we increase h, because at the same time the density estimation loses its locality, providing inaccurate estimates of the boundaries between well and badly constrained regions.

Overall, as in the case of the choice of η, the selection of the bandwidth h, while being critical in pure density estimation tasks [141], does not affect the convergence of the algorithm. By setting h in a reasonable range, on the basis of the image size, one gets the lowest estimation errors.

Finally, we show the estimation results for the camera pair 2-3, using as starting point the worst possible initialization of the entire stream. Figure 2.16 shows again consistent results both in terms of RMSE and of Max error (curves are plotted in semilog scale for easier understanding). We are able to decrease the overall RMSE from 58.9 to 0.6, while reducing the Max error on the whole image space from 232.4 to 2 pixels.

2.8.2 Pose estimation versus pose refinement - PETS 2009

Experimental setup

PETS 2009[40] is a well-known and widely used dataset [18, 35, 146, 172] which provides multi-sensor sequences of moving pedestrians for tracking [102, 110, 155, 170, 184], density estimation and counting [21, 43, 154], and event recognition [45, 171]. The authors provide a full calibration of the system, which was performed using the Tsai calibration method [161]. From the calibration data, the ground truth pose estimation may be represented in the form of a fundamental matrix F_{GT}. The image resolution is 768×576 and the videos are recorded at 7 fps. We consider for experiments the *City Center 12:34* sequence, which contains a moderate number of freely moving pedestrians (Figure 2.17).

There are two main limitations of the provided geometry. First, the pose estimation is more accurate in the central part of the image which was covered comprehensively by the calibration procedure. This fact encourages the use of a limited area of interest for analysis which is more restrictive than the actual common field of view [108, 123, 163]. Secondly, the calibration allows for multiple camera data fusion at object level (mid level) or trajectory level (high level). However, and also owing to synchronization issues, the calibration is not accurate enough in order to allow pixel/voxel level (low level) data fusion algorithms [36, 79, 122, 139] to perform reliably due to significant pedestrian displacements [40, 163].

Since synchronization errors are critical for pose estimation, we have manually inspected a subset of the sequence in order to evaluate the temporal displacement at each timestep based on the pedestrian precise limb arrangements. Figure 2.18 presents these displacements for the first 100 timesteps, and the values confirm that most frame pairs exhibit a slight lag, which is occasionally significant. We chose to run the proposed pose estimation algorithm on the raw data in order to evaluate the robustness to persistent desynchronization.

Finally, some additional factors worth noting and leading to a difficult pose estimation problem are the slight errors related to radial distortion which are noticeable on the borders, the photometric differences among the distinct types of camera sensors and the significant scale variations.

We apply the same procedure as presented in Section 2.7, by manually selecting and then refining matches only on accurately synchronized frames. To the extent of our knowledge, this is the first time for PETS 2009 that the accuracy of the provided ground truth is also evaluated quantitatively (the standard approach being the validation against the provided ground truth, i.e. [77]).

Experimental results

In Figure 2.19 we show the errors when performing a robust estimation with the ORSA algorithm on a single image pair. For most frames we get extremely high RMSE values, which reflect how challenging the calibration procedure is in such scenario. For the PETS dataset a $\sigma_H = 200$ has been used, an order of magnitude higher than in the *Regents Park* dataset case. The choice of such high σ_H comes directly from the observation of the number of inlier matches retained by the single pair estimation. At frame 0 for example, only 9 inliers are maintained in the estimation, and this number is clearly insufficient in order to represent a robust support set for the inferred pose.

Full FOV analysis Table 2.3 shows the RMSE of the ground truth provided pose F_{GT}, compared with that of our algorithm, at different initialization times, for the entire area which is visible from

t_0	Init RMSE	RMSE
F_{GT}	-	2.58
$t_0 = 0$	621.88	3.32
$t_0 = 99$	6.05	3.02

Table 2.3: RMSE on F_{GT} and different initializations times t_0 of our algorithm on the PETS 2009 dataset. The final error is always comparable with the ground truth one, while the initial RMSE not affecting the convergence of the solution to a close final error.

t_0	Init RMSE	RMSE	Init Max	Max
F_{GT}	-	1.14	-	2.65
$t_0 = 0$	612.45	1.14	2336.94	3.83
$t_0 = 99$	4.79	1.05	15.05	2.98
$F_{refined}$	1.14	0.83	2.65	2.08

Table 2.4: RMSE and Max error on the F_{GT}, and our algorithm having different initializations: pose estimation starting at time $t_0 = 0$ or $t_0 = 99$, and pose refinement with initialization provided by F_{GT}. Errors are evaluated on the region of interest R0 of PETS 2009 dataset.

the two cameras. Comparing the solution directly with F_{GT}, without using the manual ground truth, would have hidden away the actual F_{GT} imprecision. The RMSE values obtained by running our algorithm directly on the video sequence are less than 1 pixel off compared to the errors of F_{GT} estimated using the Tsai calibration. Moreover, for two different initialization times characterized by a low RMSE (6.05 pixels) and by the worst observable configuration (620.2 pixels), we note the minimal impact on the final convergence result. Regarding the Max error, the F_{GT} presents a 11.54 pixel error, while our method reaches Max error of 16.36 (starting from 3444.14) for $t_0 = 0$, and of 15.34 (starting from 17.97) for $t_0 = 99$.

AOI analysis First of all, the localization of the highest errors in the bottom left area of camera 1 suggests that border errors are less reliable for the analysis due to the impact of the image undistortion. More importantly, our method, while being able to decrease significantly the Max error, presents a higher Max final error than F_{GT} due to the fact that on the image borders no pedestrian action occurs (for the manual annotations, we used moving pedestrians from other sequences of the dataset in order to cover border areas). Thus, the lack of observations limits the algorithm to refining locally the solution. For the two reasons above, we consider a region of interest R0 on camera 1, which is defined as the moving pedestrian convex hull and which allows us to provide an unbiased comparison in the actual analysis area used for the detection and tracking tasks (see Figure 2.21 for the spatial extent of R0). Such area consists in all the walkway region, including also for completeness the area which is strongly cluttered by the tree in camera 3.

Table 2.4 shows the errors for the F_{GT} and our algorithm (at different initialization times) in the R0 region. Even when starting from an almost random initialization ($t_0 = 0$), our method is able to achieve the same RMSE as the F_{GT} (even slightly lower in the case of $t_0 = 99$). The Max error for the two solutions is close to the F_{GT} one, showing that our method is able to provide a good quality solution in the area of interest without relying on any calibration device, as in the F_{GT} case. Figure 2.20 shows the error variation in time (both RMSE and Max) when we start from the worst possible initialization ($t_0 = 0$). The characterization of the algorithm behavior in such case is critical due to the use of a large value for σ_H, which, being more permissive, may introduce instabilities in the results. However, due to the use of the sigmoid, the algorithm is capable after a few steps to follow a smooth convergence, due to the gradual increase of confidence in the output solution at higher inliers densities.

Pose refinement Finally, we show the results obtained when refining an existing pose, which is F_{GT} in our case. The interest of pose refinement is that the estimation of F_{GT} has been carried out with helper objects, which may not cover the entire image space exhaustively. Starting from F_{GT}, we aim to refine the pose in the tracking region of interest R0, by including the rich visual information that is provided by the actual data.

Table 2.4 shows the RMSE and Max error of F_{GT} compared with $F_{refined}$, obtained by refining the provided pose on the entire *City Center 12:34* sequence. The $F_{refined}$ achieves a consistent improvement of both RMSE and Max error. In Figure 2.21 it is possible to inspect the average errors for each bucket in R0. The $F_{refined}$ is able to reduce the estimation errors across almost all the discretized image space, and to reach an average error per bucket below 1 pixel, except on two buckets for which the average error is 1.1 pixels.

2.8.3 Pose refinement - Hamlyn Centre Laparoscopic / Endoscopic Video Dataset

Experimental setup

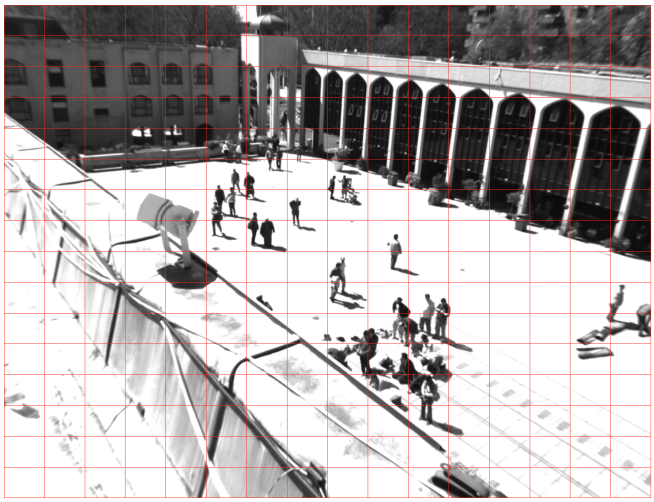
The dataset [113] consists of multiple monocular and stereo medical video sequences which are widely used for validating a variety of applications such as Shape-from-Shading [167], surface reconstruction [94, 104], deformable surface tracking [129, 177, 178] and SLAM [103, 114, 159]. For all sequences, the dataset maintainers provide high-quality intrinsic and extrinsic calibration information, estimated in the laboratory using a checkerboard helper object. For our experiments, we consider stereo data provided by a moving laparoscope visualizing an abdominal porcine wall (Dataset6). The image size is 640×480, and the video is recorded at 30 fps. We choose a sampling value $\Delta_t = 15$.

Experimental results

For the medical dataset, our objective is to refine the pose which was provided for the stereo rig, given that for stereo navigation or dense reconstruction algorithms any stereo calibration error weighs on the 3D estimations, since the stereo pose is assumed to be fixed.

The creation of a manually annotated ground truth for validating the pose is unfeasible in practice on this type of data due to the absence of highly salient small structures which are needed by a human subject. Thus, we demonstrate the interest of our refinement step using the live recorded data by showing some qualitative results on eight manually matched structures. The $\sigma_H = 5$ remains unchanged with respect to the *Regents Park* dataset tests. Figure 2.22 demonstrates the improvements of the proposed refined matrix on the test point selected in the image space. The red epipolar line is drawn from the F_{GT} matrix provided by the dataset maintainers. While F_{GT} shows good performance in the left part of the space, it presents higher errors (up to 3 pixels on the test points) in some border regions of the image, especially in the right and top parts. The green epipolar line is drawn from the $F_{refined}$ matrix, which decreases the errors in the critical areas, while maintaining good performance in the parts which are already well covered (our solution achieves less than 0.5 pixels error in the test points).

Such refinement step has no additional cost in terms of data acquisition (the already available raw data can be used), and is capable to provide a better quality calibration which is essential when applied to e.g. 3D projection and reconstruction tasks.



(a)

0.34	0.37				0.07	0.15									
0.34	0.45	0.20	0.26	0.34	0.29	0.41	0.38	0.54	0.73	0.67	1.18				
0.77	0.45	0.34	0.42	0.30	0.29	0.37	0.17	0.58	0.47	0.76	1.16				
0.72	0.81	0.66	0.49	0.37	0.35	0.33	0.36	0.34	0.32	0.27	0.23				
	0.72	0.72	0.76	0.40	0.48	0.42	0.31	0.39	0.71	0.39	0.41				
		1.31	1.08	0.9	1.06	0.30	0.27	0.45	0.92	0.88	1.39				
				1.87	1.20	0.62	0.49	0.39	0.88	1.16	2.38				
				2.13	1.20	0.92	0.84	0.62	0.62	1.22	2.69	3.44			
					1.24	1.15	0.74	0.62	0.86	1.62	2.27	3.88	5.96		
						0.85	0.37	0.42	0.66	1.37	2.95	4.80	6.32		
							0.36	0.31	0.45	1.61	3.38	4.94	6.85		
								0.50	0.95	2.13	3.69	5.43			
									0.15	1.24					

(b)

0.12	0.27				0.38	0.39									
0.12	0.27	0.17	0.24	0.50	0.53	0.58	0.62	0.54	0.06	0.26	0.15				
0.25	0.33	0.26	0.36	0.33	0.54	0.33	0.07	0.63	0.42	0.33	0.17				
0.20	0.22	0.28	0.38	0.39	0.55	0.45	0.35	0.42	0.52	0.32	0.29				
	0.28	0.19	0.34	0.20	0.43	0.49	0.48	0.50	0.35	0.21	0.18				
		0.18	0.47	0.35	0.47	0.18	0.23	0.22	0.16	0.30	0.13				
				0.39	0.41	0.39	0.30	0.13	0.42	0.29	0.19				
				0.34	0.16	0.46	0.46	0.54	0.67	0.48	0.42	0.46			
					0.65	0.16	0.42	0.43	0.23	0.23	0.38	0.35	0.44		
						0.42	0.54	0.52	0.45	0.52	0.33	0.27	0.53		
							0.31	0.36	0.41	0.17	0.26	0.19	0.25		
								0.31	0.11	0.48	0.20	0.27			
									0.25	0.24					

(c)

Figure 2.10: Resulting spatial distribution of the symmetric geometric error with respect to a dense manually annotated ground truth. Errors less than 1 pixel are highlighted in green, between 1 and 2 pixels in yellow, and more than 2 pixels in red. The gray buckets correspond to areas outside the common field of view. (a) Reference frame subdivided in buckets. (b) Average errors per bucket using the single image frame. (c) Average errors per bucket using the proposed method.

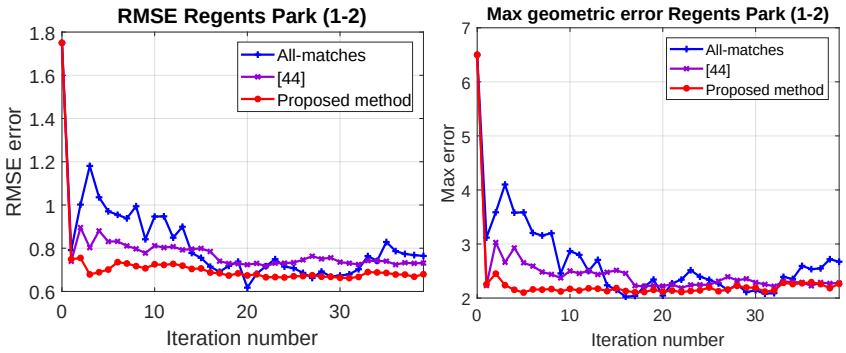


Figure 2.11: RMSE and Max geometric error by applying the *All-matches* strategy, the method in [120] and our algorithm on 1-2 camera pair of *Regents Park* dataset. Our selection is more reliable, and we are able to improve the initial estimation significantly and robustly, with a lower RMSE and less oscillations than [120].

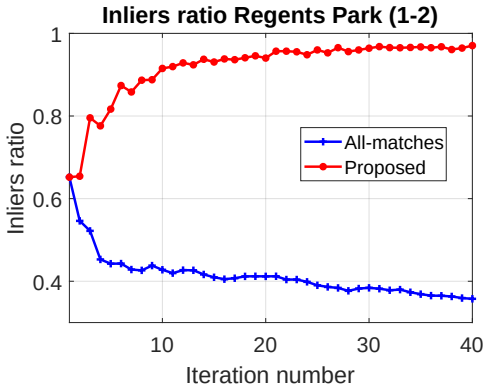


Figure 2.12: The inliers ratio at each iteration for the *All-matches* and for our approach.

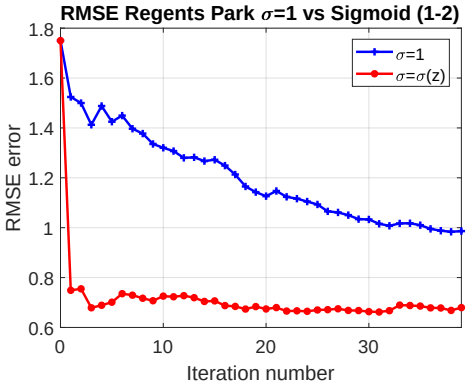


Figure 2.13: RMSE by applying our method on the 1-2 camera pair by using a fixed $\sigma = \sigma_L = 1$ value, and by using the adaptive sigmoid shaped σ introduced in Section 2.5

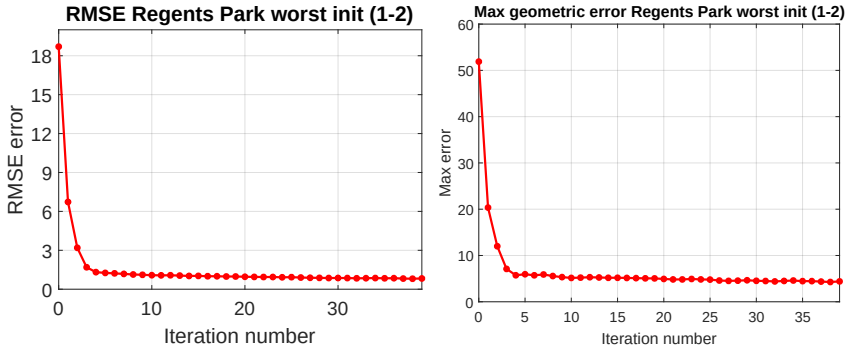


Figure 2.14: RMSE and Max geometric error by applying our algorithm on the worst possible initialization of the 1-2 camera pair sequence (*Regents Park* dataset). Our estimation is cabable of successfully converge independently of the initialization chosen

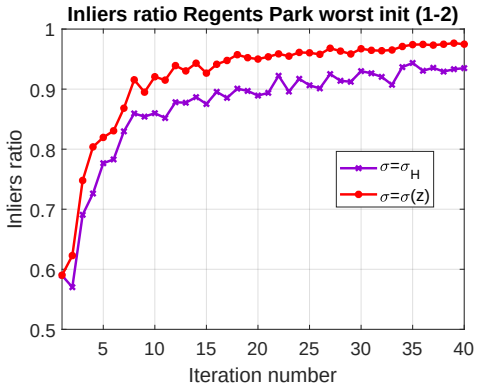


Figure 2.15: The inliers ratio at each iteration on the worst initialization of the camera pair 1-2 sequence (*Regents Park* dataset) by using a fixed $\sigma = \sigma_H = 5$ or our adaptive sigmoid shaped σ introduced in Section 2.5.

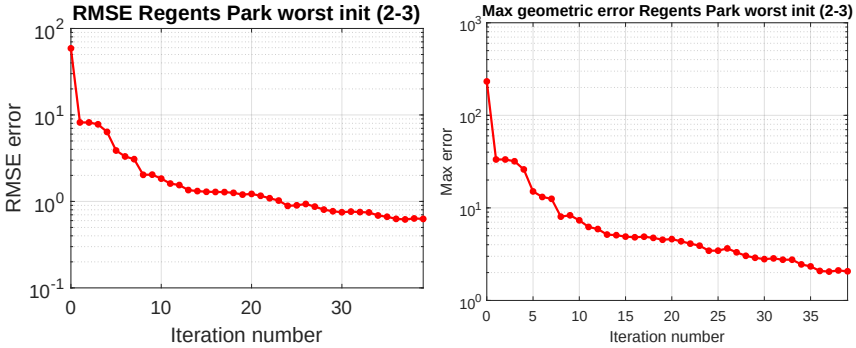


Figure 2.16: RMSE and Max geometric error (in semilog scale) obtained by applying our method for the 2-3 camera pair (*Regents Park* dataset), with the worst possible initialization

(a) **(b)**

Figure 2.17: Sample frames from PETS 2009 dataset. (a) Camera 1, (b) Camera 3

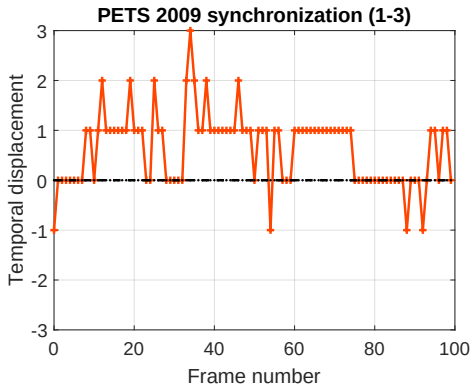


Figure 2.18: Temporal displacement (i.e. synchronization error value) of the first 100 frames from view 3 with respect to the ones of view 1 of the *City Center 12:34* sequence (PETS 2009 dataset).

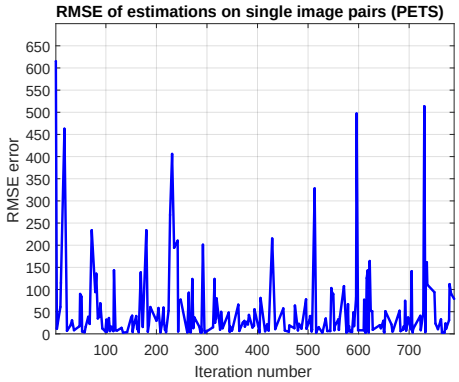


Figure 2.19: RMSE by applying ORSA in each frame pair of the *City Center 12:34* (PETS 2009) independently. Streams from cameras 1 and 3 are used.

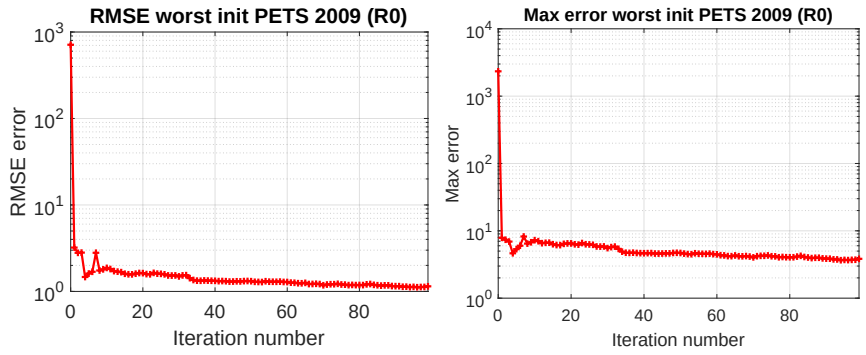


Figure 2.20: RMSE and Max geometric error (in semilog scale) obtained by applying our method on region R0 (PETS 2009), with the worst possible initialization ($t_0 = 0$)

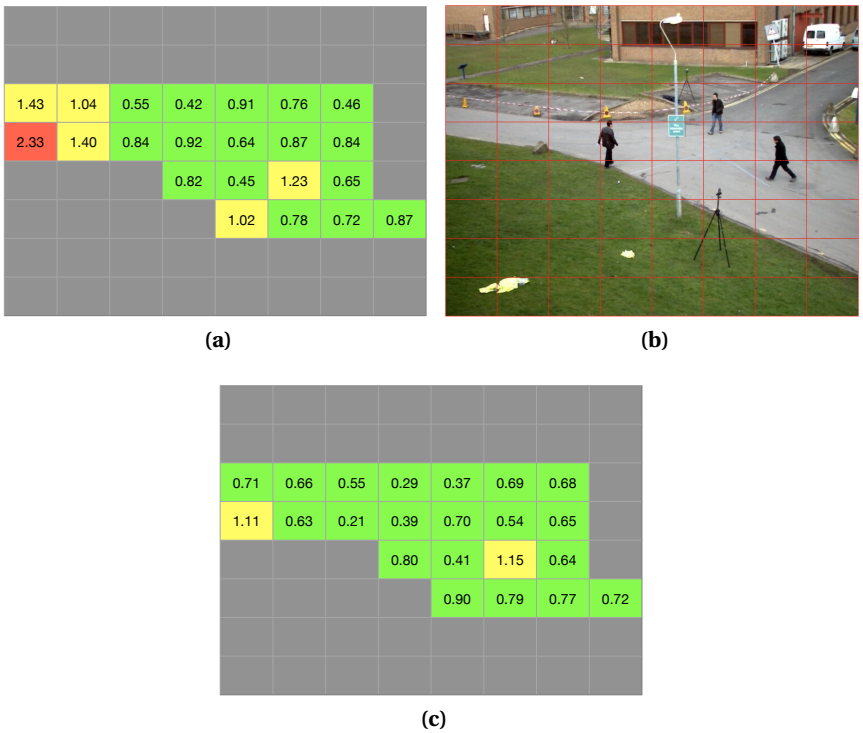
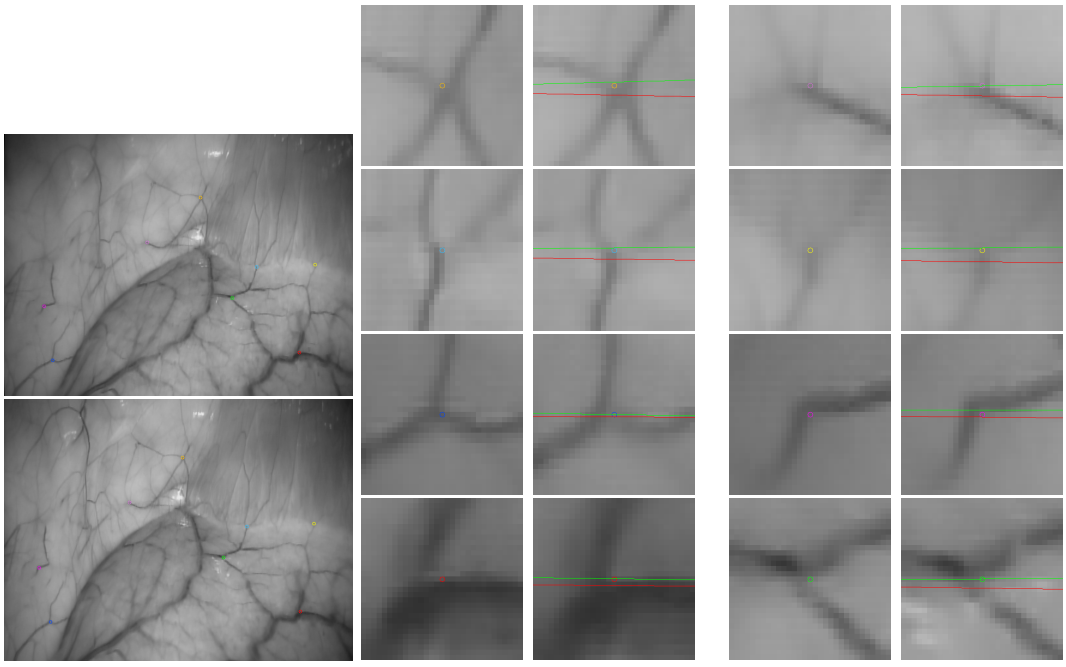


Figure 2.21: Resulting spatial distribution of the symmetric geometric error with respect to a dense manually annotated ground truth (PETS 2009), in the region of interest R0 (colored buckets). Errors less than 1 pixel are highlighted in green, between 1 and 2 pixels in yellow, and more than 2 pixels in red. (a) Average errors per bucket using the provided F_{GT}. (b) Reference frame of subdivided in buckets. (c) Average errors per bucket after executing the proposed refinement.



(a) (b)

Figure 2.22: Qualitative results obtained from the refinement of the provided pose of *Hamlyn Centre Laparoscopic/Endoscopic Video* dataset. (a) Stereo pair, with eight manually selected control points highlighted in different colors. (b) Zoomed views of the local patches around the control points (their color refers to the one in subfigure (a)), with two epipolar lines being drawn each time: the one from the provided F_{GT} (red) and the one from our refinement (green). A small but noticeable displacement is present for F_{GT}; the proposed refinement is successful in removing it.

2.9 Conclusion

In this chapter we have detailed a new approach for solving difficult relative pose estimation problems based on a guided selection of new matches from video. We select new matches in order to constrain the estimation robustly, by adapting the search process with respect to the local inlier distribution. This results in a fast convergence towards a high-quality solution, which is being highlighted by the manual ground-truth we created for two difficult scenes. In our experiments, we show that this video accumulation strategy converges robustly to globally effective pose estimations, irrespectively of the scene configuration during initialization. We have also proposed an extension able to perform data-driven pose refinement based on a prior pose initialization, and which is aimed at stereo systems requiring frequent high-quality extrinsic re-calibrations. During experiments, our self-calibration procedure was able to improve consistently the prior pose with no overhead in terms of data acquisition procedures.

Our approach can be largely beneficial for such fields where online precise re-calibration is crucial. For example, pan-tilt-zoom surveillance cameras need constant re-calibration in order to be exploited for multiple views pedestrian analysis. Due to the continuous movement of such equipment, the algorithm has to be able to *follow* the change in pose by updating the fundamental matrix estimation at every new frame. Since our method benefits from smooth and convergent improvements from an imprecise solution, it would be suitable for this scope.

The proposed work is crucial for the following steps of our project. Once the cameras are calibrated with a high confidence in the solution, multiple views can be exploited concurrently for pedestrian detection in the 3D space. Moreover, it allows for ground plane registration among all the views, making possible to perform smart detection fusion directly in the metric space, by overcoming critical problems in the image domains, e.g. ambiguous distances and occlusion.

Part II

Multiple Camera Pedestrian Detection in Dense Crowds

Chapter 3

Multiple Camera Pedestrian Detection: an overview

Contents

3.1 Motivation

Pedestrian detection is a fundamental task in computer vision, closely related to applications such as video surveillance, autonomous driving or action recognition. Some characteristics of the analyzed scene and camera setup improve significantly the reliability of the detection: a low pedestrian density, close to vertical optical axis, and a good resolution representation of individuals.

However, the recent focus on the analysis of large, densely crowded outdoor areas underlines the current limitations in presence of persistent, heavy clutter. Detection strategies based on multiple overlapping views may be used to achieve more robust inference provided that sensor data are fused prior to detection. This still leaves open the problem of *how to associate data* among views which may exhibit significant geometric and photometric variation. Above all, the *joint projection* of visual information in a common reference system is conditioned by an accurate estimation of the relative camera poses, and of the ground plane.

As some of the underlying assumptions are violated (i.e. persistent clutter for foreground extraction, heavy occlusions for part-based detectors, homogeneous crowd dynamics for independent motion based inference), the detector breaks down. Moreover, resilience to camera pose variations or to people appearance comes at a cost, in the form of human intervention for calibration procedures or for scene-dependent supervised learning.

By addressing the problems highlighted above, we aim to build a fully unsupervised pedestrian detector which can be applicable for head detection to large, cluttered scene analysis. Such detector may exhibit complementary strengths and weaknesses with respect to the classical single view supervised detectors, so providing a favorable configuration for data fusion.

3.2 Ground plane registration: variable height-homographies

In order to propose a multiple camera detector, a metric registration of the ground plane is an essential preliminary step. Having such information, one can not only relate pixels in different views

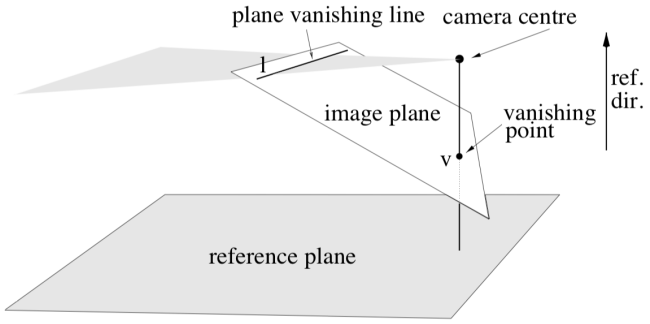


Figure 3.1: Baseline geometry. The plane vanishing line corresponds to the intersection between the image plane and the plane parallel to the reference plane and passing through the camera center. The vertical vanishing point is the intersection of the image plane with the line parallel to the camera reference direction and passing through the camera center. Image taken from [22].

by their estimated height above the ground, but one can also infer the metric distance between any pixel in a single view. The first idea is to measure the metric distance between any two 3D parallel planes (parallel to the ground, in our case), when two image points are known to be lying on those two planes. The authors of [22] show that if the vanishing points of the image are known, relative distances can be measured by cross-ratio relationships. In the image space, a projective transformation between two image points belonging to two parallel world planes is called **planar homology** [22]. Let us consider the world-to-image projection matrix **P** as:

$$\mathbf{x} = \mathbf{P}\mathbf{X} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_3 & \mathbf{p}_4 \end{bmatrix} \mathbf{X},$$

where $\mathbf{x} \sim (x, y, w)$, $\mathbf{X} \sim (\mathrm{X}, \mathrm{Y}, \mathrm{Z}, \mathrm{W})$ are homogeneous vectors.

Let us denote the vanishing points on both directions as $\mathbf{v}_\mathrm{X}$, $\mathbf{v}_\mathrm{Y}$ and $\mathbf{v}$. The points $\mathbf{v}_\mathrm{X}$ and $\mathbf{v}_\mathrm{Y}$ are two distinct points on the **vanishing line l**, which corresponds to the intersection of the image plane with the plane parallel to the ground plane and passing through the camera center. The point **v** is the **vertical vanishing point**, corresponding to the intersection of the image plane with a line parallel to the camera's reference direction and passing through the camera center (see Figure 3.1).

One can re-parametrize the projection matrix **P** as a function of the vanishing points. The first three columns can be expressed as the three vanishing points, while the last column is the projection of the world's origin. For linear independence such projection cannot lie on the vanishing line, and thus is arbitrarily chosen as $\widehat{\mathbf{l}} = \mathbf{l} / \|\mathbf{l}\|$. Finally [22]:

$$\mathbf{P} = \begin{bmatrix} \mathbf{v}_\mathrm{X} & \mathbf{v}_\mathrm{Y} & \alpha\mathbf{v} & \widehat{\mathbf{l}} \end{bmatrix},$$

where α is a metric scale factor.

Let us consider two points **b** and **t**, the first lying on the ground plane and the second on a parallel plane of height h. The authors of [22] demonstrate the following relationship:

$$\alpha h = \frac{-\|\mathbf{b} \times \mathbf{t}\|}{(\widehat{\mathbf{l}} \cdot \mathbf{b}) \|\mathbf{v} \times \mathbf{t}\|}. \tag{3.1}$$

Thus, if α is known, one can have the absolute metric distance h between the two points. On the other hand, if h is known, then one compute the value of α. Thus, in theory, a single pair of parallel points at known height can be used to solve the metric scale ambiguity. Figure 4.2 depicts how cross-ratios are employed for distance calculation for a homology transformation. Four points, **b**, **t**, the point **i** (the intersection between the vanishing line and and the line passing through **b** and **t**), and the vertical vanishing point **v**, are sufficient for relative distance calculation.

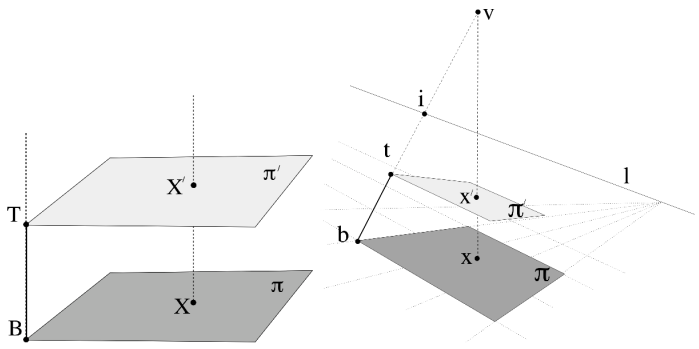


Figure 3.2: Schematic representation of homology between two planes. The first figure shows the 3D relationship between two reference points **B** and **T** and two generic points **X** and **X**′. The second figure shows their respective projections in the image space. In order to evaluate the relative distance between the reference points, the cross-ratio between four points is sufficient. Such knowledge can be applied to any new point pair. Image taken from [22].

With the given representation of the camera matrix P, one translates easily the world coordinate system along the reference direction of the camera. For a plane at an height h over the ground, one has [22]:

$$\mathbf{P}_h = \left[\mathbf{v}_X \quad \mathbf{v}_Y \quad \alpha\mathbf{v} \quad \alpha h\mathbf{v} + \widehat{\mathbf{l}}\right].$$

One can get the plane to image homography by removing the third column from the matrix **P** [22];

$$\mathbf{H}_0 = \left[\mathbf{v}_X \quad \mathbf{v}_Y \quad \widehat{\mathbf{l}}\right]$$

$$\mathbf{H}^h = \left[\mathbf{v}_X \quad \mathbf{v}_Y \quad \alpha h\mathbf{v} + \widehat{\mathbf{l}}\right].$$

Finally the planar homology matrix:

$$\mathbf{B}^h = \mathbf{H}^h \left(\mathbf{H}^0\right)^{-1} = \mathrm{I} + \alpha h\mathbf{v}\widehat{\mathbf{l}}^{\mathrm{T}}, \tag{3.2}$$

maps image points on the reference plane to image points on the parallel plane at distance h. The ability to evaluate the homology matrix between each plane parallel to the reference allows us to have a mapping between any point on one image i, to a single point in another image j, given that we know the metric height of the point, and the ground plane homography $\mathbf{H}^0_{ij}$. Basically, a point p_i is projected into the ground with the inverse of the homology $\mathbf{B}^h_i$, then mapped in the image j with the ground plane homography $\mathbf{H}^0_{ij}$, and then projected to the corresponding point at the correct height with the homology matrix $\mathbf{B}^h_j$. Thus, we can define the **variable-height homography** $\mathbf{H}^{\mathbf{h}}_{\mathbf{ij}}$ as:

$$\mathbf{H}^h_{ij} = \mathbf{B}^h_j \mathbf{H}^0_{ij} \left(\mathbf{B}^h_i\right)^{-1} \tag{3.3}$$

3.3 Related works

There exists a large body of research for pedestrian detection, therefore the following sections focus on approaches suited for moderate to high density scenes. For identifying strongly occluding people, multiple cameras are better positioned in order to resolve ambiguities in at least a subset of the available views. Depending on how sensor data are combined, detection methods rely on raw data level (low-level), on object level (mid-level) or on trajectory level (high-level) data fusion.

3.3.1 Scene geometry

Difficult detection scenarios benefit from strategies which avoid performing the detections in individual views. However, the lower the fusion level, the greater will be the impact of the geometry

Figure 3.3: Camera network for the pedestrian detection experiments in [36].

alignment among the cameras and the scene. In a number of studies, the relative camera poses and the ground plane orientation are identified by relying on a robust estimation of the ground plane homography using ground inliers [79, 149]. A precise estimation requires a large uniformly distributed set of observations which are difficult to obtain, considering the homogeneity of the ground and the potentially significant pose variations among cameras. Some studies rely on manual ground annotations [26, 56], but for such a solution to be constrained accurately across the work area, the annotations should ideally be uniformly and densely performed. Finally, classical extrinsic calibration relies on specific objects being observed in multiple views prior to the analysis, but this approach does not scale to large outdoor areas. Moreover, the concomitant presence of calibration objects in different views [36] during analysis is non-viable in cluttered realistic conditions.

3.3.2 Ground plane projection

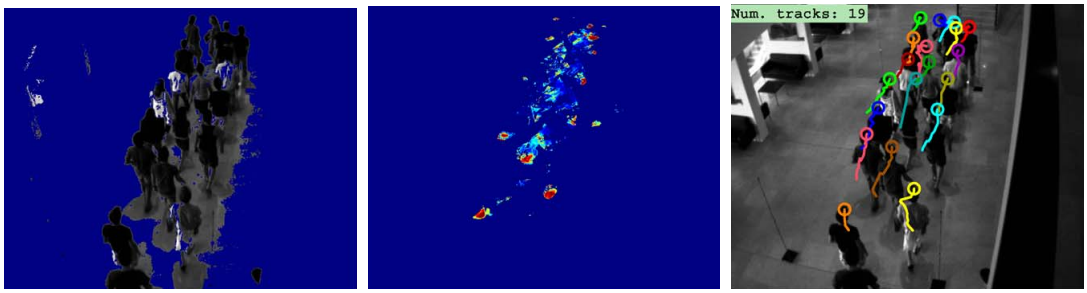
A common ground plane hypothesis is adopted by most multiple view based detectors (a notable exception being [1]), due to the simplificatory assumptions it allows for data fusion. In order to simplify data association among cameras while at the same time avoiding the difficult detection task in single views, the vast majority of subsequent works rely on foreground extraction and the combination of foreground maps in the ground plane reference, as opposed to high-level approaches which apply multi-view homographies onto single-view detections [80]. Early works such as [111] relied on basic appearance cues such as color in order to find correspondences across cameras. In subsequent studies, the data fusion may be performed under various forms, such as a probabilistic occupancy map relying on a generative model [42], silhouette based extraction [2, 51], stochastic spatial models [47, 162], or a joint foreground and appearance likelihood objective function [93].

3.3.3 Multiple homography methods

Methods such as the ones proposed by Khan and Shah [79] or by Eshel and Moses [36] rely on a detection performed at varying heights with respect to the reference plane. Moreover, while Khan and Shah [79] rely on non metric multiple homographies, Eshel and Moses [36] estimate the metric scale by manual intervention.

Tracking in a Dense Crowd Using Multiple Cameras [36]

The work in [36] aims to detect and track pedestrians from multiple views. Starting from a camera network (see Figure 3.3), a reference image is selected. Let us denote as a reference the camera 1. The authors use some helper objects (LED poles with lights at known heights), in order to perform the estimation of variable-height homographies $\mathbf{H}^{h}_{1,i}$. Background subtraction is performed independently on each view in order to focus only on the moving objects.



(a) **(b)** **(c)**

Figure 3.4: Main steps of the method in [36]. (a) Background subtraction. (b) Saliency map of intensity variances. (c) Segmented detections and tracking.

Then, for a fixed set of heights ($150cm$, $155cm$, ..., $190cm$), the following steps are performed:

1. For each foreground pixel in the reference view, compute a hyperpixel containing the pixel itself and its homographic projections in the other views.

2. Create a saliency map with the intensity variances among the components of each hyperpixel.

3. Perform hysteresis thresholding, head segmentation and top-head detection.

The top-heads detections from multiple candidate heights are then fused in order to get a cumulative detection map.

The main limitations of such approach (that we aim to overcome with our method) summarize as follows:

1. The estimation of the homography requires manual intervention, due to the use of helper objects.

2. The intensity variance correlation metric is highly sensitive to perspective and illumination, thus it is not applicable in real surveillance scenarios.

3. Camera setup is highly constrained (high pitch angles, limited baselines), and, in the experiments, more than 5 cameras are needed in order to achieve acceptable performance.

Moreover, all the methods presented above, other variations based on 3D carving [128, 139] and extensions based on the spatio-temporal evolution of detections [17, 71, 123] rely heavily at an incipient stage on foreground extraction, and are significantly impaired if the foreground cannot be segmented. A typical example is [79] which requires feet visibility in order to work. Unfortunately, cameras placed with low pitch angles as it is generally the case for high-density crowd surveillance would not observe sufficient empty areas among proximate pedestrians in order to benefit from foreground extraction.

In conclusion, although methods exploiting multiple cameras for pedestrian detection have been developed within the Computer Vision community in the last decade, a more attentive analysis underlines that all these methods require strong assumptions about the scene layout or content which render them ineffective in large scale, outdoor areas where the environment can not be controlled accordingly. Moreover, in order to overcome the challenges raised by realistic urban scenes, not only purely engineering but also methodological advances are necessary in order to tackle the aspects related to illumination and perspective variation, or to the efficient search of a global solution.

Chapter 4

Geometry-based Multiple Camera Pedestrian Detection

Contents

We are inspired by the work of Eshel and Moses [36], who underline the graceful degradation of homography-based head detection as crowd density increases. In our work, we turn the pedestrian detection problem in a height map estimation, where the prior on the neighborhood is easier to formulate than in depth map based approaches, thus jointly estimating occupation and 3D location of each pedestrian. The contributions of our work can be summarized as follows: **(i)** we model the head detection problem as a stereo MRF-based optimization of a dense pedestrian height map, which exploits a first-order regularization term which constraints the height variation; **(ii)** in order to be able to cope with height labels, we demonstrate a new fully unsupervised method for relative camera pose and homography estimation which avoids placing calibrating objects inside the investigated area, either during the detection algorithm or prior to the analysis; **(iii)** we rely on a data association cost among camera views which is able to cope with intensity and perspective variations specific to outdoor.

By addressing these key points, we believe the proposed advances will improve the applicability of geometry-based strategies for head detection to large, cluttered scene analysis (Fig. 4.1).

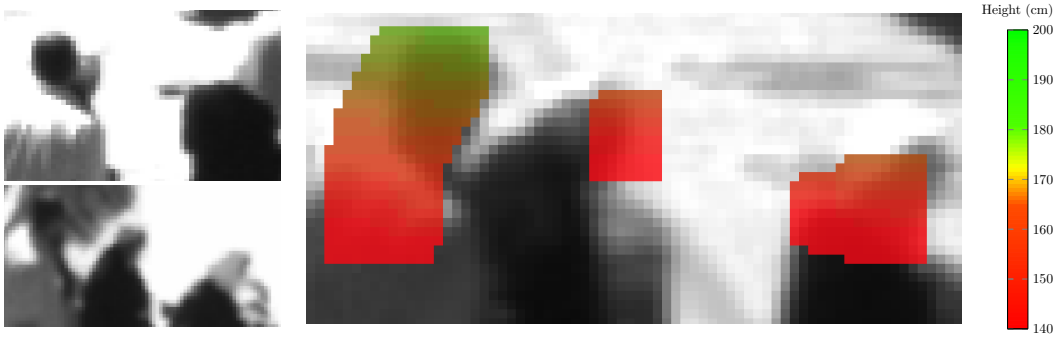


Figure 4.1: Head detection detail with a height map overlay in the central camera view. Two additional views (see left) are used for the height map estimation. Note the height gradient following the local vertical direction, and the middle detection for which a strong occlusion is present in one of the lateral views.

4.1 Inferring scene and camera geometry

The acquisition system consists of a central reference camera C_i and a set of neighboring cameras $\mathcal{N}(C_i)$. The geometry analysis can be divided into two parts.

First, the epipolar geometry between pairs of adjacent cameras needs to be estimated, and the relative camera poses extracted. Then, following the idea of Eshel and Moses [36], we restrict the search space to a volume contained between two planes parallel to the ground plane. This amounts to detecting pedestrians with heights in a specified interval $[h_{min}, h_{max}]$. In terms of camera geometry, this requires a metric registration of each camera with respect to the ground plane in terms of variable-height homographies.

4.1.1 Relative pose estimation

A fundamental matrix $\mathbf{F}_{ij}$ between C_i and each $C_j \in \mathcal{N}(C_i)$ is estimated using the unsupervised method proposed in Chapter 2, by robust accumulation of inliers S_{ij} from a pair of synchronized video streams. From each $\mathbf{F}_{ij}$, the relative pose $(\mathbf{R}_{ij}, \mathbf{t}_{ij})$ is obtained by SVD decomposition. The SVD step introduces numerical errors which are counteracted by a first bundle adjustment (BA) optimization using the inliers S_{ij} from the pair C_i-C_j.

At this point, we also enforce a metric scale t^m_{ij} for each pair by setting the norm of $\mathbf{t}_{ij}$ to the actual distance D^l_{ij} between the cameras measured with a standard handheld laser device: $t^m_{ij} = D^l_{ij} \cdot \mathbf{t}_{ij} / \|\mathbf{t}_{ij}\|$. This simple operation is *the only manual procedure* we require in order to inject the real-world scale of the scene into the estimations.

The following mandatory step is to enforce a common metric scale to all the camera poses, as any imprecision introduced in the computation of the different $(\mathbf{R}_{ij}, \mathbf{t}_{ij})$ will have a negative impact on the data association. For any triplet C_{i-j-k}, we rely on triple matches (which are a subset of all matches identified during the fundamental matrix estimation) to propagate the scale, while including the simple matches as well in the BA in order to stabilize the other degrees of freedom of the problem. Then, if more then three cameras are used, a global BA may be applied over all the poses and available observations.

4.1.2 Variable-height homographies estimation

Let us consider the estimation of the variable-height homography between the reference camera C_i and some neighbor camera $C_j \in \mathcal{N}(C_i)$.

According to Equation (3.3), estimating the variable-height homography between two views reduces to the estimation of the homology transformation of each independent view, and of the ground plane homography. Then, according to Equation (3.2), the estimation of the vanishing points and of the metric scale factor are needed for inferring the homology.

Vanishing line and vertical vanishing point $\mathbf{v}_k$ and $\mathbf{l}_k$

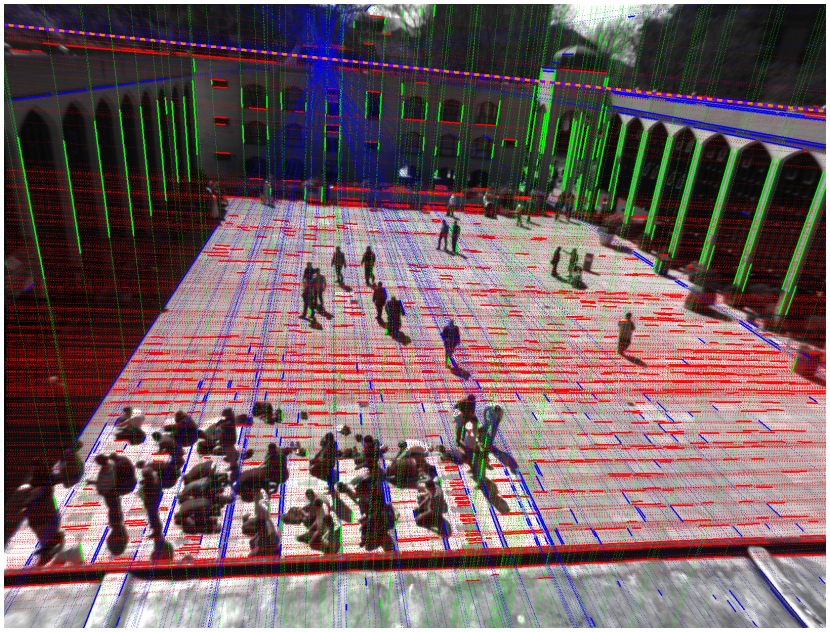


Figure 4.2: Example of vanishing points extraction on the *Regent's Park* dataset by using the method of [90]. Segments of different colors are clustered together to provide a robust estimation of a single vanishing point location. The dotted line on top is the corresponding vanishing line. Please note that the vertical vanishing point falls outside the image space.

The $\mathbf{v}_k$ and $\mathbf{l}_k$ for each camera are estimated under Manhattan world assumptions by using the method of Lezama *et al.* [90]. The underlying justification is that, although urban repetitive patterns are difficult to match reliably for inferring relative poses, they can be used in *individual* views for estimating $\mathbf{v}_k$ and $\mathbf{l}_k$.

Ground plane homography $\mathbf{H}^0_{ij}$

The estimation of the ground plane homography $\mathbf{H}^0_{ij}$ can be carried out by detecting in the two images point matches lying on the desired plane. In order to perform an automated estimation of $\mathbf{H}^0_{ij}$, we propose to extract a candidate set of point matches from the inlier set S_{ij} provided by the computation of $\mathbf{F}_{ij}$. The point extraction from video relies on the dynamics of people moving in the scene for the registration. This implies that the final 3D cloud of inliers can be clustered into points belonging to dominant planes (ground and building facades), and into points originated from pedestrian bodies moving across the scene. Given a point correspondence (p^n_i, p^n_j) in the two views, we assign to it a label $\tilde{h}^n$, corresponding to the estimated camera height under the assumption that the related 3D point is on the ground, which can be easily calculated as follows by using the epipolar geometry and the vanishing points. Only point matches located under the camera vanishing lines are considered.

Let us call $\mathbf{r}^n_i$ the straight line passing through p^n_i and $\mathbf{v}_i$ (we define $\mathbf{r}^n_j$ accordingly). As depicted in Fig. 4.4, the position of a point $\tilde{p}^n_i$ is evaluated as the intersection between $\mathbf{r}^n_i$ and the vanishing line. The point, $\tilde{p}^n_i$ corresponds to the projection of p^n_i on a plane which is parallel to the ground and at the height of the camera. In the second view, the point $\tilde{p}^n_j$ is detected as the intersection between the epipolar line $\mathbf{F}_{ij}\tilde{p}^n_j$ and $\mathbf{r}^n_j$. Let us call P^n and $\tilde{\mathrm{P}}^n$ the 3D points obtained from the triangulation of (p^n_i, p^n_j) and $(\tilde{p}^n_i, \tilde{p}^n_j)$ respectively. Then, $\tilde{h}^n = \| \mathrm{P}^n - \tilde{\mathrm{P}}^n \|$.

We turn then the ground identification step into a data clustering problem with random variable $\tilde{\mathrm{H}}^n$. We intend to separate the set of points generated by moving pedestrians on the ground (occasional visible feet locations, shadows) from the points generated by body parts in the pres-

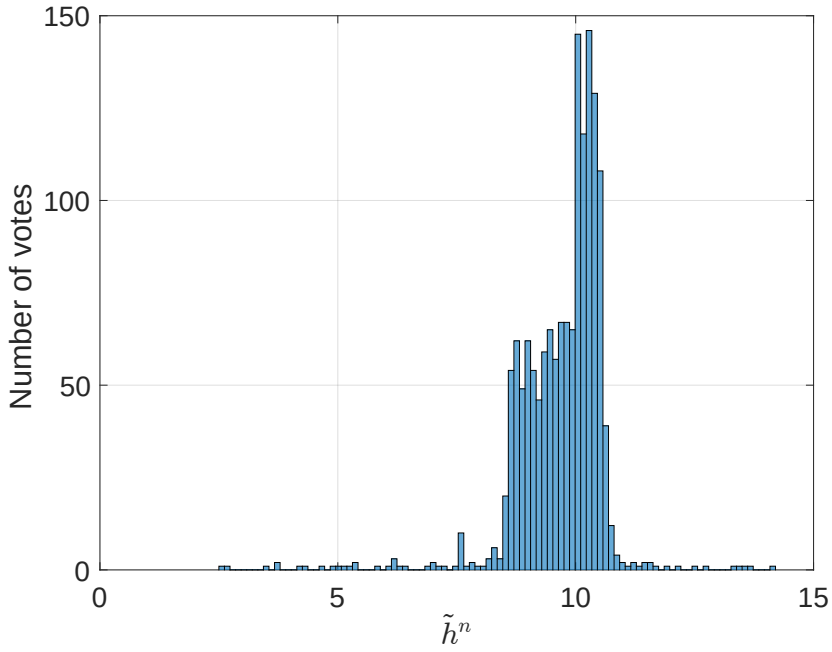


Figure 4.3: Distribution in *meters* of camera height to ground values, that is used to identify the ground location using robust EM [48].

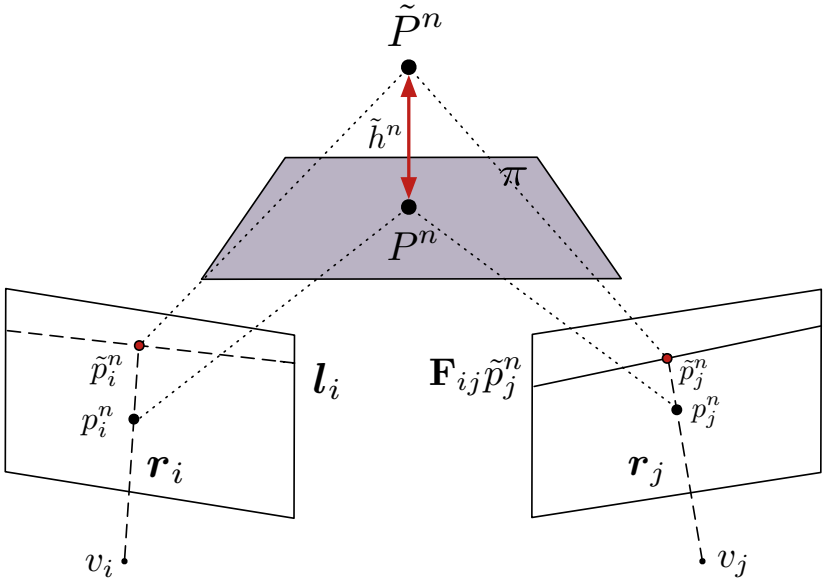


Figure 4.4: Computation of a height hypothesis $\tilde{h}^n$, starting from a point pair (p_i^n, p_j^n).

ence of other matches (i.e. building interest points), that we will define as *outliers* of the process. We perform this robust estimation task using the method of [48], which introduces an EM-algorithm robust to data outliers.

Figure 4.3 shows an example of data clustering, where two clusters are detected and the one with higher mean and lower variance is extracted as the candidate set of ground matches. As might be expected, the proximity of the distributions leads to the inclusion of false positives in the output, but the actual homography estimation is performed with a RANSAC strategy [41], therefore the presence of some outliers is well tolerated. As a numerical example, the homography estimated after running the EM-algorithm on the distribution of Figure 4.3 exhibited a RMSE=1.5 pixels with respect to a manually labeled ground truth, compared to a RMSE=1.25 pixels obtained

by selecting manually ground matches from the set S_{ij} (comparable results have been obtained for all the camera pairs).

Metric scale coefficient α_k

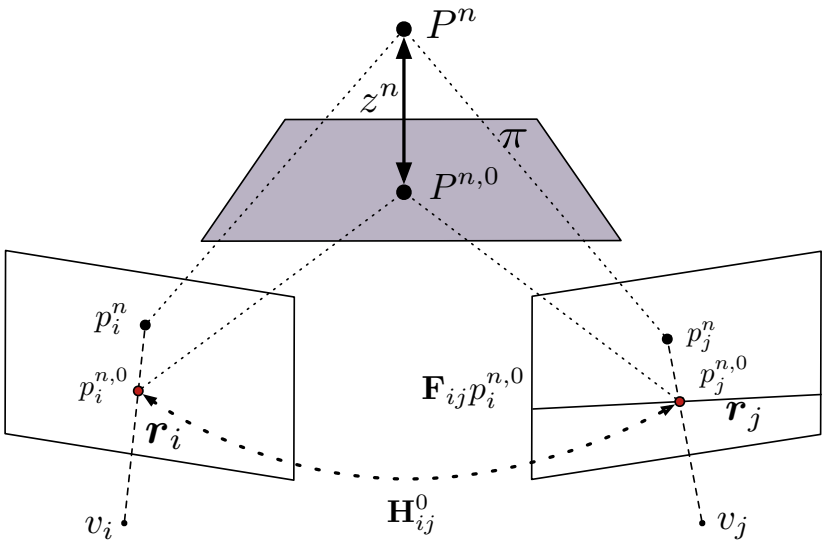


Figure 4.5: Ground projection $(p_i^{n,0}, p_j^{n,0})$ for the inlier pair (p_i^n, p_j^n).

Once $\mathbf{H}_{ij}^0$ is estimated, the metric scale coefficients α_i and α_j may be computed. According to [22], α_k (with $k = \{i, j\}$) can be derived from an image point, along with its projection on the ground plane, and with their metric distance in the real world. A convenient way of computing α_k is to take two reference points at a known distance on the scene. Our procedure for the automatic estimation of the α_k values consists in exploiting the set of inlier matches S_{ij}, in such a way that every pair of the set, which does not correspond to a point on the ground plane, votes for global candidates α_i and α_j. Given the match (p_i^n, p_j^n), the calculation of α_k requires the identification of a corresponding match of their projection on the ground $(p_i^{n,0}, p_j^{n,0})$ (see Fig. 4.5). Since the height of such point pair is unknown, the corresponding point $p_i^{n,0}$ on the ground plane lies on the segment connecting p_i^n and the vertical vanishing point $\mathbf{v_i}$. Thus, the corresponding point pair $(p_i^{n,0}, p_j^{n,0})$ must lie on $\mathbf{r}_i$ and $\mathbf{r}_j$ respectively, and satisfy both the homography and epipolar constraints.

Given the ground plane homography $\mathbf{H}_{ij}^0$, the identification can be cast as the following optimization problem:

$$\underset{p_i^{n,0}, p_j^{n,0}}{\arg\min} \left[d\left(\mathbf{H}_{ij}^0 p_i^{n,0}, p_j^{n,0}\right)^2 + d\left((\mathbf{H}_{ij}^0)^{-1} p_j^{n,0}, p_i^{n,0}\right)^2 \right] \; s.t. \; p_i^{n,0} \in \mathbf{r}_i, p_j^{n,0} = \mathbf{F}_{ij} p_i^{n,0} \cap \mathbf{r}_j \tag{4.1}$$

where $d(\cdot,\cdot)$ is the Euclidian distance operator, and $\cap$ indicates the intersection between two lines. With the given constraints the problem can be reduced to a single variable optimization problem, as the cost can be expressed as a function of one of the two components $p_i^{n,0} = (x_i^{n,0}, y_i^{n,0})$. We obtain a solution with the Levenberg-Marquardt algorithm, by choosing as starting value $y_0 = y_i^n$ of $p_i^n = (x_i^n, y_i^n)$. Then, α_i and α_j may be obtained from the point pairs $(p_i^n, p_i^{n,0})$ and $(p_j^n, p_j^{n,0})$ respectively, from equation (3.1). Once every match within S_{ij} votes for a hypothesis (α_i, α_j), we select the variable-height homography model which best fits the given data. We define a reprojection error corresponding to (p_i^n, p_j^n) as:

$$\epsilon_n = \frac{1}{2}\left(d\left(\mathbf{H}_{ij}^{z_n} p_i^n, p_j^n\right) + d\left((\mathbf{H}_{ij}^{z_n})^{-1} p_j^n, p_i^n\right)\right), \tag{4.2}$$

where $\mathbf{H}_{ij}^{z_n}$ is the homography at height z^n evaluated with the current estimation of $(\alpha_i, \alpha j)$. Finally, an LMedS estimator is applied to select the best model.

For a generic pixel p_i in the image space of camera C_i, we define $p_{j,min} \sim \mathbf{H}_{ij}^{h_{min}} p_i$ and $p_{j,max} \sim \mathbf{H}_{ij}^{h_{max}} p_i$. These two points lie on the epipolar line $\mathbf{F}_{ij} p_i$ in the image space of camera C_j, and define the extremes of an *epipolar segment* which corresponds to the search space of our stereo matching algorithm.

4.2 Pedestrian map computation

The proposed pedestrian detection method makes use of a Markov Random Field (MRF) based stereo matching. The objective is to minimize the pairwise MRF energy function:

$$E(l) = \sum_{p \in \mathscr{I}} D_p(l_p) + \lambda \sum_{(p,q) \in \mathscr{N}} V_{p,q}(l_p, l_q) \tag{4.3}$$

where: (i) p is a pixel belonging to the image $\mathscr{I}$; (ii) given the finite label set $\mathscr{L}$, l is a labeling assigning a value $l_p \in \mathscr{L}$ to each $p \in \mathscr{I}$; (iii) $\mathscr{N}$ is the set of edges of the image graph (4-connectivity is assumed); (iv) D_p is the data cost function; (v) $V_{p,q}$ is the discontinuity cost function; (vi) λ is a regularization parameter. Even if the formulation of the MRF problem is rather classic, the main contribution of our work resides on the definition of the label space in terms of height values, leading to an original formulation of the discontinuity cost function term.

4.2.1 Label definition: height-based optimization

A crucial point of our algorithm is the choice of the type of labels used. A common choice in state-of-the-art stereo matching is to use *depth* as label. Conversely, our method performs an optimization based on *height* labels. First, the choice of height is more natural if we consider how we build the search space (the volume between two planes at predefined heights). Moreover, the following task which benefits from a pedestrian detection map is the tracking on the reference plane, and, while height alone is enough to perform the ground projection (given the estimated geometry), depth information needs to be converted nonlinearly into a ground-related variable (typically height) and regularization behaviors, particularly at discontinuities, are not equivalent following the different representations.

The height label allows us to define more sophisticated constraints on local image patches (e.g. head patches) without the need of a higher-order MRF. While constant depth assumption expresses heads as planes fronto-parallel to the camera plane, the height and vertical vanishing points can be exploited to constraint the head to resemble locally a plane perpendicular to the ground.

The label set $\mathscr{L} = \{h_{min}, h_{min} + \Delta_h, \ldots, h_{max}, u\}$ is defined in the interval $[h_{min}, h_{max}]$ with a sampling step Δ_h, and is augmented by an *unknown* label u, meaning that no pedestrian is found at the specified location. We set $h_{min} = 140cm$, $h_{max} = 200cm$, $\Delta_h = 2.5cm$.

4.2.2 Data cost function

The choice of a local region descriptor for dense matching is guided by the fact that our method is supposed to work even in a wide-baseline scenario with consistent perspective distortion. We employ the DAISY descriptor from [157], which has proven to be robust to perspective and illumination changes while showing a good computational efficiency. We express the data cost between C_i and C_j as the DAISY dissimilarity [157]:

$$D_p^{i,j}(l_p) = \frac{1}{S} \sum_{k=1}^{S} \| D_i^{[k]}(p) - D_j^{[k]}(\mathbf{H}_{i,j}^{l_p} p) \| \tag{4.4}$$

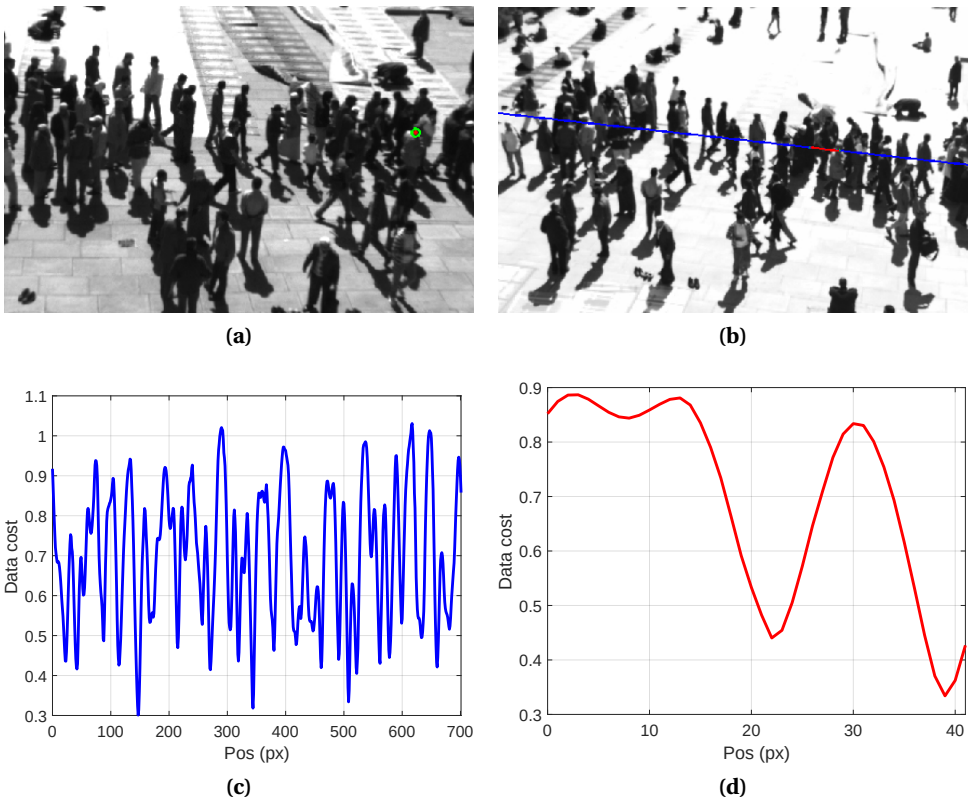


Figure 4.6: DAISY dissimilarity: a) projected pixel b) search segment corresponding to $[h_{min}, h_{max}]$ c) DAISY dissimilarity along the entire epipolar line d) DAISY dissimilarity restricted to the search segment.

where S is the number of histograms of the DAISY descriptors, $\mathrm{D}_i^{[k]}(p)$ is the *k-th* histogram evaluated at pixel p of camera C_i, $\mathrm{D}_j^{[k]}(\mathbf{H}_{i,j}^{l_p} p)$ is the *k-th* histogram of the DAISY descriptor evaluated at the projection of pixel p at an height l_p on the *epipolar segment* at camera C_j, by using the variable-height homography $\mathbf{H}_{i,j}^{l_p}$. Figure 4.6 presents the typical behaviour of the dissimilarity, and how restricting accurately the search space to the epipolar segment helps.

The special *unknown* label u is assigned with a constant data cost value $\mathrm{K}_{d,u}$. The total data cost function $\mathrm{D}_p(l_p)$ will be a combination of each $\mathrm{D}_p^{i,j}(l_p)$ computed between the reference camera C_i and any $\mathrm{C}_j \in \mathcal{N}(\mathrm{C}_i)$. Our experiments have been carried with $|\mathcal{N}| = 2$ neighbors, so a simple average of the two curves has demonstrated an effective combination, while with an increasing number of cameras an outlier robust cost merging method, like the one proposed by [168], is necessary.

4.2.3 Discontinuity cost function

By using the heights as labels we need an efficient method to estimate at each pixel location the expected local height variation. Given a point $p \in \mathcal{I}$, the direction of maximum variation of the height will be along the line $\mathbf{r}_p$ connecting p with the vertical vanishing point (see Figure 4.7). In order to provide a fast and reliable estimation of the height variation around p, we consider a small patch area, with the length of an average head L, centered in p as a planar surface. The direction of maximum height variation favors an orientation of the corresponding 3D patch which is perpendicular to the ground plane. We define as $|\nabla_p|$ the estimated absolute value of the height variation along $\mathbf{r}_p$ at a unit distance from p. Given the planar locality assumption, this quantity is expressed as:

$$|\nabla_p| = \frac{\mathrm{L}}{d(p, p^{\mathrm{L}})} \tag{4.5}$$

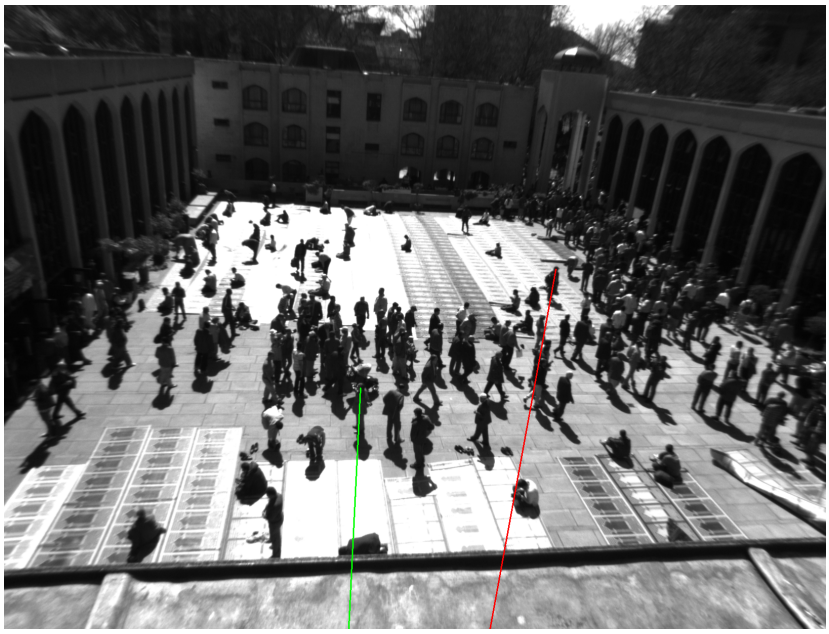


Figure 4.7: Example of lines connecting a given pixel to the vertical vanishing point, in order to estimate the maximum height variation of a pedestrian. The difference in the expected gradient $|\nabla_p|$ can be appreciated: the red head has a radius of 4px and $|\nabla_p| = 2.5cm$; the green head has a radius of 6p and $|\nabla_p| = 1.6cm$.

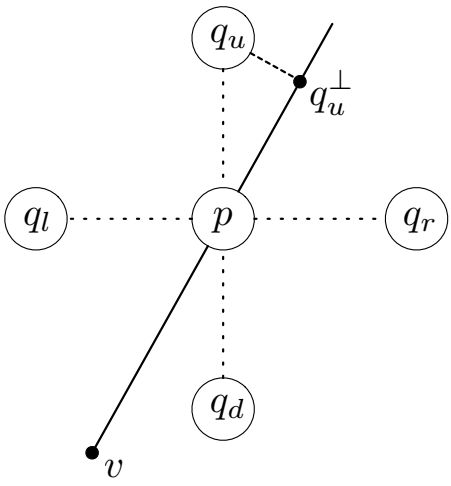


Figure 4.8: Neighborhood definition for the calculation of the discontinuity cost function. Expected gradient $|\nabla_p|$ between to neighboring pixels is re-scaled proportionally to the distance between the point and the projection of its neighbor on the line connecting it to the vertical vanishing point.

where p^{L} is the image projection of pixel p into the 3D parallel plane located at distance L from the 3D plane determined by p. The point p^{L} is evaluated using homologies as: $p^{\mathrm{L}} = \mathbf{B}^{\overline{h}-\mathrm{L}}\left(\mathbf{B}^{\overline{h}}\right)^{-1} p$, where $\overline{h}$ is the central value of the height interval $[h_{min}, h_{max}]$. The choice of a constant $\overline{h}$ value is justified by the negligibly small variation of $|\nabla_p|$ for different h values with respect to Δ_h and for a given p, leading to no effect on the final evaluation of the discontinuity function. As a consequence, Equation (4.5) allows us to estimate the $|\nabla_p|$ map once during the algorithm initialization step.

Let us consider the neighbor points $p = (x_p, y_p), q = (x_q, y_q) \in \mathscr{I}$. The expected height variation between p and q is proportional to $|\nabla_p|$ and to the projection of q on the line $\mathbf{r}_p$. The point $q^{\perp} = (x_{q^{\perp}}, y_{q^{\perp}})$ represents the orthogonal projection of q on $\mathbf{r}_p$ (see Figure 4.8). In order for this

projection to be valid, we use the hypothesis that for neighboring pixel in the image space, the angles with the u axis of the u-v image reference system of the lines $\mathbf{r}_p$ and $\mathbf{r}_q$ are almost identical: $\theta_{r_p} \approx \theta_{r_q}$. We can define the following distance function:

$$\mathrm{D}_{p,q}(l_p, l_q) = \left| l_p - l_q - s_p |\nabla_p| d(p, q^{\perp}) \right| \tag{4.6}$$

where $s_p = 1$ if $y_p < y_{q^{\perp}}$ and $s_p = -1$ otherwise, meaning that the height value has to decrease when moving to lower pixels in the image space. Please note that since $\theta_{r_p} \approx \theta_{r_q}$ and $|\nabla_p| \approx |\nabla_q|$, it follows that $\mathrm{D}_{p,q}(l_p, l_q) \approx \mathrm{D}_{q,p}(l_q, l_p)$, but the equality does not hold strictly numerically (the maximum difference observed during experiments is of the order of 10^{-3}). We enforce a symmetrical message flow by evaluating for each pair p, q the distance functions in both directions, and by considering the one which provides the highest error. The discontinuity function between two labels which are not both unknown is defined as a truncated distance:

$$\hat{\mathrm{V}}_{p,q}(l_p, l_q) = \min \left[\frac{\max\left(\mathrm{D}_{p,q}(l_p, l_q), \mathrm{D}_{q,p}(l_q, l_p)\right)}{\Delta_h}, \mathrm{K} \right] \tag{4.7}$$

The total discontinuity cost function is expressed as:

$$\mathrm{V}_{p,q}(l_p, l_q) = \begin{cases} \hat{\mathrm{V}}_{p,q}(l_p, l_q) & (l_p \neq u) \wedge (l_q \neq u) \\ \mathrm{K}_{\mathrm{V},u} & (l_p = u \wedge l_q \neq u) \vee (l_p \neq u \wedge l_q = u) \\ 0 & (l_p = u) \wedge (l_q = u) \end{cases} \tag{4.8}$$

where $\mathrm{K}_{\mathrm{V},u}$ is a constant discontinuity cost for *unknown* labels (in the experiments $\mathrm{K}_{\mathrm{V},u} = \mathrm{K}$ for convenience).

The proposed pairwise discontinuity cost does not satisfy the submodularity property, therefore an alpha-expansion graph cut algorithm is not applicable to the optimization process. We provide results of the MRF optimization with Loopy Belief Propagation [179], but other techniques such as tree-reweighted message passing [83, 152] may be used as well.

4.2.4 Temporal filtering

In order to illustrate qualitatively and quantitatively the interest of our work, we perform a simple temporal filtering of the detection results. The temporal filtering step outputs motion consistent trajectory fragments, denoted as tracklets. A more involved approach for the validation of instantaneous detections would require the use of appearance information either in the optimization or in a subsequent tracking algorithm, but these extensions go beyond the scope of the present work.

The temporal filtering is applied as follows:

1 The height map points are projected on the reference plane.

2 Local maxima (in terms of height) are identified in the projected data.

3 Each point is clustered with respect to the closest local maximum, and a centroid is computed for each cluster.

4 Tracklet creation and extension: each centroid may either be associated to an existing tracklet if it is located closer than θ_d from the linearly predicted tracklet location, and if its height is closer than θ_h to the tracklet average height; otherwise, a new tracklet is created. For all our experiments we used $\theta_d = 20cm$ and $\theta_h = 15cm$.

5 Any tracklet which is not extended is terminated.

At the end of the temporal filtering part, tracklets of length equal or less than a tracklet thresholding parameter θ_l are discarded. For our experiments we considered values of $\theta_l \in \{0, \ldots, 3\}$, with 0 meaning no filtering.

4.3 Experiments

We evaluate the proposed algorithm on the crowded scene *Regent's Park*, used initially for wide baseline relative pose estimation. The laser measured distances between the central camera and

λ value	Recall	Precision
0.08	89.89%	42.65%
0.15	74.37%	66.78%
0.20	62.23%	82.82%

Table 4.1: Recall and precision on the *Sparse* sequence depending on the regularization parameter λ. Here $\theta_l = 1$.

θ_l value	Recall	Precision
0	85.60%	65.67%
1	80.20%	72.35%
2	75.10%	77.18%
3	70.26%	79.81%

Table 4.2: Recall and precision on the *Dense* sequence depending on the tracklet threshold θ_l.

(a) left camera **(b)** central camera **(c)** right camera

Figure 4.9: Regents Park Mosque dataset, *Dense* sequence.

the other two are 9.35 and 10.1 m. We present the results obtained in the central part of the scene which corresponds to the overlapping area of the three cameras, and which has roughly 400 m^2. For clarity, we recall that the related works on unsupervised detection of Eshel and Moses [36] and Khan and Shah [79] are not suitable for comparison in such kind of scenes. The method in [36] is based on scene constraints which are not transposable in a wide baseline open environment; the work in [79] tracks feet locations, and this operation is not applicable in higher density scenes as the one proposed.

In order to highlight better the specific behavior of our method, we process two different manually annotated sequences, the first one denoted as *Sparse* containing 200 frames and 2969 manually annotated pedestrians, and a second sequence denoted as *Dense* containing 500 frames and 18567 annotated pedestrians, most of them being clustered in a transit zone (Fig. 4.9). The ground truth annotations are used to evaluate the overall object level precision and the recall of our method in each sequence.

4.3.1 Impact of the tracklet threshold θ_l

The parameter θ_l controls in a simple manner the geometric consistency of the tracklet and is quite effective in removing spurious trajectories, which are uncommon in dense crowds. In Table (**??**) we present the results in terms of precision and recall obtained on the *Dense* sequence, using values of $\theta_l \in \{0,\ldots,3\}$. The regularization parameter is set to $\lambda = 0.07$. Beyond these lengths, the majority of tracklets are correct and the thresholding becomes detrimental. Figure 4.10 presents the result of the detection on frame 5 of the *Dense* sequence.

4.3.2 Impact of the regularization parameter λ

The parameter λ of Equation (4.3) has a significant impact on the results as it controls the enforcement of the height gradient constraint with respect to the photometric term. In Table 4.1 we present the results in terms of precision and recall obtained on the *Sparse* sequence, using different values of λ which cover the entire effective range. The tracklet threshold is set to $\theta_l = 1$. The impact of λ is even more marked in a sparse setting, as the geometry constraint allows for removing the false positives created by the ground surface, in the absence of an object appearance model. Figures 4.11a-4.11c present the detection result on frame 27, with increasing values of λ.

Figure 4.10: Detections on the *Dense* sequence, prior to temporal filtering.

(a) $\lambda = 0.08$ **(b)** $\lambda = 0.15$ **(c)** $\lambda = 0.2$

Figure 4.11: Detections *Sparse* sequence with varying levels of regularization. The tracklet threshold is set to $\theta_l = 1$.

4.3.3 Final discussion of results and failure cases

The algorithm we propose clearly shows an excellent performance on strongly occluded crowd images, despite the use of only three views and the absence of appearance related terms. Besides, the influence of the parameters θ_l and λ on the results may be exploited in order to favor precision or recall. Finally, due to the absence of appearance information in our framework, some expected difficulties arise. At low densities, ground areas may be visible and any ground related phantom will be persistent, and impossible to differentiate from a still individual (Fig. 4.11a) - we highlight this behaviour in Table 4.1, while it also worth noting that even in *Dense* the precision in Table 4.2 is impacted by some occasional frames with lower densities. A second failure case arises whenever in a detection blob corresponding to multiple pedestrians, there is only one local height maximum. In some cases, objects which are not heads and which are raised at above-torso level are correctly detected as being located at a valid height from the ground. We expect these cases to be easily corrected by taking into account an appearance based model. Since we address the detection problem as an energy minimization, one straightforward way to extend our work is to include an additional term related to appearance in the data cost function. Alternatively, a standard tracking algorithm including appearance cues would address these cases as well.

4.4 GPU acceleration of pedestrian map computation

The following section introduces a body of work which has been performed in order to tackle a more practical aspect of the aforementioned algorithm, mainly the running time. Although multiple approaches are available in order to solve the minimization introduced initially in Equation 4.3, one of the reasons for relying on the LBP algorithm has been also its potential for high parallelization.

4.4.1 Overview of the GPU architecture

In recent years, general purpose processing on GPU (Graphics Processing Unit) provided significant support for many scientific fields in which efficiency and speed is a vital factor. Specifically, NVIDIA CUDA framework has enabled us to rely on a GPU parallel environment with greater ease. The authors of [4] describe the structure of NVIDIA GPUs in two levels: each GPU chip consists of streaming multiprocessors (SM), which have their own cores (Figure 4.12). CUDA uses the term 'grid of thread blocks', where multiple blocks map onto multiple SMs and multiple threads map onto the cores. Each thread block contains several threads. Each thread has its own local memory while it can also communicate with other threads inside the block via shared memory. Every thread has also access to a bigger and slower global memory. GPU groups threads together in *execution warps*. Threads inside a warp execute concurrently. The size of warps depends on the device being used (in our case it equals to 32). Generally, it is not necessary to follow the exact hardware specifications when utilizing CUDA; we can use more threads than cores and leave the scheduling to the hardware. Having these details in mind, the next section will describe our approach to the GPU implementation.

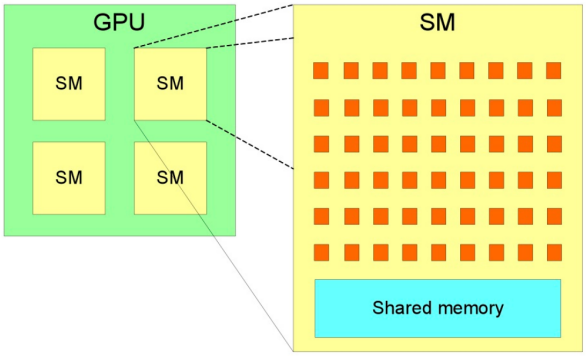


Figure 4.12: GPU architecture hierarchy (from [4])

4.4.2 Basic optimizations

To optimize the pedestrian detection algorithm, we started with basic changes from a mostly serial CPU implementation to a highly parallel GPU code. The next three sections iterate over the main decisions taken in order to make the translation as efficient as possible.

Thread mapping

The first step to translate a serial algorithm to a parallel paradigm is to assign the responsibility of each processing unit (in our case GPU threads). In our problem, this means to specify how the grid of threads is related to our computation grids. As our data come from 2D images, if we decide on different dimensions for the thread blocks we need a proper translation between the two. In the remainder of this section N will denote the number of pixels, M the number of messages for each pixel and V number of labels.

First we choose the responsibility of each thread block.

One pixel per thread block: In this configuration, we will have N thread blocks either in 1D fashion or similar to the image in 2D with a (Width,Height) matrix of thread blocks. Each block tends to messages originating from only one pixel. So each thread depending on its block index tends to a different pixel.

Multiple pixels per thread block: This way, we assign more than one pixel for each thread block in hope of doing more work in parallel. Each thread, depending on its block index and also its thread idx (for example third dimension of the thread index) knows which pixel to access.

In practice, putting computation of more than one pixel in a block will not give us any improvement. In our tested GPU we had the limit of 32 active blocks and 64 active warps per SM. This means with just 2 active warps per thread block we can theoretically achieve 100% occupancy for the GPU. In our case, for computing four messages for one pixel we need four active warps (assuming warp size is 32 and V < 32). Therefore, there is no actual need for more active warps per thread block.

We also have to arrange threads inside a block.

Each block containing M*V threads: Each thread is in charge of calculating the message for one neighbor regarding one particular label. Whether the formation is in one dimension or two will not affect the performance but structuring the block as M vectors of length V helps the programming process.

Each block containing M*32 threads: This means instead of using the number of labels as the width of the block we use the nearest power of two greater than V. This way we make sure each computation warp only deals with messages related two one neighbor.

As mentioned in [57], GPU architecture follows a Single Instruction Multiple Data (SIMD) execution model, which is not suited for kernels with divergent execution flow. Thus, as a general rule we need to make sure threads in the same warp follow the same (or similar) path. In our case, there is significant divergence between code execution paths for different messages. Therefore, in order to keep the divergence minimal in each warp, we chose the dimensions of the blocks to be M*32 (assuming the warp size is 32), limiting each warp to one message.

After deciding on how to map to the GPU threads, the conversion to parallel code is straightforward, as loop indexes are replaced by thread and block indexes. In the next two sections we describe in more details two major parts of the parallel code.

Parallel reduction

Part of the message passing process used in this algorithm is to compute minimum and sum of messages. Both of these fall under the family of reduction vector operations (i.e. deriving one value from a K-sized vector). Therefore, to parallelize these operations we can follow the same procedure. To choose the best option, we considered two conventional approaches to a parallel reduction mentioned in [60] and [101]. The first approach uses each thread block shared memory to reduce at each step two elements of the vector. This means we will need a vector of size V in shared memory for each reduction. The reduction algorithm made possible on new GPUs [101], uses a process called shuffling to communicate a variable between threads inside a block. This way there is no need for shared memory and synchronized access. After testing in our case, which we uses around 400k blocks and 25 labels, shuffling has been more effective as expected, although the difference is not considerable but significant.

Computing the discontinuity cost function

During each LBP iteration, considerable time is dedicated to the calculation of the discontinuity cost function. This part includes many summations and multiplications which are the same at each iteration. Therefore it is a good idea to pre-compute this part and exclude it from the runtime and turn it into a single memory look up. Each run of the function needs six arguments: two pairs of pixel locations and two labels. Since the locations are always neighbors, we can reduce

the input to one pair for a pixel location and another argument indicating the direction of the neighbor. This means for completely pre-computing this function we will need a five dimensional matrix. The size S of the matrix will be equal to:

$$S = HEIGHT * WIDTH * 4 * V^2 \tag{4.9}$$

Where HEIGHT and WIDTH are the input image dimensions. Complete pre-computation of the function led to speed up in runtime of each iteration; but at the same time cost more than 4GB of graphics memory, for 400k pixels and a label set of 25. This could prove problematic if we decide to increase the label-set or use larger images. Therefore another approach was used to divide the function in two parts: one was to be computed beforehand and one to be calculated at every iteration. The part of the function related to the geometry (height gradient multiplied by neighbor pixel projection) is precomputed and stored in a matrix with a size proportional to the size of the image. The computation depending on the labels proved to be very simple and manageable in each iteration. This way we decreased the amount of required memory to around 1.5GB which is a fairly reasonable usage. This new approach also decreased the number of global memory accesses. Before we had V^2 global memory access per message. Now we have only one per message. This gave us an overall improvement on iteration runtime.

4.4.3 Further optimization

In this part, we will briefly iterate over some further optimizations done in order to run the algorithm as efficiently as possible. Some of these points are general best practices which will work on any parallel GPU program, but in some cases the changes make sense only in the context of our problem.

Memory optimization

In the process of minimizing the mentioned energy function, there are three major group of stored data: pre-computed data cost function, partially pre-computed discontinuity cost function and previously calculated messages. All these sources are needed for computing new messages, therefore in each iteration there are many memory reads from the GPU global memory. Also, in the final step of the computation we need to store the messages in the global memory again to be used in the next iteration. This makes a calculated memory access scheme vital for the algorithm to run as fast as possible. We will mention two general important points to achieve a better access time.
Benefiting from memory hierarchy: In some cases we need to use the same data more than once. The data cost function in our case is frequently needed. In these cases, it is not advisable to access the global memory each time. The best choice is to load the data once and store it in a faster memory, either shared memory or each thread's local memory. The choice between the two depends how much of each memory we have available to use. Overuse of local memory can lead to using more registers which can reduce the overall occupancy of GPU. For example, in our tested GPU each block was limited to 65536 registers and 98KB of shared memory.
Coalesced access: [59] considers coalescing memory operations as one of the general optimization directives for a parallel program which can lead up to 10x speedup. In simple words, this means consecutive threads access consecutive memory addresses (among other conditions). When accessing incoming messages to each pixel it is not possible to maintain a coalesced access but in other cases maintaining such access gave us considerable improvement.

Instruction optimization

After optimizing the memory operations, the bottleneck of the kernel falls on too many mathematical instructions. Mostly in the discontinuity cost function, floating point division and multiplication caused a considerable delay. One possible solution to alleviate this problem is to sacrifice accuracy for more speed by using the fast mathematical instructions of the GPU. Actual usefulness

of this approach obviously varies case by case. In our algorithm, considerable trial and error with these functions led to use of __fdividef, __fmul_rd and __fsub_rd instead of regular division, multiplication and subtraction in limited cases. This gave us reasonable imprecision which did not affect the end result while saving significant time.

Algorithm optimization

The last step in our optimization process was to investigate whether or not any change in the core algorithm can be helpful. Following the advise of [39], we decided to alternatively calculate only half of the messages in each iteration. Basically, if we divide the pixels into two subsets A and B following a checkerboard pattern, in each iteration we only compute the outgoing messages of pixels belonging to either A or B. This makes sense because the messages sent from nodes in A only depend on outgoing messages of nodes in B. The same can be said about calculation of messages from nodes in B. [39] also describes the new message from node p to q at iteration t as: if t is odd (even) then

$$\bar{m}^t_{p\to q} = \begin{cases} m^t_{p\to q} & if\ p \in \text{A}\ (if\ p \in \text{B}) \\ m^{t-1}_{p\to q} & otherwise \end{cases} \tag{4.10}$$

This means the new messages are almost the same as the standard ones, and regarding convergence this method also converges to the same fixed point i.e. after convergence $m^{t-1}_{p\to q} = m^t_{p\to q}$ While being an approximation, this change practically did not affect the end result of the algorithm but it managed to decrase the iteration time by half.

4.4.4 Results

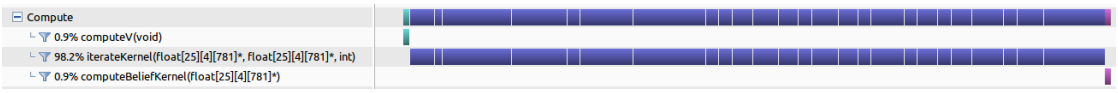


Figure 4.13: Overall time-line of the execution of belief propagation for one frame with 100 iterations

In this section, we will report the recorded execution time of the algorithm using real captured data on a selected platform. For testing, we have used a system with NVIDIA Geforce GTX1080 graphic card and 8GB of graphical memory. The system is also equipped with an Intel® Core™ i7-6900K CPU with 3.20GHz processing speed. The iteration time was measured using NVIDIA's own profiler *nvprof*. The images for testing are the ones from the *Dense* dataset, which are by default cropped down to the size of 781*621 which will give us around 500K pixels to work with. As mentioned before, the algorithm is tested with a label set of size 25; from 1.4 meters to 1.975 with inclusion of a special label for pixels without heads. The algorithm consists of three main kernels - one for pre-computation, one for each iteration of loopy belief propagation, and one for final belief computation. The first and last kernel will run once for each image while the second one will run depending on the number of iteration needed for convergence, in this case 100. Figure 4.13, shows the general time-line of the program. As we can see, about 98% of the runtime is occupied by the iteration kernel, which makes it the most important kernel to optimize. Figure 4.3 shows the execution time of these kernels using ten consecutive frames.

Since more than 98% of execution time belongs to the iteration kernel, we will conduct our comparison with CPU code only for this kernel. Generally we achieved over 3000x speed up for each iteration. As said earlier, the process of getting to this level of efficiency involved several general purpose optimization as well as detailed and specific to our problem changes. Table 4.4 shows the progression of iteration time for a sample frame in different version of the algorithm; from initial serial code to the current most optimized state.

It was crucial to make sure no part of the optimization introduces any intolerable deviation from the original output. Inherently, CPU and GPU codes give us slightly different floating point operations which can cause differences in final labels. On top of that, as covered in Section 4.4.3,

Frame	precomputation	Average iteration	Belief computation
1	3.854	3.781	3.201
2	4.268	4.267	5.415
3	3.66	4.147	4.016
4	3.788	3.799	3.054
5	3.899	4.401	3.983
6	3.936	4.032	3.203
7	3.886	3.934	3.704
8	3.79	3.8	3.235
9	5.031	4.262	3.116
10	4.284	4.101	3.094
average	4.0396	4.0524	3.6021

Table 4.3: Time taken by the three main kernels in 10 consecutive frames. Times are in millisecond

Version	Iteration time
CPU code	~11s
Naive GPU implementation	~1s
Using initial pre-computation	~0.25s
Memory optimizations	~0.05s
Improved pre-computation	~0.02s
Using fast instruction approximation	~0.01s
Message passing approximation	~0.004s

Table 4.4: Step by step optimizations and their time in seconds

we have used two approximations namely the use of fast arithmetic operations and computing alternatively only half of the outgoing messages. Depending on each frame these changes cause some misidentification and misjudgment of height labels. A comparison of both versions of the algorithm was done on a set of 15 consecutive frames. Table 4.5 shows the result of this comparison. Three metrics are considered: the number of pixels in which both algorithms detect a head but disagree on the height (second column); the number of pixels in which the two implementations disagree on the presence of a head (third column), and finally average of absolute difference for the pixels which disagree on the height. As we can see, considering the total number of pixels, very few misidentifications happen and also the height difference stays around 2.5cm which is the elementary height increment.

Figure 4.14 shows the overall execution time for each frame in both CPU and GPU code. These times include the reading of precomputed data cost function from disk. Overall, a solid 1100x speedup means we can process a long video sequence which used to take several days, in a matter of minutes.

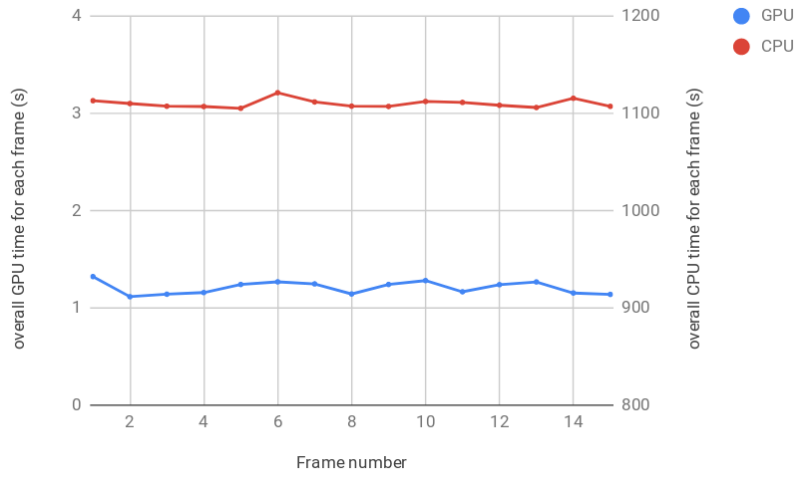


Figure 4.14: Comparison between the CPU and GPU code in total execution time for each frame

Frame	Number of pixels with wrong height	Number of pixels with wrong identification of head presence	average absolute difference of height (meter)
1	18	5	0.025
2	44	62	0.025
3	41	12	0.025
4	34	11	0.025
5	31	5	0.025
6	2	3	0.025
7	47	112	0.025
8	14	9	0.025
9	15	3	0.025
10	3	2	0.025
11	41	32	0.032
12	60	15	0.046
13	43	10	0.04
14	50	15	0.027
15	18	1	0.038

Table 4.5: Result of comparing the output of optimized code with the original CPU code. Window size is 781*621 (485001 pixels).

4.5 Conclusion

We proposed a method for locating pedestrian occupancy in a cluttered outdoor scene using a multiple camera network under realistic conditions of size, illumination and pose variations. Moreover, we have demonstrate that the algorithm can achieve competitive performance with a minimal number of three cameras. We have iterated over several steps in order to provide an optimized parallel implementation of the algorithm, which can run efficiently on GPU, by providing over 1000x speedup while preserving over 99.9% accuracy.

With this advancement, our approach may be complemented using different appearance based terms, both for low level (embed appearance knowledge in the data cost) or high level (perform detection fusion from multiple sensors) data fusion. For the following, we will focus our attention to the high level fusion task. The output of such iteration of our work is a set of pedestrian detections on the ground plane. A robust and efficient data fusion approach has to be proposed for two-dimensional spaces, both for spatial (multiple sensors, e.g. supervised detectors) and temporal (e.g. tracking) information fusion.

Part III

Information Fusion in Two Dimensional Spaces

Chapter 5

Belief Function Theory

Contents

The belief functions theory (BFT), also known as Dempster-Shafer theory [140], represents a generalization of probability theory. Shafer theory [105], evidence theory or the transferable belief model [109], are variants of the same idea of definying a framework for modeling partial knowledge and uncertain information. It is also a generalization of possibility theory [181] and it has a direct relationship with several other theories, like random sets [117] and imprecise probabilities [169].

In this chapter we will present the notions and operators useful for our study.

5.1 Belief representation

5.1.1 Mass function

Let us denote by Ω the *discernment frame*, i.e. the set of mutually exclusive hypotheses representing the possible solutions. The power set 2^Ω is the set of the Ω subsets, i.e. the disjunctions of the set of singleton hypotheses in Ω, having cardinality $2^{|\Omega|}$.

Definition 5.1.1. *A **mass function**, specifying a basic belief assignment (BBA), is a function over the power set 2^Ω, $m^\Omega : 2^\Omega \rightarrow [0,1]$, which satisfies:*

$$\sum_{A \subseteq \Omega} m^\Omega(A) = 1. \tag{5.1}$$

The superscript indicating the discernment frame Ω can be omitted when there is no ambiguity about such space. Every set $A \subseteq \Omega$ such that $m(A) > 0$ is called ***focal element***, or *focal set*, of m.

Definition 5.1.2. *A mass function is said to be **normalized** if:*

$$m(\emptyset) = 0,$$

where the empty set $\emptyset$ is the null hypothesis of the given discernment frame. In the case of *unnormalized* BBAs, the mass on the empty set can be interpreted as the degree of support of the hypothesis that the solution lies outside Ω. Usually normalization property is applied under **closed-world assumption**, i.e. Ω is defined on an exhaustive set of hypotheses.

Definition 5.1.3. *A mass function is said to be **vacuous** if it has Ω as unique focal element, i.e. $m(\Omega) = 1$.*

A vacuous mass function, under closed-world assumption, conveys total ignorance as information piece.

Definition 5.1.4. *A non-vacuous mass function that has only one focal element is said to be **categorical**:*

$$\exists!\ \mathrm{A} \subset \Omega \ s.t.\ m(\mathrm{A}) = 1 \tag{5.2}$$

A categorical mass function conveys certain information, but possibly imprecise.

Definition 5.1.5. *A mass function is said to be **Bayesian** if it has only singleton hypotheses as focal elements:*

$$\forall\ \mathrm{A} \subseteq \Omega \ s.t.\ m(\mathrm{A}) > 0 \Rightarrow |\mathrm{A}| = 1. \tag{5.3}$$

Thus, probability distributions boil down to a particular case of a mass function, when precise information has to be modeled.

Definition 5.1.6. *A mass function is said to be **dogmatic** if Ω is not a focal element, i.e. $m(\Omega) = 0$.*

Definition 5.1.7. *A mass function A^w, $w \in [0,1]$, is said to be **simple** if it has at most two focal elements, including Ω, such that:*

$$m(\mathrm{A}) = 1 - w, \quad m(\Omega) = w.$$

A simple mass function A^0 is categorical, while A^1 is vacuous. For practical usage (e.g. canonical decomposition) only non-dogmatic BBAs are considered, so that $w \in (0,1]$.

5.1.2 Alternative representations

The information encoded by a mass function can be represented in different equivalent formulations, most notably belief (*Bel*), plausibility (*Pl*) and commonality (q).

Definition 5.1.8. *Belief, plausibility and commonality are defined, respectively, as:*

$$\mathrm{B}el(\mathrm{A}) = \sum_{\mathrm{B} \subseteq \mathrm{A}} m(\mathrm{B}), \quad \mathrm{P}l(\mathrm{A}) = \sum_{\mathrm{B} \cap \mathrm{A} \neq \emptyset} m(\mathrm{B}), \quad q(\mathrm{A}) = \sum_{\mathrm{B} \supseteq \mathrm{A}} m(\mathrm{B}), \tag{5.4}$$

for all $\mathrm{A} \subseteq \Omega$.

The belief of A can be interpreted as the degree to which the evidence induces A.

The plausibility of A can be thought as the degree with which A is consistence with the evidence. Equivalently, $\mathrm{P}l(\mathrm{A}) = \mathrm{P}l(\Omega) - \mathrm{B}el(\overline{\mathrm{A}})$ measures the lack of support of the complement of A. From the definition $\mathrm{P}l(\mathrm{A}) \geq \mathrm{B}el(\mathrm{A}), \forall \mathrm{A} \subseteq \Omega$. Please note that in case of a Bayesian mass function the equality holds, and the two representations are equivalent to a probability function.

The commonality of A has mainly a computational role as explained further.

The three representations are equivalent to the mass function formulation because there exists a one-to-one mapping between any two of those. For example, under closed-assumption, the normalized mass function can be computed as:

$$m(\mathrm{A}) = \sum_{\mathrm{B}\subseteq\mathrm{A}} (-1)^{|\mathrm{A}\setminus\mathrm{B}|} \mathrm{B}el(\mathrm{B})$$

$$m(\mathrm{A}) = \sum_{\mathrm{B}\subseteq\mathrm{A}} (-1)^{|\mathrm{A}\setminus\mathrm{B}|} (1 - \mathrm{P}l(\overline{\mathrm{B}}))$$

$$m(\mathrm{A}) = \sum_{\mathrm{B}\supseteq\mathrm{A}} (-1)^{|\mathrm{A}\setminus\mathrm{B}|} q(\mathrm{B})$$

5.1.3 Consonant belief functions

Definition 5.1.9. *A mass function is said to be **consonant** if the focal elements are nested:*

$$\forall\, (\mathrm{A},\mathrm{B}) \in 2^{\Omega} \times 2^{\Omega}\, m(\mathrm{A}) > 0, m(\mathrm{B}) > 0 \Rightarrow \mathrm{A}\subseteq\mathrm{B} \vee \mathrm{B}\subseteq\mathrm{A}. \tag{5.5}$$

For consonant mass functions the following relation hold:

$$\mathrm{P}l(\mathrm{A}\cup\mathrm{B}) = max(\mathrm{P}l(\mathrm{A}), \mathrm{P}l(\mathrm{B})), \forall \mathrm{A},\mathrm{B}\subseteq\Omega \tag{5.6}$$

The plausibility of a consonant mass function is a possibility measure, defining a possibility distribution [181]. The possibility distribution is the contour function:

$$\pi(x) = \mathrm{P}l(\{x\}), \quad \forall x \in \Omega \tag{5.7}$$

Thus, the theory of belief functions can be considered as a generalization of the possibility theory.

5.1.4 Discounting

In the BFT, discounting can be used in order to model the reliability of some source of information [140]. Among the different discounting ways proposed (e.g. contextual discounting [109], contextual reinforcement [125], etc.), the simplest uses a discounting factor in order to reallocate some mass to the ignorance state.

Definition 5.1.10. *Given a discounting factor* $\alpha \in [0,1]$*, the discounted mass function* ${}^{\alpha}m$ *is defined as:*

$$\begin{aligned} {}^{\alpha}m(\mathrm{A}) &= \alpha m(\mathrm{A}), \quad \forall \mathrm{A}\subset\Omega \\ {}^{\alpha}m(\Omega) &= \alpha m(\Omega) + 1 - \alpha \end{aligned} \tag{5.8}$$

When $\alpha = 0$, the information is considered not reliable, and the corresponding mass function is vacuous; when $\alpha = 1$, the information is considered reliable and the mass function is kept as it is.

5.1.5 Information commitment

In the theory of belief functions, commitment expresses how much informative a mass function is. One can define an ordering of commitment.

Definition 5.1.11. *Given two mass functions* m_1 *and* m_2*,* m_1 *is **q-more committed** than* m_2*,* $m_1 \sqsubseteq_q m_2$*, if:*

$$q_1(\mathrm{A}) \leq q_2(\mathrm{A}), \quad \forall \mathrm{A}\subseteq\Omega. \tag{5.9}$$

Definition 5.1.12 (Least Commitment Principle [145])**.** *Given several belief functions compatible with a set of constraints, the least informative (committed) according to some informational ordering (if it exists) should be selected.*

5.2 Belief function combination

5.2.1 Conjunctive combination

Definition 5.2.1. *Given two mass functions m_1 and m_2 induced by two independent items of evidence, the conjunctive rule (unnormalized Dempster's rule) is defined as:*

$$m_1 \textcircled{\cap} m_2(\mathrm{A}) = \sum_{\mathrm{B}\cap\mathrm{C}=\mathrm{A}} m_1(\mathrm{B})\, m_2(\mathrm{C}), \quad \forall \mathrm{A} \in 2^{\Omega}. \tag{5.10}$$

Let us define the *degree of conflict* k between two mass functions as:

$$k = \sum_{\mathrm{B}\cap\mathrm{C}=\emptyset} m_1(\mathrm{B})\, m_2(\mathrm{C}). \tag{5.11}$$

The normalized version of the Dempster's rule is then defined as:

$$\begin{aligned} &m_1 \oplus m_2(\mathrm{A}) = \frac{1}{1-k} \sum_{\mathrm{B}\cap\mathrm{C}=\mathrm{A}} m_1(\mathrm{B})\, m_2(\mathrm{C}), \forall \mathrm{A} \in 2^{\Omega} \setminus \{\emptyset\}, \\ &m_1 \oplus m_2(\emptyset) = 0. \end{aligned} \tag{5.12}$$

The conjunctive rule has a simple definition in terms of commonalities:

$$q_1 \textcircled{\cap} q_2(\mathrm{A}) = q_1(\mathrm{A})\, q_2(\mathrm{A}), \quad \forall \mathrm{A} \in 2^{\Omega}. \tag{5.13}$$

The Dempster's rule has properties of commutativity and associativity, and, when applied to Bayesian mass functions, it is equivalent to the probabilistic product rule. Dempster's rule, like its unnormalized version (called the conjunctive rule from now on), assume that both pieces of evidence are reliable.

5.2.2 Disjunctive combination

Definition 5.2.2. *Given two mass functions m_1 and m_2 induced by two independent items of evidence, the disjunctive rule is defined as:*

$$m_1 \textcircled{\cup} m_2(\mathrm{A}) = \sum_{\mathrm{B}\cup\mathrm{C}=\mathrm{A}} m_1(\mathrm{B})\, m_2(\mathrm{C}), \quad \forall \mathrm{A} \in 2^{\Omega}. \tag{5.14}$$

The disjunctive rule assumes that at least one of the two pieces of evidence is correct. Like the conjunctive rule, it is commutative and associative.

5.2.3 Canonical decomposition and Denoeux's cautious rule

Canonical decomposition of a belief function allows us to represent a complex non-dogmatic BBA[1] as the result of a combination (conjunctive combination will be referred throughout the discussion) of elementary belief states, namely Simple Support Functions (SSF) if the decomposed BBA is *separable* or a mixture of SSF and Inverse Simple Support Functions (ISSF) otherwise [143]. The canonical decomposition, besides being a convenient representation for some combinations, has its interest into allowing the introduction of new combination rules, as the Denoeux cautious conjunctive rule, that is the least committed rule among conjunctive ones [29].

The decomposition of a non-dogmatic BBA , defined by Smets [143], uses the concept of generalized Simple Support Function (GSSF), defined as:

$$\begin{aligned} \mu : 2^{\Omega} \to \mathbb{R}, \quad & \mu(\mathrm{A}) = 1 - w, \\ & \mu(\Omega) = w, \\ & \mu(\mathrm{B}) = 0 \quad \forall \mathrm{B} \in 2^{\Omega} \setminus \{\mathrm{A}, \Omega\}, \end{aligned}$$

[1] Dogmatic BBAs are transformed into non-dogmatic ones by an ϵ discounting.

where $A \neq \Omega$ and the weight $w \in \mathbb{R}^+$. The original BBA m can be then expressed as a combination of basic GSSFs: $m = \textcircled{\scriptsize$\cap$}_{A \subset \Omega} A^{w(A)}$. The conjunctive weight function $w(\cdot)$ is associated to any hypothesis included in discernment frame Ω:

$$\ln w(A) = -\sum_{B \supseteq A} (-1)^{|B|-|A|} \ln q(B), \quad \forall A \subset \Omega$$

According to GSSF definition, only weights $w(A) \neq 1$ are useful for representing the original BBA (i.e., not leading to a vacuous GSSF).

The canonical decomposition approach allows us to use an alternative combination rule, which is particularly useful when the source independence assumption (assumed by the conjunctive rule) is not valid. The Denoeux's cautious rule [29] between two sources m_1 and m_2, with w_1 and w_2 associated canonical decomposition weight functions, is defined as:

$$m_1 \textcircled{\scriptsize$\wedge$} m_2 = \bigoplus_{A \subset \Omega} A^{w_1(A) \wedge w_2(A)}$$

where $\wedge$ denotes the minimum operator.

5.2.4 Evidential q-relaxation

Recent work [124] introduces a BBA combination robust to unreliable sources. The *evidential q-relaxation*, inspired by its equivalent in interval analysis (IA), allows us to relax a given number of sources when combining several belief functions. Let us denote H_r^N the hypothesis that only r out of N sources are reliable, i.e. $q = N - r$ have to be relaxed. Let us call $\mathbf{A} = \{A_1, \ldots, A_N\}$, an N-tuple of hypotheses, for $A_i \subset \Omega, i \in [1, N]$. Out of the N hypotheses forming an N-tuple, only r must be kept. Such meta-knowledge can be mapped as [124]:

$$\Gamma_{\mathbf{A}}(H_r^N) = \bigcup_{\mathbf{A}' \subseteq \{A_1, \ldots, A_N\}, |\mathbf{A}'| = r} \left(\bigcap_{A \in \mathbf{A}'} A \right)$$

For any element $B \subseteq \Omega$, its mass will be the sum, over all $\Gamma_{\mathbf{A}}(H_r^N)$ which are equal to B, of the products of masses of the focal elements of A:

$$\forall B \subseteq \Omega, m\left[H_r^N\right](B) = \sum_{\mathbf{A} \subseteq \Omega^N, \Gamma_{\mathbf{A}}(H_r^N) = B} \left[\prod_{i=1}^{N} m_i^{\Omega}(A_i) \right]$$

Such a rule corresponds to a generalization of classic combination rules, since the special case of $r = N$ (i.e., do not relax any source) corresponds to the conjunctive rule, while the case $r = 1$ corresponds to the disjunctive rule.

5.3 Decision making

Among the decision rules that have been proposed (based on contour function etc.), we focus on the one proposed by [144], that is probably the most widely used. Decision making is usually performed at *pignistic* level, by transforming a mass function into a probability measure with a **pignistic transformation**:

$$\forall \omega \in \Omega, BetP(\omega) = \frac{1}{1 - m(\emptyset)} \sum_{B \supseteq \omega} \frac{m(B)}{|B|},$$

where ω is a singleton hypothesis.

Chapter 6

2CoBel: A Scalable Belief Function Representation for 2D Discernment Frames

Contents

6.1 Introduction

Belief function theory is widely used in fundamental tasks which benefit from multi-modal information fusion, such as object detection and data association for assisted driving [19, 30, 88], tracking [54, 160], object construction [134], outdoor localization [183], autonomous robot mapping and tracking [87, 156], medical imaging [11], remote sensing [89], video surveillance [85] , aircraft classification [44].

The main limitation, when dealing with such theory, since it copes with compound hypotheses, is the size of the set of hypotheses to handle, which may become intractable when the size of the discernment frame increases. Such issue becomes critical especially in high dimensions, as when dealing with two-dimensional (2D) spaces. Let us define a set (e.g. a frame or a focal set) as being a 2D set if its elements are elements of the Cartesian product of two totally ordered sets. Accordingly, a 2D discernment frame is defined as a frame of discernment which handles 2D focal elements. Now such 2D discernment frames can be encountered for instance for information fusion in the image domain (e.g. [134]), for box particle filtering [160], or in localization applications (e.g. [183]).

While several public belief function theory libraries exist [86, 105, 133], all of them limited to 1D representations, the use of 2D spaces for information fusion has been only recently explored for various tasks. Note that, active theoretical studies in BFT [24] propose geometric interpretations to the classical belief assignment, in order to formalize and solve problems such as probabilistic approximation and canonical decomposition. However, such works propose a geometrical formulation of the basic belief assignment (BBA) itself, while our focus is on geometrical representation of 2D focal elements.

In [134] the authors aim to reconstruct objects from fragmentary detections in the image space. The discernment frame corresponds to the 2D image lattice. BFT is then exploited in order to perform object-detection data association, spatial extension of objects when new fragments are found, temporal conditioning for object displacement/disappear modeling and spatial conditioning for object separation modeling. Focal elements are represented as sets of non-intersecting 2D-boxes.

In [183] 2D BFT is applied to the global navigation satellite system (GNSS) localization problem, where the information is represented by imprecise position measurements provided by several satellite sources, where complex focal elements shapes (ring sectors) are modeled as sets of boxes.

In [87] the authors perform scene environmental mapping by making extensive use of evidential grids for spatial and temporal fusion, by converting the original 2D domain in a map of 1D BBAs.

In this study, we focus on 2D discrete discernment frames. An exhaustive representation of Ω discrete hypotheses usually involves a discretization of the area as a grid, where each cell of the grid represents a singleton hypothesis [5, 87]. Focal elements are then expressed by using a binary word, where a bit equal to 1 means that the cell belongs to the focal set. Such straightforward binary-word representation of hypotheses allows for the definition of operators on sets through simple bitwise operations. However, such a representation suffers from major drawbacks when used in real world applications. Since there are $2^{|\Omega|}$ potential focal elements, large discernment frames become intractable, when the discretization resolution or the size of the whole area increases (different tasks may require different levels of precision for the solution, thus calling for a 2D space discretization which would increase quadratically the representation space size).

For such reasons, some works rely on different approaches to handle the 2D case: by proposing a smart sub-sampling of the 2D space to maintain tractability [5]; by proposing a sparse representation of the set of hypotheses, and by keeping in memory only the ones which are carrying non-null information [183]. In order to make the representation manageable, [5] proposes to condition the detections acquired from one sensor in its field of view, and to perform a coarsening at a lower spatial resolution of the focal elements, depending on the physical properties of the sensor. While these workarounds help in practice, they do not make the application fully scalable with the size of the scene, and they involve approximations such as the already cited coarsening, or frequent BBA simplification, which aims at maintaining under control the number of focal elements of the BBAs.

Such limitations derive from the fact that the complexity of any basic operator between focal elements (e.g. intersection, union) depends on the cardinality of the focal elements themselves. The works in [134] overcome this limitation by proposing a representation of any focal element as a set of rectangular boxes, and then by expressing the basic operators as performed on arrays of rectangles. In this setting the complexity of the basic operators will be a function of the number of boxes, but it will be independent of the cardinality of the discernment frame. However, such representation suffers from some practical limitations. First, the representation is not unique. The same focal element may be represented by different sets of boxes, which do not allow for fast focal element comparisons and lookup. Second, the box set representation implies a non-unique approximation of the real focal element shape once edges are not parallel to the axes of reference. Geometric approximations of such focal elements may require a very large set of boxes when precision is a concern. Moreover, subsequent operations involve increasing box fragmentation which

may be detrimental both for performance and for memory load. In order to avoid deep fragmentation, in [183] some representation simplification procedures are presented, which in turn increase the cost of BBA management.

In [150], the authors rely on a description of multidimensional focal elements by discretizing their support as a point cloud, and by deriving an approximation of Dempster's rule by Monte Carlo simulation. They underline the fundamental issue raised by representations based on parametric functions which lead to difficult implementation of the elementary operations (intersection, union, and complementation), which are needed for evidential reasoning. Alternatively to this representation, in this paper we propose a novel approach, which overcomes the efficiency issues of parametric representations.

Following the idea of providing a sparse representation for 2D BFT, and motivated by the great benefit that an efficient representation would carry to high dimensional problems, we propose a new two-dimensional representation which has full scalability properties with respect to the size of the discernment frame, while allowing a theoretical infinite precision (bounded by the hardware precision limitations).

6.2 BBA representation

In the following sections, two complementary representations are proposed.

In Section 6.2.1, a compact polygon-based geometric focal element description is detailed. It ensures *low-level* scalability for primitive operators, while exhibiting fast comparison and lookup capabilities. It represents a generalization of the state-of-the-art representations [134] [5], which overcomes their limitations (outlined in Section 6.1).

In Section 6.2.2, a graph-based BBA description is proposed. It encodes the spatial relationships between focal elements, and provides *high-level* scalability capabilities for e.g. decomposition methods (Section 6.3.3) and decision making algorithms (Section 6.4). Graph construction, optimization and traversal strategies are discussed.

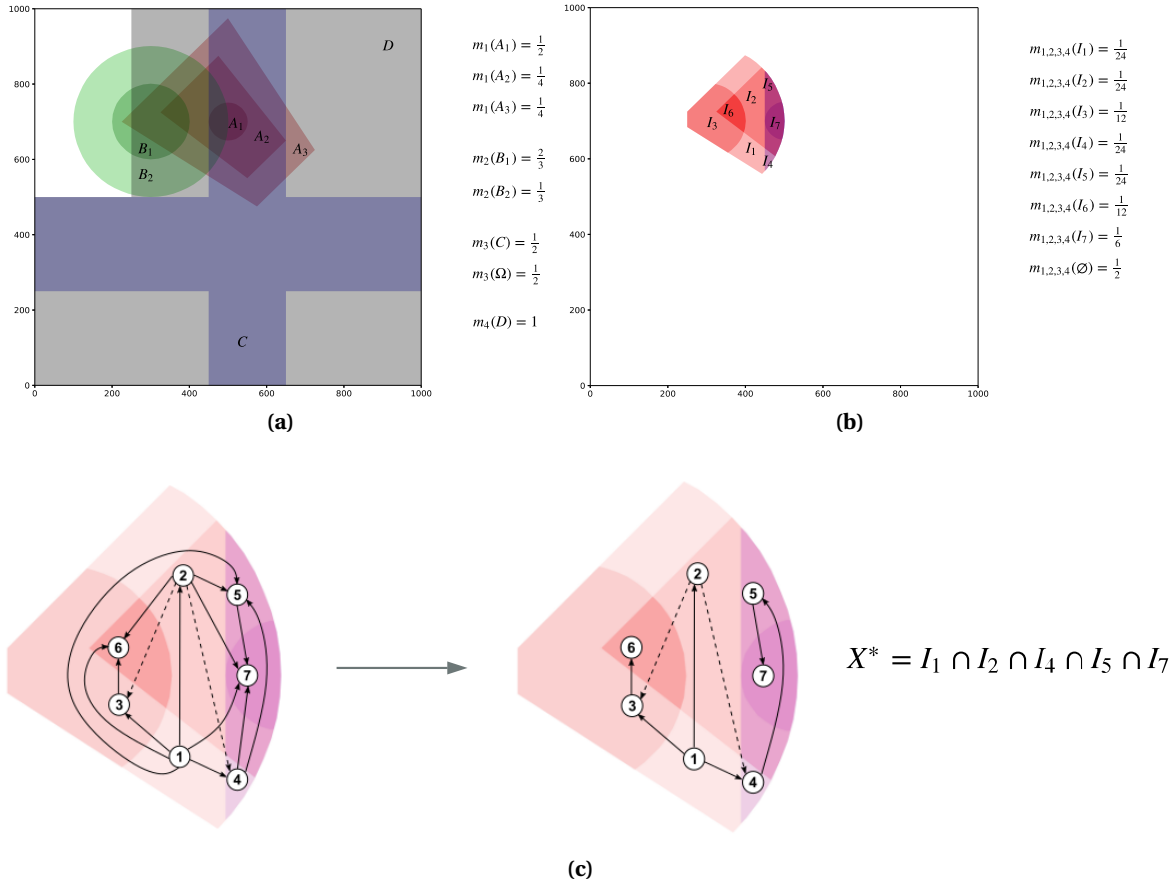
Let us consider a 2D discernment frame Ω. We will refer to the illustrative example in Figure 6.1. Such an example, inspired by [5], represents a typical localization scenario, where the discernment frame is a bounded region representing the ground plane.

6.2.1 Focal element geometric representation

As mentioned in the Introduction, 2D focal element representations based on an exhaustive representation of Ω and binary words become intractable once the size of one axis of the discernment frame becomes greater than a few tens of units, and the box set representation [134] suffers from geometric approximation due to the fact that the number of boxes needs to be limited to small values.

In this work, we propose to represent the focal elements as generic polygons (or sets of polygons for focal elements having multiple components, e.g. focal elements with holes or which are split after a difference operation), by exploiting the capabilities of the generic 2D polygon clipping algorithms, efficient methods for basic operator implementations (intersection, union, difference and XOR). A focal element is represented by a set of closed paths, each of them represented by an ordered array of vertexes (counter-clockwise for positive areas, clockwise for holes). As operator implementations, we exploit an extension of Vatti's algorithm for clipping [165] implemented in the Clipper library [72].

The polygons are constrained to be simple, i.e. defined by closed simple paths (no crossing), with a minimum number of vertexes (no vertex joining two co-linear edges). Figure 6.2 shows an example of focal element representation as two polygons (one of them representing a hole). Please note that, in case of multiple polygons per focal element, an additional constraint needs to hold: Given the set of paths composing a focal element, no edge of one path can cross an edge of another path. Under these constraints, the complexity of the basic operators between two poly-



(a) **(b)** **(c)**

Figure 6.1: Illustrative localization example. (a) BBA definition through its focal elements: camera detection m_1 (red), track at $t-1$ m_2 (green), road presence prior m_3 (blue), building presence mask m_4 (gray). (b) Focal elements obtained as a result of performing a conjunctive combination over the defined BBAs. (c) Intersection-inclusion graph and the result of graph simplification. The solid lines show the inclusion relationships, while the dashed lines highlight the intersection relationships. X* is the set with maximum BetP value, retrieved as the result of the proposed BetP maximization method.

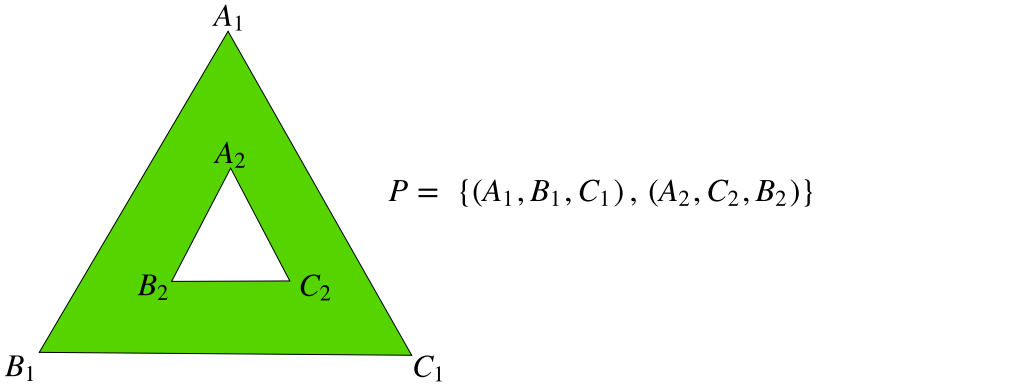


Figure 6.2: Example of representation of the focal element P (containing a hole), as a set of polygons. Please note as the external and the internal circular paths are stored in counter-clockwise and clockwise directions respectively.

gons having n and m numbers of vertexes respectively, is $O(nm)$. Such lightweight representation presents also the advantages of uniqueness and precision. The (circular) vector of vertexes of a focal element (polygon) provides a unique representation. The vertex coordinates use integer values for numerical robustness and correctness. This means that the continuous representation provided by polygons implies an underlining discretization. However, differently from the previous approaches, the coordinates can be rescaled at the desired level of precision (up to $\approx 10^{19}$) without any impact on the speed and memory requirements of the algorithm, being bounded only by the numerical representation limits of the hardware. This implies full scalability of the focal elements with respect to their size.

Example 1. *Figure 6.1a shows an example of focal elements in the case of a localization application. The camera detection (red) is represented as a disk focal element, whereas the focal elements which have the shape of ring sectors embed the imprecision of the location and the ill-knowledge of the camera extrinsic parameters; the track (green) represents the location of the target at the previous frame, whereas its dilation is used in order to model the imprecision in its position introduced by time; the gray and blue focal elements belong to two different BBAs representing scene priors, of building and road presence respectively. The disk shaped focal elements are modeled as 64 to 128 vertexes regular polygons.*

6.2.2 Graph-based representation

In the previous Section, we have highlighted that an efficient geometric representation, as the one proposed, may lead to the definition of basic operators (e.g. intersection, union) which are cardinality independent (and thus scalable). However, such representation alone cannot guarantee full scalability properties when dealing, for example, with decision making algorithms, which work at singleton hypothesis level.

Thus, claiming that a representation is spatially scalable requires some additional mechanisms which enforce the scalability at an higher operational level than primitive operators on sets. Together with an efficient geometric representation, we propose a generic representation, independent from the actual geometric representation chosen, which expresses the relationships between the focal elements of a given BBA. Indeed, many operations on BBAs, used for BBAs combination and decision making, can be expressed as algorithms which depend only on how the focal elements intersect with each other, irregardless of the actual shape or size of such elements. We propose to encode the relevant topological links between the focal elements as an *intersection-inclusion* graph, i.e an intersection graph where an edge is augmented in the case of a (directional) inclusion relationship.

Let us define the focal elements set as $\mathscr{A} = \{A_1, A_2, \ldots, A_n\}$. For optimization reasons explained further, the focal elements are labeled according to decreasing cardinality and the ordering follows the element label:

$$\forall \left(A_i, A_j\right) \in \mathscr{A} \times \mathscr{A}, \quad i < j \implies |A_i| \geq \left|A_j\right|.$$

Note that, in the case of different focal elements with the same cardinality, the topological order is not unique. Thus, the graph representation itself, differently from the geometric one, is not unique in general. However, non-uniqueness, given the low-level polygon representation, is not a necessary property. Moreover, graph optimizations and graph-based algorithms (detailed further) do not require uniqueness as a prerequisite.

We build a directed acyclic graph (DAG) $G = (V, E)$ where each node $v \in V$ is a focal element and each edge $e \in E$ represents a non empty intersection between two focal elements, with the direction of the edge respecting the topological ordering. The inclusion relationship information is encoded into separate arrays. Each node has a reference to its including nodes with the lowest and highest label. For example, if focal element A_5 is included into A_1, A_3, A_4, then the node v_5 carries two pieces of information, namely $l_5^l = 1$ and $l_5^h = 4$, where l_j^l and l_j^h stands for lowest and highest label including focal element A_j.

Let us define the k^{th} path of length m in the intersection-inclusion graph $G = (V, E)$ as $P_k^{(m)} = \langle v_{k,1}, v_{k,2}, \ldots, v_{k,m} \rangle$. Such path represents the intersection between all the focal elements related to the nodes included in the path. In the following, we will refer to the intersection *derived* from a path as the one computed from all its nodes.

Proposition 6.2.1. *For any non empty intersection* I *derived from a set* $\widehat{A}$ *of focal elements, there exists a path* P *in the intersection-inclusion graph* G*, connecting the elements of* $\widehat{A}$*.*

Proof. Let us consider $\widehat{A} = \{A_1, \ldots, A_m\}$ as the target set of focal elements ($I = \bigcap_{A_i \in \widehat{A_i}} A$). It follows that, given the graph $G = (V, E)$:

$$\forall (A_i, A_j) \in \widehat{A} \times \widehat{A}, i < j \quad \left|A_i \cap A_j\right| \neq 0 \Rightarrow v_i, v_j \in V, (v_i, v_j) \in E.$$

Since $I \neq \emptyset$, any node at index i is connected to every node at index j, such that $j > i$.

The above formula implies:

$$\forall v_i \in V, (v_i, v_{i+1}) \in E,$$

a sufficient condition for the existence of the path $P^{(m)} = \langle v_1, v_2, \ldots, v_m \rangle$. □

Definition 6.2.1. *A path* $P_k^{(m)}$ *is called* ***dead*** *if the intersection among the m focal elements corresponding to its nodes is the empty set.*

Definition 6.2.2. *Given two paths* $P_k^{(m)}$ *and* $P_h^{(n)}$*,* $P_k^{(m)}$ *is called* ***superpath*** *of* $P_h^{(n)}$ *if* $m > n$ *and:*

$$\forall v \in P_h^{(n)} \Rightarrow v \in P_k^{(m)}.$$

Conversely, $P_h^{(n)}$ *is called* ***subpath*** *of* $P_k^{(m)}$*.*

Definition 6.2.3. *A not dead path* $P_k^{(m)}$*, leading to the intersection* I_k *is called* ***redundant*** *if there exists another path* $P_h^{(n)}$ *leading to the intersection* I_h*, such that* $I_h = I_k$*, and* $P_h^{(n)}$ *is a superpath of* $P_k^{(m)}$*.*

In the following, we will refer to a path which is not redundant, equivalently, as a non-redundant path. While the graph structure can be used to explore all the possible intersections between focal elements, it shall be as efficient as possible in order to avoid exploring *dead* paths, while traversing only *non-redundant* paths, since they are by construction the paths carrying the greatest amount of structural knowledge (they gather all the focal elements which include the target intersection set).

The determination of the useful paths (i.e. neither *dead* nor *redundant*) is performed through graph traversal. According to the chosen ordering (decreasing cardinality), each node is iteratively selected as the root. For each root, a depth first search strategy is used to traverse the graph. Then, given the current node v_i, the intersection between all the nodes of the current path is propagated as I_i; given an edge $e = (i, j)$, the node A_j (for notation shortness a node is equivalently called by its represented focal element) is explored if $\left|I_j\right| = \left|I_i \cap A_j\right| > 0$. Such an operation is equivalent to performing a dynamic graph pruning which is a function of the current path, as soon as some branch leads to a *dead* path. Then, even if the number of node visits can be very large according to a brute force exploration, the dynamic pruning helps to cut out early *dead* paths, making the number of operations much lower in practice. In this form, however, the worst case for pruning happens with consonant BBAs, where every edge represents an inclusion relationship. In such case, the graph is complete, and the entire graph would need to be explored without any pruning possibility, even if most of the paths would be redundant. Such an observation, together with the fact that in practice consonant BBAs are widely used as basic representation of the initial imprecision increase with certainty, e.g. in the Dubois and Prade BBA allocation [33], justifies the use of the inclusion information for simplifying the graph and optimizing the traversal. Three main sources of optimization will be presented.

Proposition 6.2.2 (Root suppression)**.** *Given the current root node $v_j \in \mathrm{V}$, if:*

$$\exists v_i \in \mathrm{V}, (v_i, v_j) \in \mathrm{E}, i < j \quad s.t. \quad \mathrm{A}_j \subsetneq \mathrm{A}_i,$$

every path originating from root v_j is redundant.

Proof. Let us consider a generic path starting at the root v_j: $\mathrm{P}_k^{(m+1)} = \{v_j, v_{k,1}, \ldots, v_{k,m}\}$. By definition of a DAG, the index of the nodes $v_{k,h}, h \in \{1 \ldots m\}$, is higher than j, and, thus, than i. Let us consider the corresponding set of focal elements $\widehat{\mathrm{A}}_k = \{\mathrm{A}_j, \mathrm{A}_{k,1}, \ldots, \mathrm{A}_{k,m}\}$,

$$\mathrm{I}_k = \mathrm{A}_j \cap \left(\bigcap_{h=1\ldots m} \mathrm{A}_{k,h} \right),$$

and the augmented set $\widehat{\mathrm{A}}'_k = \{\mathrm{A}_i, \mathrm{A}_j, \mathrm{A}_{k,1}, \ldots, \mathrm{A}_{k,m}\}$,

$$\mathrm{I}'_k = \left(\mathrm{A}_i \cap \mathrm{A}_j\right) \cap \left(\bigcap_{h=1\ldots m} \mathrm{A}_{k,h} \right) = \mathrm{A}_j \cap \left(\bigcap_{h=1\ldots m} \mathrm{A}_{k,h} \right) = \mathrm{I}_k.$$

Moreover, the path $\mathrm{P}_k'^{(m+2)} = \{v_i, v_j, v_{k,1}, \ldots, v_{k,m}\}$ is a superpath of $\mathrm{P}_k^{(m+1)}$. It follows that $\mathrm{P}_k^{(m+1)}$ is a redundant path.

□

Root suppression implies that only root nodes which correspond to focal elements not included in some others preceding them in topological order, can produce paths which are *non-redundant*. Thus, all the other nodes are suppressed as possible roots for the traversal. Such property justifies the choice of a topological sorting in descending order of cardinality. Algorithm 6 shows where root suppression is used. After constructing the graph, each node is taken into account as candidate root for depth first search. Root suppression is used in order to filter root candidates for the following graph traversal operations.

Proposition 6.2.3 (Early stopping)**.** *Given the current root v_r, a path containing a node v_j included in a node $v_h, h < r$ is redundant.*

Proof. Let us consider the generic path starting at the root v_r and containing v_j: $\mathrm{P}_{a,b}^{(m+n+2)} = \{v_r, v_{a,1}, \ldots, v_{a,m}, v_j, v_{b,1}, \ldots v_{b,n}\}$. The corresponding set of focal elements is $\widehat{\mathrm{A}}_{a,b} = \{\mathrm{A}_r, \mathrm{A}_{a,1}, \ldots, \mathrm{A}_{a,m}, \mathrm{A}_j, \mathrm{A}_{b,1}, \ldots \mathrm{A}_{b,n}\}$ leading to the intersection $\mathrm{I}_{a,b}$. Since $\mathrm{A}_j \subset \mathrm{A}_h$, we can define the augmented set $\widehat{\mathrm{A}}'_{a,b} = \{\mathrm{A}_h, \mathrm{A}_r, \mathrm{A}_{a1}, \ldots, \mathrm{A}_{a,m}, \mathrm{A}_j, \mathrm{A}_{b1}, \ldots \mathrm{A}_{b,n}\}$, leading to the intersection:

$$\mathrm{I}'_{a,b} = \left(\mathrm{A}_h \cap \mathrm{A}_j\right) \cap \mathrm{A}_r \cap \left(\bigcap_{i=1\ldots m} \mathrm{A}_{a,i} \right) \cap \left(\bigcap_{i=1\ldots n} \mathrm{A}_{b,i} \right) = \mathrm{I}_{a,b},$$

since the term $\left(\mathrm{A}_h \cap \mathrm{A}_j\right)$ reduces to A_j.

Proposition 6.2.1 guarantees that the superpath $\mathrm{P}_{a,b}'^{(m+n+3)} = \{v_h, v_r, v_{a,1}, \ldots, v_{a,m}, v_j, v_{b,1}, \ldots v_{b,n}\}$ exists. Thus, the path $\mathrm{P}_{a,b}^{(m+n+2)}$ is redundant. □

The early stopping criterion allows us to stop exploring a node if it is included in an already explored root. The constraint is equivalently expressed as the fact that early stopping is performed at v_j if $l_j^l < r$ (among the nodes/focal elements that included v_j, indexed in $\left[l_j^l, l_j^h\right]$, there is at least one that has already been used as a root).

Early stopping is applied during graph exploration (see Algorithm 4 and Algorithm 5). It serves as a precondition for exploring or not a child of the current node.

Proposition 6.2.4 (Graph simplification)**.** *Given a node v_j which has multiple incoming inclusion edges from $\{v_i^h\}_{h=1\ldots m}$, all the edges but the one from the highest indexed node in topological order, (v_i^m, v_j), belong to redundant paths.*

Proof. First, one can demonstrate that, after removing the inclusion edges from $\{v_i^h\}_{h=1\dots m-1}$, v_j is still reachable from v_i^1. Since the edge between v_i^m and v_j is kept, it is equivalent to demonstrate that v_i^m is reachable from v_i^1.

$$\forall \mathrm{A}_i^k, \mathrm{A}_i^s, (k,s) \in \{1,\dots,m\}^2, \mathrm{A}_i^k \cap \mathrm{A}_j = \mathrm{A}_j, \mathrm{A}_i^s \cap \mathrm{A}_j = \mathrm{A}_j \Rightarrow \left|\mathrm{A}_i^k \cap \mathrm{A}_i^s\right| \geq \left|\mathrm{A}_j\right| \neq 0 \tag{6.1}$$

Thus, there exists a path from any node $\{v_i^h\}_{h=1\dots m-1}$ to v_i^m, and, consequently, to v_j.

Finally, we demonstrate that, if one of the removed edges is included in a path, that path is redundant.

Let us consider the path $\mathrm{P}_{a,b}^{(n+q+2)} = \{v_{a,1},\dots,v_{a,n},v_i^h,v_j,v_{b,1},\dots v_{b,q}\}$, where $h < m$, leading to the intersection $\mathrm{I}_{a,b}$. Let us consider the node v_i^k, $h < k \leq m$, which, by topological ordering, cannot be already included into $\mathrm{P}_{a,b}^{(n+q+2)}$.

Let us consider the superpath $\mathrm{P}_{a,b}^{\prime(n+q+3)} = \{v_{a,1},\dots,v_{a,n},v_i^h,v_i^k,v_j,v_{b,1},\dots v_{b,q}\}$, leading to the intersection $\mathrm{I}'_{a,b}$. Two edges have been added: (v_i^h, v_i^k), guaranteed to exist by Equation (6.1); (v_i^k, v_j), that exists by definition of the problem.

$$\mathrm{I}'_{a,b} = \mathrm{A}_i^h \cap \left(\mathrm{A}_i^k \cap \mathrm{A}_j\right) \cap \left(\bigcap_{i=1\dots n} \mathrm{A}_{a,i}\right) \cap \left(\bigcap_{i=1\dots q} \mathrm{A}_{b,i}\right) = \mathrm{I}_{a,b},$$

since the term $\left(\mathrm{A}_i^k \cap \mathrm{A}_j\right)$ reduces to A_j.

Thus, $\mathrm{P}_{a,b}^{(n+q+2)}$ is a redundant path. □

Graph simplification boils down to keeping, for each v_j, only the incoming inclusion connection from the node with the highest index in topological order l_j^h. Algorithm 3 details the graph construction steps, and shows how graph simplification is exploited. Any intersection edge between two nodes is added immediately, while addition of the inclusion edges is delayed until all the pairs of nodes are inspected (only l_j^l and l_j^h indexes are updated). At the end, for each node j, only the inclusion edge from l_j^h is created (if l_j^h is not null).

Now consider the case of a consonant BBA with k focal elements. In its pure form, the representation leads to a complete DAG, with 2^k possible paths. However, after graph simplification, only the edges going from element A_i to A_{i+1} are kept, resulting into $k-1$ effective edges. Moreover, due to root suppression, only the first node will be used as root, so k nodes in total will be explored, leading to k non-redundant paths, providing k different intersections, equal to the k original focal elements.

Node	l^l	l^h	Use as root	Deleted edges (simplification)
1	null	null	yes	-
2	1	1	no	-
3	1	1	no	-
4	1	1	no	-
5	1	4	no	$(v_1 \rightarrow v_5), (v_2 \rightarrow v_5)$
6	1	3	no	$(v_1 \rightarrow v_6), (v_2 \rightarrow v_6)$
7	1	5	no	$(v_1 \rightarrow v_7), (v_2 \rightarrow v_7), (v_4 \rightarrow v_7)$

Table 6.1: Graph optimization main steps for the illustrative example in Figure 6.1.

Example 2. *We refer to the example in Figure 6.1. The graph on the left of Figure 6.1c illustrates the result of unoptimized graph construction. Intersection relationships are shown as dashed arrows, while inclusion relationships are depicted as solid arrows. The optimization steps are shown in Table 6.1. First, v_1 includes all other focal elements, thus only v_1 will serve as a root for graph traversal.*

Then, since v_5, v_6, v_7 have multiple including nodes, for each of such nodes, all the incoming inclusion edges are deleted but the one arriving from the node with label l^h (4, 3 and 5, respectively). The graph on the right side of Figure 6.1c shows the final form of the intersection-inclusion graph for the given BBA. The X^* set corresponds to the BetP maximizer set from the optimized graph, explained in the Example 6 in Section 6.4.

Algorithm 3: BUILDGRAPH

Data: Set of focal elements $\mathscr{A}$ ordered by decreasing cardinality;
Result: Simplified DAG G = (V, E); lowest including set label array $\mathbf{l^l}$; highest including set label array $\mathbf{l^h}$;

```
1  begin
2      V = {};
3      E = {};
4      for each A_i ∈ 𝒜 do
5          V ← V ∪ {v_i};
6          for each A_j ∈ 𝒜, j > i do
7              if A_j ⊂ A_i then
8                  l^h_j = i;
9                  if l^l_j is null then
10                     l^l_j = i;
11                 end
12                 //delay inclusion edge storage (graph simplification)
13             end
14             else if |A_i ∩ A_j| > 0 then
15                 E ← E ∪ {(v_i → v_j)}; //storing intersection edge
16             end
17         end
18     end
19     for j in 1...|𝒜| do
20         if l^h_j is not null then
21             E ← E ∪ {(v_{l^h_j} → v_j)}; //storing inclusion edge
22         end
23     end
24 end
```

Algorithm 4: DFS (SET OF DISJOINT SETS EXTRACTION)

Data: DAG G = (V, E); set of focal elements $\mathscr{A}$; root label r; lowest including set label array $\mathbf{l}^{\mathbf{l}}$; current node v_i; current intersection I_i; current path p; sets of indexes of the disjoint sets included in each focal element S; output: set of disjoint sets $\mathscr{D}$.

```
1  begin
2      p ← p ∪ {v_i};
3      for each e_ij = (v_i → v_j) ∈ E do
4          if l^l_j < r then
5              continue; //early stopping
6          end
7          I_j = I_i ∩ A_j;
8          if |I_j| > 0 then
9              DFS(G, 𝒜, r, l^l, v_j, I_j, p, S, 𝒟);
10         end
11     end
12     h_p = {1, ..., |𝒟|};
13     for v ∈ p do
14         h_p = h_p ∩ S[v]; //common disjoint sets among elements of the path
15     end
16     D = I_i
17     for h ∈ h_l do
18         D = D \ 𝒟[h]; //subtract from D the included disjoint sets
19     end
20     if |D| > 0 then
21         𝒟 ← 𝒟 ∪ {D};
22         for v ∈ p do
23             S[v] ← S[v] ∪ {|𝒟|}; //focal element at node v contains D
24         end
25     end
26     p ← p \ {v_i};
27 end
```

Algorithm 5: DFS (SET OF MAXIMAL INTERSECTIONS EXTRACTION)

Data: DAG G = (V, E); set of focal elements $\mathscr{A}$; root label r; lowest including set label array $\mathbf{l^l}$; current node v_i; current intersection I_i; current path p; set of paths leading to each maximal intersection P; output: set of maximal intersections $\mathscr{I}$.

```
1  begin
2      p ← p ∪ {v_i};
3      leaf= true;
4      for each e_ij = (v_i → v_j) ∈ E do
5          if l^l_j < r then
6              continue; //early stopping
7          end
8          I_j = I_i ∩ A_j;
9          if |I_j| > 0 then
10             leaf= false;
11             DFS(G, 𝒜, r, l^l, v_j, I_j, p, P, ℐ);
12         end
13     end
14     if is a leaf then
15         maximal= true;
16         for p_I ∈ P do
17             if p ⊂ p_I then
18                 maximal= false;
19                 break;
20             end
21         end
22         if is maximal then
23             ℐ ← ℐ ∪ {I_i};
24             P ← P ∪ {p};
25         end
26     end
27     p ← p \ {v_i};
28 end
```

Algorithm 6: Graph construction and traversal

Data: Set of focal elements $\mathscr{A}$ ordered by decreasing cardinality;

```
1  begin
2      (V, E, l^l, l^h) =BUILDGRAPH(𝒜);
3      for each v_i ∈ V do
4          if l^l_j is not null then
5              continue; //root suppression
6          end
7          DFS((V, E), 𝒜, i, l^l, v_i, A_i, {}, {}, {}) //either Algorithm 4 or 5
8      end
9  end
```

6.3 BBAs combination

6.3.1 Classical combination rules and hashing

Numerous combination rules exist in order to mix the information provided by two sources. When the sources m_1 and m_2 are cognitively independent, the conjunctive combination rule is the most popular among them (see Section 5.2.1 of Chapter II.3).

In computational terms, the rule involves the construction of a new BBA by performing intersection operations between all pairs of focal elements from the two BBAs. According to the sum in the previous equation, when creating a new focal element from an intersection, one has to check for its existence and to add up some elementary mass product value if it already exists. The necessity of accumulating elementary masses into already existent focal elements (maybe computed with different operators than intersection, e.g. union in the case of the disjunctive rule), and thus to do an existence check every time a new mass value is computed, is not specific of the conjunctive rule, but it is shared with several other rules (e.g. the disjunctive rule [29], cautious rule [29] and evidential q-relaxation [124]).

The above considerations justify the need for a BBA representation which allows for a fast lookup of a focal element in an array. The uniqueness and compactness of the proposed representation allow for an efficient and low collision prone hashing. The sparse set of focal elements of a given BBA can be stored in a hash table, where the circular vector of vertexes is used to compute the hash. For a given polygon, its hash will be unique provided that we fix a policy to decide the starting vertex (e.g. the top left). The array hashing function is equivalent to the one implemented in the Boost library's [12] *hash_range* method.

The binary-word representation, in comparison, uses the full word as a unique key. However, the key length (in number of bits) grows linearly with the cardinality of the discernment frame, requiring the use of big data structures in order to store it. On the contrary, the proposed hash exhibits collision resistance property despite a fixed length. The box set representation [183], being not unique, does not allow for direct hashing without the extraction of the minimal set of vertexes on the boundary. A cheap alternative could be to hash the bounding box of the focal element, but this could cause frequent collisions, since it is common to have spatially close focal elements related to the same BBA. On the contrary, polygon hashing can make direct use of the vertex data, thus not requiring any additional preprocessing step.

Note that the hashing capability of the proposed representation may provide benefits not only in case of BBAs combination rules, but for any operator which implies mass accumulation. Let us consider the case of coarsening [31]. Given two discernment frames Ω_1 and a finer Ω_2, we define a refining function ρ from 2^{Ω_1} to 2^{Ω_2}, such that $\{\rho(\{o\}), o \in \Omega_1\}$ is a partition of Ω_2 and $\forall A \subseteq \Omega_1, \rho(A) = \bigcup_{o \in A} \rho(o)$. Let us consider the coarsening function ρ^{-1}. According to [31], the least committed solution for defining ρ^{-1} is the following outer reduction function:

$$\forall B \subseteq \Omega_2, \rho^{-1}(B) = \{o \in \Omega_1 \; s.t. \; \rho(o) \cap B \neq \emptyset\}.$$

The BBA $m^{\Omega_2 \downarrow \Omega_1}$ defined on Ω_1 from a given BBA m^{Ω_2}, defined on Ω_2, and the coarsening function ρ^{-1}, is given by:

$$\forall A \in 2^{\Omega_1}, m^{\Omega_2 \downarrow \Omega_1}(A) = \sum_{B \subseteq \Omega_2, \rho^{-1}(B) = A} m^{\Omega_2}(B).$$

It follows that the derivation of $m^{\Omega_2 \downarrow \Omega_1}$ can largely benefit from hashing, since a focal element mass may be the result of several fragmented masses. Similar deductions can be derived for operations such as conditioning [140], or uncertainty propagation [34].

Example 3. *Figure 6.1b illustrates the result of the conjunctive combination of the sources introduced in Figure 6.1a. Seven focal elements are produced.*

6.3.2 Evidential q-relaxation

In Section 5.2.4 of II.3 we have described evidential q-relaxation as a BBA combination robust to unreliable sources. In computational terms, when considering the computation of the $\Gamma_{\mathbf{A}}(\mathrm{H}_r^{\mathrm{N}})$ terms that appear combinatorial in terms of N and r, the algorithm has higher time complexity than classic combination approaches, depending on the value of q. However, even for small values of q, such a method can dramatically improve the fusion performance in presence of outlier sources. Moreover, the proposed representation boosts (once more) the efficiency of the method, in terms of the efficient basic operators (the method makes heavy use of intersection and union operators), as well as of the use of hashing for fast accumulation of elementary masses.

6.3.3 Canonical decomposition and cautious rule

Let us consider the canonical decomposition operation on non-dogmatic BBAs (see Section 5.2.3 of Chapter II.3). When performing canonical decomposition, the step which has the highest impact in terms of computation is the calculation of the weight function. While consonant BBAs represent a special case for weight computation, where an iterative algorithm with linear complexity in terms of the number of focal elements exists, the generic case raises computational problems.

In the 1D case, the fastest weight computation approach exploits the Fast Möbius Transform (FMT) [78] for transforming the $2^{|\Omega|}$ array of masses (dense representation) to a $2^{|\Omega|}$ array of commonalities q, and finally to a $2^{|\Omega|}$ array of weights. While this procedure is convenient for small Ω cardinalities, it is computationally infeasible for large discernment frames, like in the 2D case.

In this work, we propose to compute efficiently the canonical decomposition by constructing an ad-hoc discernment frame specific to the considered BBA. The idea is that, even if the considered 2D discernment frame Ω is vast, for the considered BBA, the number of focal elements is limited and generally they are less than a few tens. Thus, the considered BBA can be represented on only a very restricted subpart of Ω with an effective spatial resolution that depends on the actual focal elements and will generally be much coarser than Ω resolution. Even more, the 2D structure of the focal elements is useless provided that the interaction properties between focal elements are preserved. Thus, for a given BBA, we aim to compute the coarsest possible equivalent representation where each element can be viewed as a unitary piece of information. If one retrieves this alternative representation, then it may be transformed into a 1D equivalent (by processing each of its elements as a singleton hypothesis) where the discernment frame is small, and thus suited for the FMT computation. Thus, the canonical decomposition no longer depends on the shape and original cardinality of the focal elements, but on their ad-hoc representation.

From m^{Ω} to coarse representation.

Definition 6.3.1. *A **set of disjoint sets** $\mathscr{D}$ is a partition of the disjunction $\mathscr{X} = \bigcup_{\mathrm{A}\in\mathscr{A}} \mathrm{A}$ of the set of focal elements $\mathscr{A}$, where each element D_i (namely, a **disjoint set**), satisfies:*

$$\forall \mathrm{D}_i \in \mathscr{D}, \begin{cases} \mathrm{D}_i & \subset & \mathscr{X}, \\ \left|\mathrm{D}_i \cap \mathrm{D}_j\right| & = & 0,\ \forall \mathrm{D}_j \in \mathscr{D}, i \neq j \\ \forall \mathrm{A} \in \mathscr{A},\ |\mathrm{D}_i \cap \mathrm{A}| \neq 0 & \Longleftrightarrow & \mathrm{D}_i \subseteq \mathrm{A} \end{cases}$$

In addition, defining the set $\hat{\mathrm{A}}_i$:

$$\hat{\mathrm{A}}_i = \{\mathrm{A}\}_{\substack{\mathrm{A}\in\mathscr{A} \\ \mathrm{D}_i\subseteq\mathrm{A}}},$$

as the set of all the focal elements including D_i, the following condition should hold (maximal coverage):

$$\forall \mathrm{D}_i \in \mathscr{D},\ \nexists \mathrm{D}_j \in \mathscr{D}, i \neq j,\ s.t.\ \hat{\mathrm{A}}_i = \hat{\mathrm{A}}_j$$

The coarsest possible representation consists in a subdivision of the discernment frame into a set $\mathscr{D}$ of *disjoint sets*. Basically, a set of disjoint set is the minimum cardinality set of non intersecting sets which do not cross the boundaries of any focal element. The set has minimal cardinality because of the maximal coverage constraint on each disjoint set.

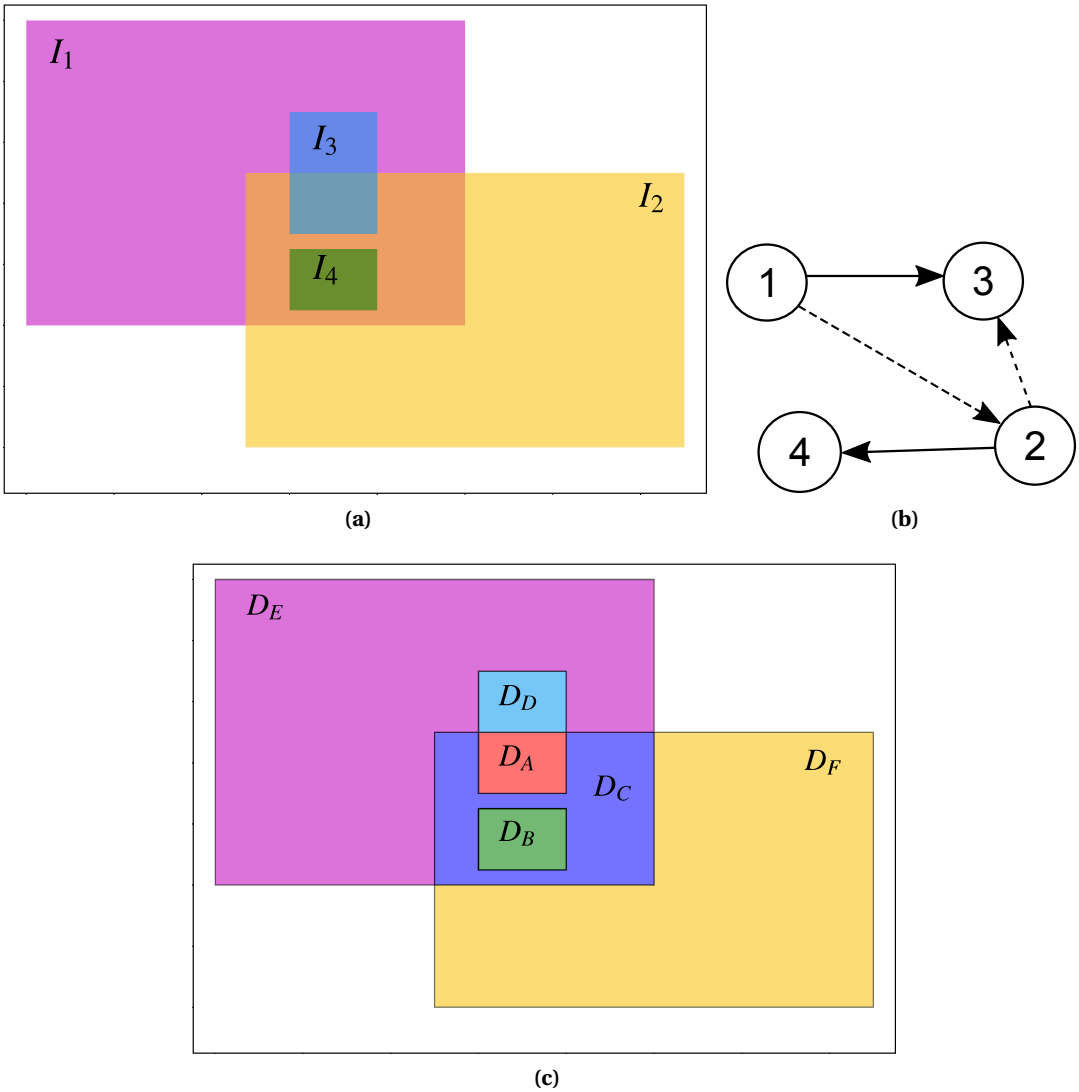


Figure 6.3: Illustrative example of coarse discernment frame extraction for canonical decomposition. (a) Initial BBA definition (focal elements are labeled); (b) Optimized intersection-inclusion graph for the given BBA: since I_4 is included in both I_1 and I_2, the edge between v_1 and v_4 has been deleted by graph simplification; (c) Final set of disjoint sets extraction (disjoint sets are labeled). In order to stress that the x and y axes are generic (dependent on the application), they are not labeled.

In order to extract the set $\mathscr{D}$, one may exploit the graph representation introduced in Section 6.2.2. In graph terms, maximal coverage constraints guarantee that exactly one disjoint set for each non-redundant path can be constructed (thus, minimizing the number of disjoint sets). The graph traversal is conducted in a depth first search manner. Let us consider the addition of a new D_i element to $\mathscr{D}$, that is the result of the intersection of all the focal elements in the path. For each A_k in the explored path, a reference to the i^{th} disjoint set is stored in an auxiliary set of labels S_k. S_k represents the set of all disjoint sets included into the focal element A_k. Now, when a new candidate $\widetilde{D}_l$ is found, it is not guaranteed to be disjoint from elements already present in $\mathscr{D}$. In order to extract the related set D_l one has to apply the *difference* operator between $\widetilde{D}_l$ and any element already included in $\mathscr{D}$. The information regarding which disjoints sets are possibly included comes from the S_k sets. If $\widetilde{D}_l$ is the result of the intersection along the path of length m $P_l = \{A_{l,1}, \ldots, A_{l,m}\}$, the indexes h_l of the disjoint sets to subtract can be retrieved as:

$$h_l = \bigcap_{k=1\ldots m} S_{l,k},$$

representing the labels of all the disjoint sets included into $\widetilde{D}_l$. The resulting D_l will be obtained as:

$$D_l = \widetilde{D}_l \setminus \{D_h\}_{h \in h_l}$$

If the sets are implemented as bit strings, the disjoint sets retrieval is as fast as m bitwise operations. Please note that after the difference operation, the resulting disjoint set could be empty, and so ignored. Algorithm 4 provides a detailed outline of the set of disjoint sets extraction procedure through depth first search traversal.

	$\widetilde{D} = \bigcap_i I_i$				D
	I_1	I_2	I_3	I_4	
A	✓	✓	✓		$D_A = \widetilde{D}_A$
B	✓	✓		✓	$D_B = \widetilde{D}_B$
C	✓	✓			$D_C = \widetilde{D}_C \setminus \{D_A, D_B\}$
D	✓		✓		$D_D = \widetilde{D}_D \setminus \{D_A\}$
E	✓				$D_E = \widetilde{D}_E \setminus \{D_A, D_B, D_C, D_D\}$
F		✓			$D_F = \widetilde{D}_F \setminus \{D_A, D_B, D_C\}$

Table 6.2: Coarse representation computation from graph representation.

Example 4. *Figure 6.3 shows the disjoint set decomposition on a 2D didactic example. Each disjoint set corresponds to a subset of one or multiple focal elements which does not span over a focal element boundary. Table 6.2 specifies the disjoint set computation procedure, as $\mathscr{D}$ elements are extracted in the order assigned by the graph traversal. Each row corresponds to a path which possibly leads to a disjoint set. The candidate disjoint set $\widetilde{D}_A$, for example, is the result of $I_1 \cap I_2 \cap I_3$. Candidates are then transformed into actual disjoint sets by the difference operator. Let us specify the case of $\widetilde{D}_D$. Since $\widetilde{D}_D$ is the intersection between I_1 and I_3, we focus on the sets $S_1 = \{A, B, C\}$ and $S_3 = \{A\}$. In this example, the content of S_1 means that, among all the already extracted $\mathscr{D}$ elements, D_A, D_B, D_C are included into I_1. The vector of indexes to subtract is computed as $h_l = S_1 \cap S_3 = \{A\}$. Thus, among all the already extracted $\mathscr{D}$ elements, $\widetilde{D}_D$ includes the disjoint set D_A, which has to be subtracted. Finally $D_D = \widetilde{D}_D \setminus \{D_A\}$. Please note that the intersections $I_2 \cap I_3$ and $I_2 \cap I_4$ are not explored, since both I_3 and I_4 are included in I_1, which is an already explored root (early stopping). Moreover, for the same reason, nodes corresponding to I_3 and I_4 are not used as root (root suppression).*

From the coarse representation to 1D BBA. As soon as the $\mathscr{D}$ set is fully constructed, the sets S_k serve to the conversion between the original BBA and a compact 1D representation. The new

1D BBA has a new discernment frame Ω' of cardinality $|\mathscr{D}|$. The indexes of the disjoint sets represent the singleton hypotheses the 1D BBA in Ω'. Each focal element A_k is converted to a compound 1D hypothesis by using the S_k, which stores the indexes of the disjoint sets in $\mathscr{D}$ which, when performing a union operation, form the exact original set. Thus, each A_k gives rise to:

$$m^{\Omega'}\left(\bigcup_{s\in S_k}\{s\}\right) = m^{\Omega}(A_k)$$

Given that the cardinality of the ad-hoc 1D discernment frame Ω' is lower than a few tens of elements in practical cases, it is then suitable for the application of the FMT for the weight computation. Once the weights along with the corresponding canonical decomposition sets are retrieved, the canonical decomposition of the original BBA can be obtained by mapping the 1D decomposition sets, expressed as union of singleton 1D hypotheses, to 2D sets, expressed as union of elements of $\mathscr{D}$.

Example 5. *We refer to the decomposition depicted in Figure 6.3. The new discernment frame is* $\Omega' = \{A, B, C, D, E, F\}$. *All the original focal elements translate to focal elements in* Ω'. *For example,* $m^{\Omega'}(\{A, D\}) = m^{\Omega}(I_3)$.

Such canonical decomposition approach is now fully scalable with the cardinality of the discernment frame, and it is especially convenient when the number of focal elements is much smaller than $|\Omega|$. Since the conjunctive canonical decomposition is trivially propagated when applying the conjunctive rule to two canonically decomposed BBAs, the typical scenario of applying the decomposition (e.g. for tracking), is at the BBA construction stage and when BBA approximation is needed. At BBA construction stage, the number of focal elements is usually contained, and the consonant property of a BBA may be exploited for even faster computation. At BBA approximation stage, the number of focal elements is intentionally reduced, while the BBA is not consonant in general, making it the ideal scenario for the exploitation of the presented approach.

As denoted in Section 5.2.3 of Chapter II.3, the canonical decomposition extraction approach allows us to use the cautious rule. In algorithmic terms a new canonical decomposition is built by including all the elements of the two initial decompositions with weight value lower than one. As for the case of the conjunctive rule, hashing can still be used for fast lookup of equal elements in the two decompositions.

6.4 Decision making

Once the different sources have been combined, the decision is generally taken on singleton hypotheses ω by maximizing the *pignistic probability* (see Section 5.3 of Chapter II.3).

Even if the search space size is now $|\Omega|$, the decision making process is dependent on the cardinality of the discernment frame, and thus not scalable, limiting the precision level which can be set for a specific context.

In order to overcome this limitation, we propose a maximization algorithm which is independent from the cardinality of the sets, and which is only related to the number of focal elements in the BBA.

Definition 6.4.1. *Given a set of focal elements* $\mathscr{A} = \{A_1, \ldots, A_n\}$ *a* ***maximal intersection*** I_m *is derived from the set of focal elements* $\tilde{\mathscr{A}} \subseteq \mathscr{A}$, *such that any different focal element added to* $\tilde{\mathscr{A}}$ *would lead to an empty intersection:*

$$I_m = \bigcap_{A_k\in\tilde{\mathscr{A}}} A_k, \tilde{\mathscr{A}} \subseteq \mathscr{A}, |I_m| > 0 \; s.t.$$
$$\nexists A_s \in \mathscr{A} \setminus \tilde{\mathscr{A}}, |A_s \cap I_m| > 0.$$

The underlying idea is that, since BetP is an additive measure, its maximum value can be achieved only for elements of the discernment frame which present *maximal intersections*.

The set X^* of hypotheses that maximize the BetP is researched within the set of maximal intersections $\mathscr{I}$:

$$X^* = \arg\max_{I_m \in \mathscr{I}} \frac{Bet P(I_m)}{|I_m|},$$

where the BetP function for compound hypotheses derives from the generalized formula:

$$\forall A \in 2^{\Omega},\ Bet P(A) = \frac{1}{1 - m(\emptyset)} \sum_{B \in \mathscr{A}, B \cap A \neq \emptyset} \frac{|A \cap B|}{|B|} m(B).$$

Consequently to this formulation, the BetP maximization algorithm boils down to the subproblem of maximal intersection search. The solution of this subproblem may exploit of the graph-based representation presented in Section 6.2.2 for fast lookup of maximal intersections.

Corollary 6.4.1. *Given the intersection-inclusion graph G, a* ***maximal intersection*** I_m *is represented by a non-redundant path* P_m *which is not a subpath of any other non-redundant path.*

Proof. Let us assume that I_m is redundant. Thus, there exists a superpath leading to the same intersection. Then, I_m cannot be maximal.

Let us assume that P_m is a subpath of another non-redundant path P_n, $n > m$. Thus, since non-redundant paths cannot be dead paths, P_n leads to a non empty intersection. Then, I_m cannot be maximal. □

The graph-related definition of maximal intersection implies that any intersection not being located at a leaf of the dynamic graph cannot be a maximal intersection. The graph is said dynamic in the sense that a leaf is not only a node with no outgoing edges, but it is any node for which, given the current path, no outgoing edge can be explored further without leading to a dead path. Then, each leaf l and the resulting I_l is a candidate for maximal intersection. However, it could be non-maximal, as its associated set $\tilde{\mathscr{A}}$ could be a subset of a maximal intersection which has already been found. So, when a maximal intersection I_m is found, the list p_m of focal sets involving it is stored (using a bit-set representation). Once the new candidate I_l is produced, the p_l list is tested for inclusion against the stored candidates (by an AND operation between the bit-sets). Algorithm 5 provide full details on the depth first search strategy for the computation of maximal intersections.

Path	Maximal intersection
$\langle v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_6 \rangle$	yes
$\langle v_1 \rightarrow v_2 \rightarrow v_4 \rightarrow v_5 \rightarrow v_7 \rangle$	yes
$\langle v_1 \rightarrow v_3 \rightarrow v_6 \rangle$	no
$\langle v_1 \rightarrow v_4 \rightarrow v_5 \rightarrow v_7 \rangle$	no

Table 6.3: Maximal intersection search details for the illustrative example in Figure 6.1.

Example 6. *Table 6.3 shows the intersection-inclusion graph traversal for maximal intersection search on the intersection-inclusion graph of the illustrative example of Figure 6.1.* X^* *is selected as the one of the two maximal intersections at maximum* BetP. *For this example, raw traversal intersection graph would perform 42 node visits, while with the graph optimizations, presented in Section 6.2.2, 12 are executed. On the other hand, a straightforward* BetP *maximization by singleton hypothesis exploration would process 1100 locations (included into at least one focal element) with a factor 10 subsampling of the discernment frame.*

6.5 Experiments

We present test results on a synthetic toy example, as well as on a real tracking application scenario, which make use of the proposed representation, as well as of our publicly available *2CoBel* library, embedding all the described methodologies, and exploited throughout the entire testing.

6.5.1 The 2CoBel library

2CoBel is an open source[1] evidential framework embedding essential functionalities for generic BBAs definition, combination and decision making. An *Evidence* object defines common operations for a BBA containing any generic type of *FocalElement*. The current supported methods are: mass to Belief Functions conversion (plausibility, belief, commonality), conjunctive, disjunctive, cautious (exploiting the proposed canonical decomposition) rules and q-relaxation, vacuous extension and marginalization, conditioning, discounting, (generalized) BetP computation, BetP maximization (with singleton hypothesis enumeration or maximal intersections). Different representations of *FocalElement* are supported, each of them defining specifically basic operators (intersection, union, equality, inclusion): *unidimensional* (hashable), representing the 1D focal element as a binary string; *2D bitmap*, providing a bitmap representation as in [5]; *2D box set*, implementing the definition and focal elements simplification operations proposed in [183]; *2D polygon* (hashable), implementing our proposed representation.

The library has full support for discernment frames which are cartesian products.

6.5.2 Case study: line estimation

As toy example for illustrating the applicability of the proposed representation to 2D domains, we tackle a fundamental problem in pattern recognition, namely the line estimation from a set of 2D points. This example allows us to compare as well the effectiveness of the different combination rules implemented by the framework. Given a set of planar points in the xy space, the objective is to infer the parameters of the line that fits at best the data. The Hough transform [66] is a classical approach to this problem, and the evidential framework allows us to handle the intrinsic imprecision and uncertainty sources of the problem in the Hough domain. By using the polar representation of lines:

$$\rho = x\cos\theta + y\sin\theta,$$

we build an accumulation space in the $(\rho;\theta)$ domain in order to infer the values of the $\rho \in (-\infty, +\infty)$ and $\theta \in [0, \pi)$ parameters. The discernment frame Ω is then defined as a rectangular polygon in the $(\rho;\theta)$ space. Since the ρ parameter is unbounded, theoretically also Ω is an open set. However, in order to respect close world assumptions, we bound Ω to extremely large values of ρ. Please note that, due to the sparse nature of the representation, the size of the discernment frame has no impact on the algorithm performance, so extreme values of ρ equivalent to the max and min integer values supported by the hardware could be chosen.

Each data point $P_i = (x_i, y_i)$ votes for a family of lines which pass through it in the xy plane. Each voted line $l^i_j = (\rho_j, \theta_j)$ corresponds to a point in the accumulation space. The locus of all the points in the accumulation space corresponds to a sinusoid function. Such toy example extends the one presented in [183], which performs straight line estimation in the $(\alpha;\beta)$ space (where $y = \alpha x + \beta$). However, such space does not parametrize any possible line, and, moreover, a small possible interval of values has been considered in order to have a discernment frame of small size. On the one hand, the $(\alpha;\beta)$ approach allows one to represent the constraints as straight lines rather than sinusoids, allowing for an inexpensive focal element representation. On the other hand, the proposed representation allows us to move to a more convenient space where complex shapes can be defined.

[1] Implementation available at:
https://github.com/MOHICANS-project/2CoBel

In classical Hough approaches, since there is an infinite number of lines passing through the same point, the space $(\rho;\theta)$ is quantized in such a way as to provide an acceptable precision. In the proposed representation, the resolution of the problem is given by the number of vertexes used to represent the polygons. However, the scalability of the problem allows us to rescale more flexibly the accumulation space for high precision estimations. In this experimentation we scale the accumulation space at a resolution of 10^{-2} for both ρ and θ (in degrees).

The BBA construction consists in widening the sinusoidal function derived from each point in the dataset with imprecision and uncertainty knowledge. We build a consonant BBA having two focal elements. The first focal element is a sinusoidal band with width equal to $\delta\rho_1$, centered around the real sinusoidal function drawn from the data. Such focal element codes the imprecision of the points location given by the line discretization in the xy space (typical if the space is in the image domain). The second focal element is a sinusoidal band with width equal to $\delta\rho_2 > \delta\rho_1$, again centered around the real sinusoid. Such focal element encodes the uncertainty of the point location given by noisy data distribution.

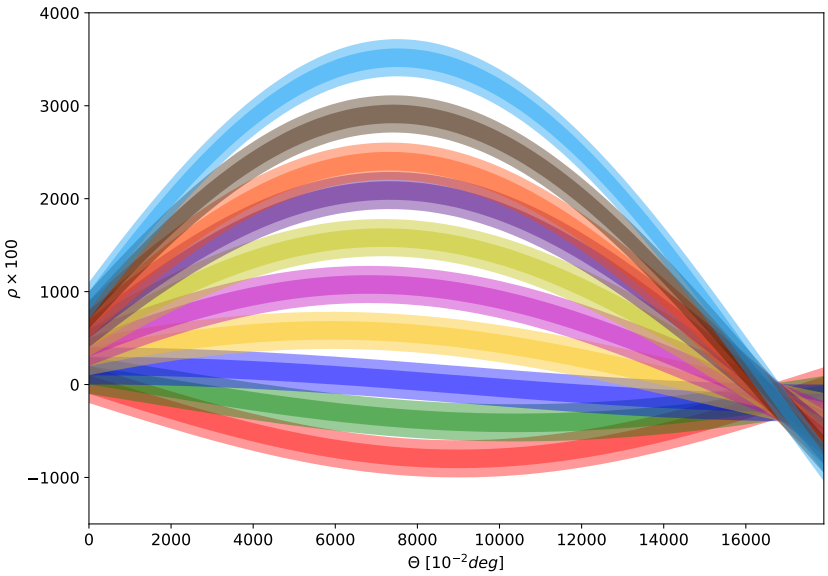


Figure 6.4: Example of BBAs construction for line estimation in the accumulation space. Every consonant BBA (one for each color) represents the information conveyed from a data point.

Figure 6.4 shows several consonant BBAs (one for each data point) represented by polygonal approximations of sinusoidal bands in the accumulation space. Given N points, the N BBAs are then combined following some combination rule into a single BBA. The solution (ρ^*,θ^*) is finally obtained by performing BetP maximization on the output BBA. Since the result of the maximization is in general a closed area, and not a single point, the barycenter of the output polygonal result is considered as the proposed solution.

We test this evidential approach on simulated data, were T = 100 lines are randomly drawn in the xy space, and M = 10 points for each line are extracted at fixed x locations, under Gaussian noise assumption for the y coordinate $y_i \sim \mathcal{N}(-\frac{\cos\theta}{\sin\theta}x_i + \frac{\rho}{\sin\theta}, \sigma)$. Moreover, in order to evaluate its robustness to unreliable sources, the system is tested for different numbers of outliers (0, 1 or 2), where an outlier is uniformly selected from the existing points and its y value is shifted by a constant value y_0. In the proposed experiments, the BBAs parameters are set as $\delta\rho_1 = 2$ and $\delta\rho_2 = 6$, and the noise parameters are set as $\sigma = 0.5$ and $y_0 = 5$.

The following different combination rules are evaluated: conjunctive rule, cautious rule, q-relaxation (with $q = 1$ and $q = 2$). The results obtained are compared with baseline least squares (LS), which is by definition the optimal estimator in presence of Gaussian noise. Figure 6.5 shows some line estimation examples extracted directly from the simulated data used for the quantitative evaluation.

Figure 6.6 shows the line estimation error distribution of the different methods under varying

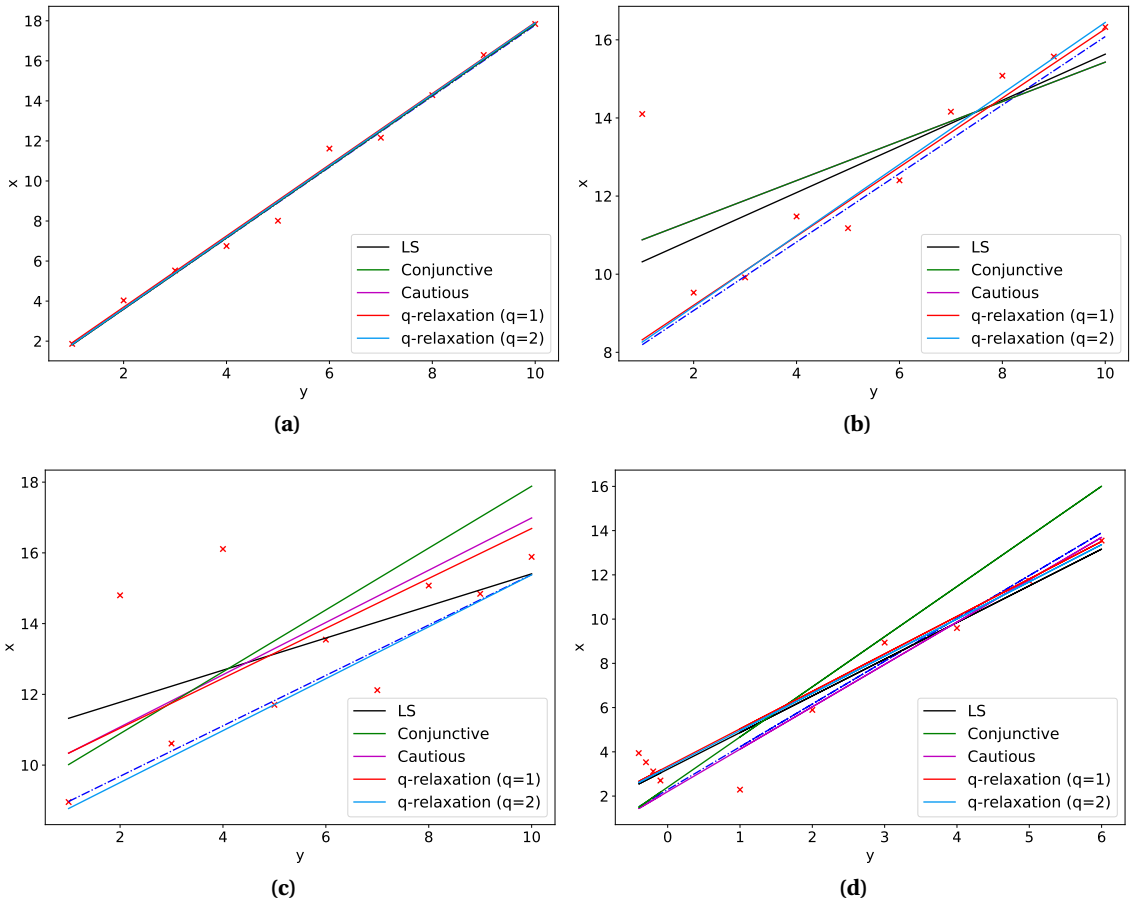


Figure 6.5: Example of line estimation results for different numbers of outliers: (a) no outliers; (b) one outlier (conjunctive and cautious rule lines are identical); (c) two outliers; (d) line estimation in presence of correlated source data. In (b)-(c) q-relaxation with q greater than the number of outliers outperforms the alternative approaches, in (d) the cautious rule outperforms the other methods.

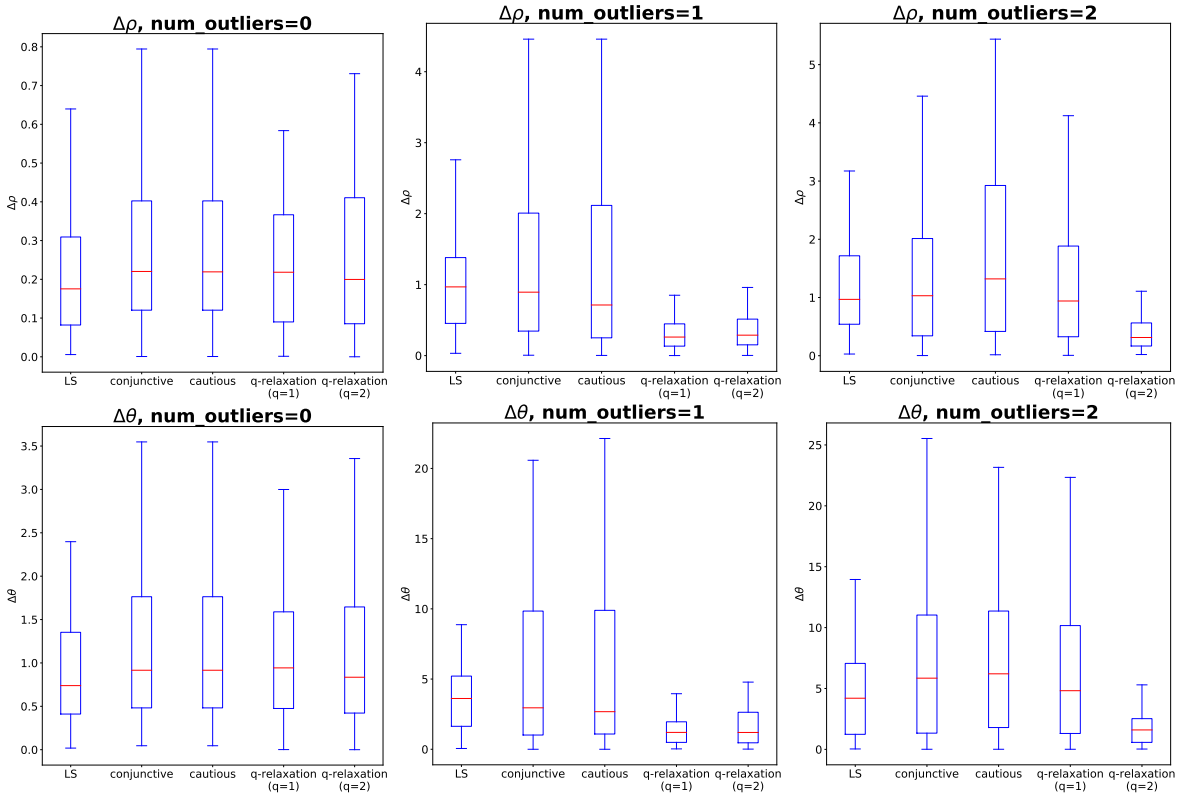


Figure 6.6: Radius error $\Delta\rho$ (first row) and angular error $\Delta\theta$ (second row), in presence of 0, 1 or 2 outliers (from left to right), for the experiments on simulated data for the line estimation toy example. In each subfigure, from left to right, the bars correspond to: least squares (LS), conjunctive rule, cautious rule, q-relaxation ($q = 1$), q-relaxation ($q = 2$).

conditions, in terms of $\Delta\rho = |\rho^* - \rho_{gt}|$ and $\Delta\theta = |\theta^* - \theta_{gt}|$, where (ρ_{gt}, θ_{gt}) are the ground truth parameters for one line of the simulation dataset.

The performance of the various combination rules when the data is free from outliers are comparable with the optimal performance, in terms of the median error, achieved by standard LS. In such context, conjunctive and cautious rules give the same results, which is a desired property when the sources are independent. In the case of q-relaxation, even though the number of unreliable sources is overestimated, the results are in line with the conjunctive rule case. When one outlier is introduced in the simulated data (by shifting the position of a random existing point), the advantage of a robust combination rule becomes evident. The conjunctive and cautious rules (which keep having comparable results with respect to each other) perform slightly better in terms of median error than the LS criterion, thus suffering less from the presence of an outlying source due to their resiliency to unreliable sources. Conversely, with respect to LS, the error distribution is more diffuse, reflecting a higher imprecision in the parameter estimation. The q-relaxation approaches ($q \in \{1,2\}$) clearly outperform the other methods both in median and variance of the errors, being able to filter out the unreliable source and perform the estimate with the inlying ones. The q-relaxation with $q = 2$ offers comparable performance to the one with $q = 1$ (which is the optimal choice in this scenario), with a slightly more imprecise estimate, given by the fact that it considers as unreliable both the outlier (possibly) and an inlier, reducing the amount of useful information exploited for decision making. When a second outlier is added, as expected, the q-relaxation with $q = 1$ becomes insufficient producing results with are only slightly better with respect to classical conjunctive rule, while q-relaxation with $q = 2$ still outperforms the others. The proposed example demonstrates the interest of q-relaxation for any 2D problem with ouliers (e.g. localization with GPS data), at the expense of a careful selection of the q hyper-parameter, as a trade-off between temporal performance and degree of robustness.

While the cautious rule is equivalent to the conjunctive rule for the proposed experiment in

the case of independent data, we show its benefit when the source independence assumption fails. Figure 6.5d shows an example of line estimation where some data is clustered in the xy space. Some of the drawn points exhibit a partial correlation in both their x and y coordinates: they are clustered in a small subsegment of the x axis, while their y coordinate being drawn from the same distribution. Thus, the derived BBAs are not independent. Moreover, the number of points composing the cluster represents a non-negligible percentage of the total number of points (40% in the example). In this case, LS clearly fails because univariate Gaussian noise assumption is violated, q-relaxation fails because it factors out few outliers, but it is still attracted by the rest of the cluster. Conjunctive rule fails because, when aggregating all the votes in the accumulation space, the cluster masses accumulate giving a strong weight to the estimation of the line from which they are drawn. Since the conjunctive rule is sensitive to the cluster size, it behaves estimating a wrong solution which tries to average the two lines. Conversely, cautious rule estimates the correct line accurately, because it processes the cluster of dependent points as a whole, thus not being influenced by the number of points composing it.

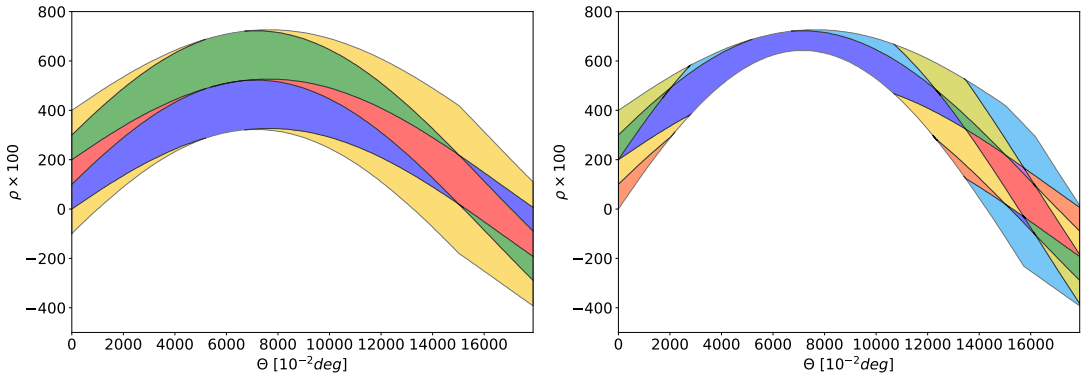


Figure 6.7: Example of disjoint sets decomposition on complex BBAs (obtained by iterative cautious combination of source BBAs); (a) after one combination; (b) after 2 combinations.

Figure 6.7 shows the disjoint set segmentation for efficient canonical decomposition (estimation of the ad-hoc 1D discernment frame, see Section 6.3.3), for a generic BBA as the one obtained by iteratively combining sinusoidal polygons. Such illustration points out that the computation of these disjoint sets is non trivial in general, while producing a segmentation of the discernment frame which is extremely convenient for canonical decomposition. In the presented scenario, since the starting BBAs are consonant, their canonical decomposition can be trivially computed as a special case at initialization time, and propagated after each cautious rule application as by definition. However, our aim here was to check the efficiency of our approach.

6.5.3 Case study: pedestrian tracking

We apply the proposed representation to the problem of tracking pedestrians detected by imprecise sensors, on the ground plane. The belief function framework allows for direct modeling of the imprecision associated with the detections and the tracks and provides a measure for data association between detections and tracks.

We make use of the detector proposed in [121] (see Chapter II.3), which performs low level information fusion from multiple cameras in order to provide a dense pedestrian detection map, together with pedestrian height estimations, in a range between 1.4 m and 2 m. The output of the detector allows us to project and track the detections on the ground plane. We demonstrate the use of the *2D polygon* representation provided in the *2CoBel* library in order to perform joint multiple target tracking in the *Sparse* sequence presented in Chapter II.3. We perform tracking on the provided detections for 200 frames of the *Sparse* sequence, and we measure the localization error of the real tracks (13 pedestrians, among them 4 standing and 9 moving) with respect to the

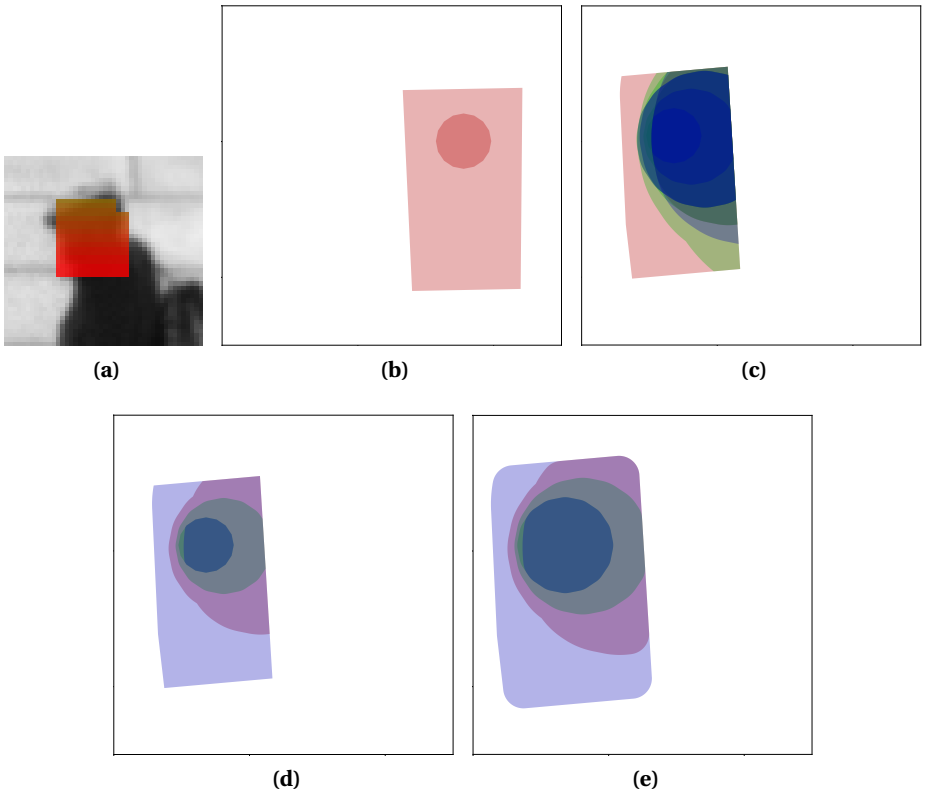


Figure 6.8: Example of pedestrian tracking steps. (a) Pedestrian detection blob. (b) Focal elements of detection BBA m_{d_0} on the ground plane at $t = 0$ (the size of the largest focal element is approximatively 1×2 square meters). (c) Focal elements of the conjunctive combination $\widetilde{m}_{t_0,7}$ between the track and the associated detection at $t = 7$ (16 focal elements). (d) Focal elements of the BBA simplification of $\widetilde{m}_{t_0,7}$ with the Jousselme's distance criterion (5 focal elements). (d) Focal elements after dilation of the track BBA $m_{t_0,8}$ by polygon offsetting.

ground truth. The tracker has to reconstruct lost tracks for given mis-detections occurring for up to six consecutive frames on the same pedestrian.

Discernment frame definition

The area under analysis is the ground plane region where the field of views of the cameras overlap. The area of the analysis region is 330 m^2. The algorithm is run at a resolution of 10^{-4} m, so that the cardinality of the discernment frame is $|\Omega| = 33 \times 10^9$. While the desired localization precision is 10^{-2} m, the chosen resolution is higher for increasing the robustness to rounding errors when performing clipping operations on integer vertices.

BBA construction and assignment

Given a detection d_i at time t located in (x_i, y_i), we build a consonant BBA with two focal elements. The first focal element is a disk centered at (x_i, y_i) and with a radius of 20 cm, taking into account the person's head and shoulder occupancy on the ground plane; the second focal element is a ring sector (approximated by a trapezoidal shape), which embeds the height uncertainty (on the direction point towards the camera location) and the camera calibration imprecision. In order to break the symmetry, the two focal elements are not assigned with 0.5 mass each, but with 0.51 for the internal disk and 0.49 for the trapezoid. In the presented case the choice of the mass allocation has a negligible impact on the quantitative results. As a future extension, the uncertainty of the head estimation which is output from the detector, could be used for BBA assignment.

Data association and combination

Given a set of tracks at time δ, $\mathscr{T} = \{t_1, \ldots, t_k\}$ and a set of detections $\mathscr{D} = \{d_1, \ldots, d_h\}$, the data association aims to compute an optimal one-to-one association solution $\mathrm{A}_l = \{(t_i, d_j), i \in \{1 \ldots k\}, j \in \{1 \ldots h\}\}$ with respect to some defined cost. One $(t_i, \emptyset)$ association means that the track is into an inactive state (so it keeps propagating until it associates with a new detection or dies), while one $(\emptyset, d_j)$ association means a new track has to be initialized with detection d_j. We make use of the criterion in [136] to define the association cost:

$$\mathrm{C}_{t_i,d_j} = -\log\left(1 - m_{t_i} \textcircled{\cap} m_{d_j}(\emptyset)\right),$$

which expresses the data association task as a conflict minimization problem, which can be solved by the use of the Hungarian algorithm [73, 116].

The data association task is followed by a conjunctive combination which produces for every (t_i, d_j) the new track:

$$\widetilde{m}_{t_i,\delta} = m_{ti,\delta} \textcircled{\cap} m_{d_j} \textcircled{\cap} m_p,$$

where m_p corresponds to the prior. It performs a masking operation on the visible region of interest of the cameras on the ground plane.

BBA simplification

A BBA simplification step is essential in tracking applications for two different reasons. First, we want to avoid that the number of focal elements grows without control as the time progresses, because it would mean that the real-time performance of the algorithm would degrade in time, bounding the maximum number of processed frames. Second, we want to avoid an excessive fragmentation of the belief. The BBA simplification aims at reducing the number of focal elements of a given BBA while respecting the least commitment principle. We adopt the method proposed in [5], which chooses iteratively two focal elements to aggregate (by performing an union operation) as the ones which minimize the Jousselme's distance [74] between the original BBA and the summarized BBA, i.e. the one obtained after the aggregation.

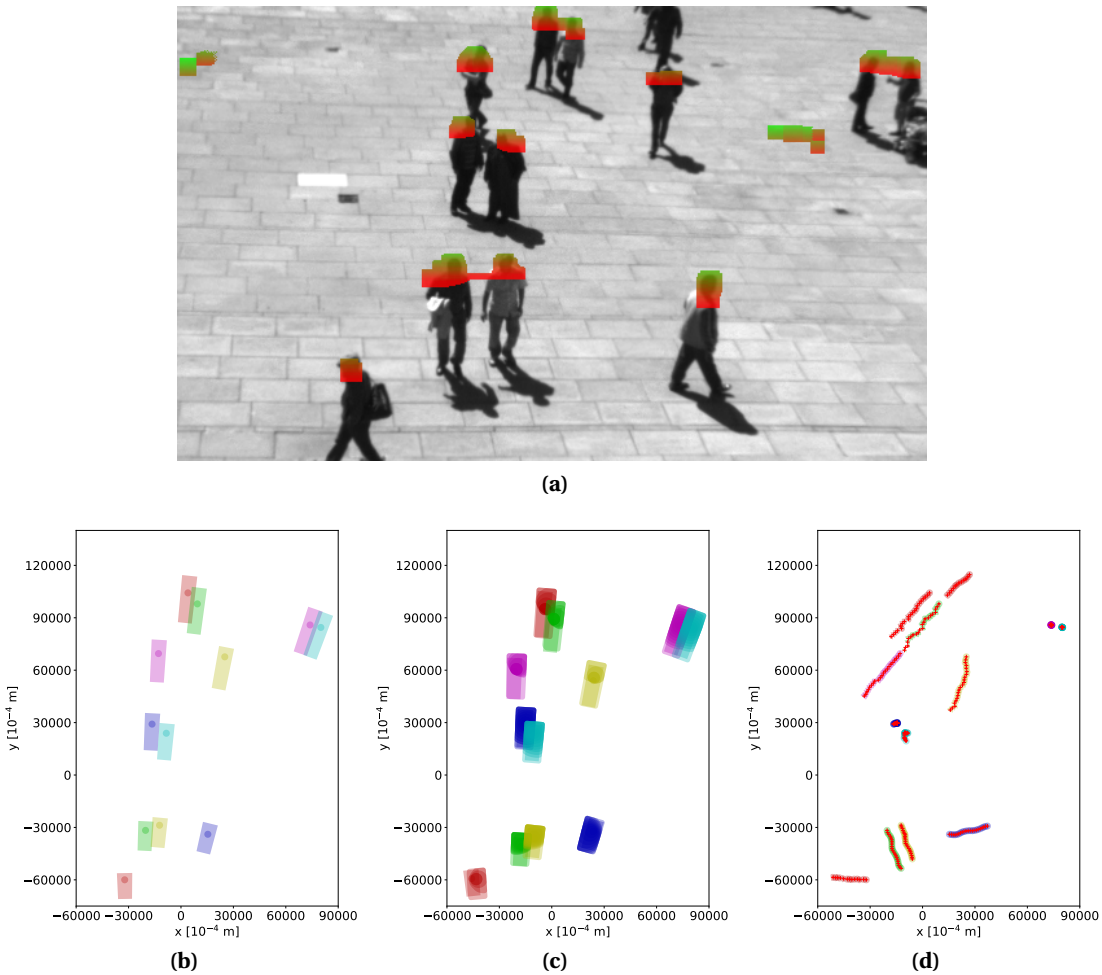


Figure 6.9: Pedestrian tracking. (a) Detection blobs on the image space ($t = 0$) estimated by the detector in [121]. Colors refer the estimated height values from 1.4 m (red) to 2 m (green). (b) Focal elements of the detection BBAs on the ground plane ($t = 0$). (c) Focal elements of track and detection BBAs on the ground plane ($t = 8$). Associated tracks and detections share the same color. (d) Final estimated tracks on first 20 frames. Red crosses refer to target locations, while colored sets correspond to regions presenting maximum Bet$P$ value.

The proposed representation allows, conversely to the one in [5] (which simplifies the BBA after each conjunctive combination), to perform the simplification on a less frequent time step. In the proposed experiment a target BBA is simplified when it reaches 15 focal elements, by producing a 5 focal element BBA.

BetP **maximization**

At each time step, we run the BetP maximization algorithm presented in Section 6.4 for each active track $\widetilde{m}_{t_i,\delta}$ in order to extract the most probable location of the target. The cardinality of the resulting polygon represents the irreducible ambiguity in the target location. The target position is then estimated as the barycenter of the polygon.

Modeling the imprecision of the tracks prediction

Given the track $\widetilde{m}_{t_i,\delta}$, which represents the result of the conjunctive combination, we need to model the imprecision of the prediction step. In order to model the track displacement from the current location, a random walk term is added to the track. Such term boils down to an isotropic dilation of the focal elements. In the proposed representation, this corresponds to applying a scalable polygon offsetting algorithm, having $O(n \log n)$ complexity, where n is the number of vertexes. Polygon offsetting allows for a dilation which respects the inclusion relationship of the original focal elements. The result of such step is the predicted track $m_{ti,\delta+1}$ at time $t+1$.

Figure 6.8 depicts an example of the proposed tracking steps for a single pedestrian, specifically the BBA geometric representation after construction, combination with the previous track, simplification and offsetting.

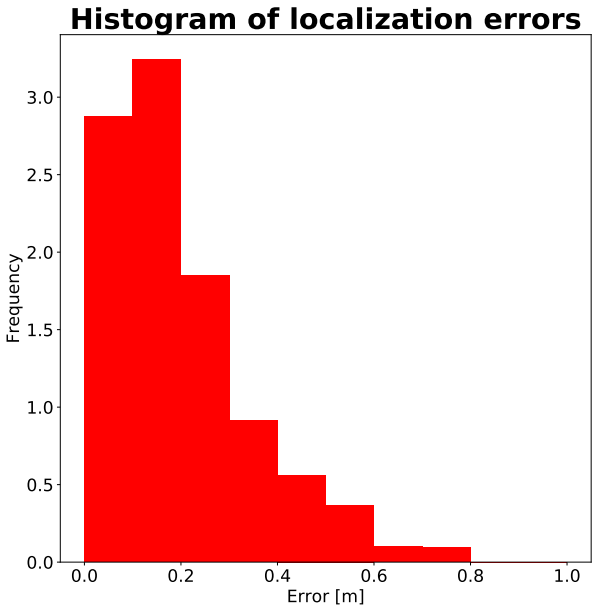


Figure 6.10: Normalized histogram of the localization error of pedestrian tracking on the *Sparse* sequence.

Results

Figure 6.9 show some qualitative results of pedestrian tracking in the *Sparse* sequence, highlighting the tracks estimated after the first 20 frames. In order to evaluate quantitatively the tracking accuracy, the target predicted locations are compared against an available ground truth. Such ground truth consists into coordinates in the image space where the heads are located. Since the height of such individuals is not known a priori, each location in the image space projects to a segment in the ground plane, allowing for any possible height in the interval of study. One computes the localization error as the distance between the target estimated location, and the ground truth

Resolution	Average Localization error
$10^{-1}\ m$	30.197 cm
$10^{-2}\ m$	22.340 cm
$10^{-3}\ m$	20.078 cm
$10^{-4}\ m$	19.944 cm
$10^{-5}\ m$	19.931 cm

Table 6.4: Average localization error on the *Sparse* sequence using different discretization resolutions. By using a representation able to deal with finer resolutions, one may achieve a significant performance gain.

head location, under the assumption that the height of such head corresponds to the predicted one. Such metric corresponds to computing the distance between the ground truth segment and a height uncertainty segment drawn at the target location. Target locations for inactive track states are estimated by linear regression fit of the estimated target positions at previous states.

Figure 6.10 shows the results in terms of (normalized) histogram of localization error. The average localization error is $\epsilon = 0.2\ m$, which reaches the empiric limit set by the intrinsic uncertainty of head spatial occupation. On the other hand, the average localization error remains steady in time, meaning that the estimated tracks do not tend to drift away from the real ones. The standard deviation of the average localization error in time is $\sigma = 2.3\ cm$.

Table 6.4 shows the average localization error obtained by the tracking algorithm for different choices of the resolution at which the discernment frame is discretized. When a coarse resolution of 10 cm is considered, the performance drops consistently. At this resolution the size of the discernment frame is already large enough to be intractable using methods based on binary representations, as in [5]. Moreover, while for the theoretically desired resolution of 1 cm the average localization error consistently drops, the proposed representation allows us to scale at finer resolutions to account for rounding errors, thus providing an additional performance boost.

For more complex tracking scenarios, the next step is to integrate the proposed representation in a more sophisticated model such as [160], which is supposed to cope with specific issues such as disambiguations or long term occlusions, and where our approach would extend box representations.

6.6 Conclusion

In this work we have proposed a new representation for multi-modal information fusion in 2D spaces in the BFT domain. Such representation exhibits uniqueness, compactness, space and precision scalability, which make it suitable for many settings constrained to large hypothesis spaces, where there is the need to extend the Belief Function framework with efficient multidimensional operators. In our experiments with actual data, we show the effectiveness of this formulation on multi-target tracking scenarios, where tenths of tracks have to be estimated on a wide region of interest. The main contributions can be summarized as follows:

- The proposal of a new polygon-based compound hypothesis representation, able to benefit from fast polygon clipping and hashing algorithms for scalability.
- The definition of an intersection-inclusion directed acyclic graph to model the interaction between focal elements.
- The outline of efficient algorithms for the fundamental operators, decision making and decomposition methods which fully exploit the potential of both the geometric and graph representation proposed.

- The release of our contribution as a public library for the community, in order to ease the reproducibility of such representation for active research.

Part IV

Single Camera Supervised Pedestrian Detection in Dense Crowds

Chapter 7

Supervised Pedestrian Detection in Dense Crowds: an overview

Contents

7.1 Motivation

In the previous chapters, we have introduced novel methodologies for performing multiple camera detection without supervision, for modeling the imprecision and uncertainty of such detections, and for allowing spatio-temporal fusion among detections from different sources.

The pedestrian detector presented in Part II, does not take into account the appearance of the objects it aims to detect. Such detector can thus be used as a complement to some supervised appearance-based detector, in order to enforce the 3D occupancy/location of each pedestrian, and to provide reliable detections when the extreme clutter makes the appearance-based detector fail. On the other hand, an appearance-based detector is crucial for any high performance detection task, even if it is agnostic about the 3D structure of the crowd. For such reason, in this Part we will focus on the single camera supervised detection problem, we will highlight its challenges in dense crowds, and we will exploit it in order to detect pedestrians independently on each view. Such new sources of information will allow for spatial information fusion at ground plane level, by integrating also the unsupervised detector of Part II.

The analysis of pedestrians in dense crowds with single cameras is a relevant topic in computer vision. The performance of crowd analysis on single cameras is critical both for scenarios where no camera network is available (or where some constraints negate the placement of multiple cameras), and as a foundation for multiple camera reasoning in the 3D space. Several works for multiple camera detection and 3D tracking [3][79] rely on pedestrian silhouette extraction in each independent view prior to estimating the 3D location of each pedestrian. Hence, the multiple cameras can be later exploited to enforce the robustness of the single detectors.

From a methodological point of view, the detection task in crowds is inherently different from the counting problem, which tries to estimate how many pedestrians are present in a crowded area without inferring their actual accurate position. The approaches employed for counting can be split in two main branches: counting-by-detection and counting-by-regression. In counting-by-detection [47] [69], first a detector is used to produce a confidence map of pedestrian presence, and then the counting stage will trivially return the number of confidence peaks in such map (with

some non maxima suppression). In recent years, counting-by-regression methods have risen up since the community has proposed alternative approaches to avoid performing counting by needing to solve the more difficult detection problem, by learning regression functions through locally extracted features, for instance by regressing on estimated density maps [174] [142].

However, for many tasks, counting is not enough in dense crowds, especially when pedestrian tracking is involved into the process, for example for studying and predicting pedestrian movement through learning their social behavior with, e.g. deep generative models [53]. Such methods, which rely directly on given track positions, need as accurate pedestrian location as possible in order to perform prediction at a microscopic level.

7.2 Related works

In the context of human detection, several works explore the use of robust hand-crafted detectors for solving the task. In the context of high density crowds, solutions which rely on local appearance cues, as color histograms, or common face detectors such as Viola-Jones [166] are unsuited, since pedestrian faces are not detailed enough.

Related to the image gradient, the Histogram of Oriented Gradients (HOG) descriptor [25] is very popular and has exhibited good performance in various contexts. The HOG descriptor has been applied in various works which try to handle pedestrian detection at higher densities, by detecting only the head of the pedestrian [9], or by using a part-based human detector with occlusion estimation for each part [28].

In high density crowds, while the contour related to the specific shape of the head and shoulders is indeed highly discriminative, it may fade away due to clutter. For this reason, shape-based descriptors are usally combined with others which encode different characteristics. Traditionally employed in texture classification, the Local Binary Pattern operator [119] has been successfully used in pedestrian detection due to its reasonable robustness to occlusion provided by its local sampling strategy. As an alternative one may use covariance matrix based descriptors [68], but their representation is less compact and the computational cost is much higher. In the category of texture representation, Gabor filter banks have been used for head detection [91] to encode the local frequencies and orientations.

Many works have proposed novel methods for the robust combination of multiple pedestrian detectors each employing a different feature descriptor (or combination of descriptors), most notably in the evidential framework [175] [164].

The recent work in [67] proposes a Convolutional Neural Network (CNN) based method in order to detect faces even at extremely small scales. Such approach re-scales the input with various scale factors, and feeds each image into a shared CNN. The response maps are then merged at the original resolution in order to get the final detections. While this method has been proven to be robust to low resolution textures, blur, and partial occlusion, it is still targeted to the face detection task, which is only a sub-problem of head detection, since a head has extreme appearance variations with change of perspective.

In the following, we will focus on CNN architecture models, and we will study their applicability to the task of head detection under severe constraints.

7.3 Convolutional Neural Networks (CNN)

CNNs solve the problem of regular neural networks of scaling for image inputs. Since in common neural networks the input nodes are fully connected with the nodes at the first hidden layer, such number of connections can become extremely high when the input nodes represent pixels of an image, thus not scaling well at higher image sizes. CNNs exploit the prior knowledge that the inputs are images to reduce the number of parameters in a clever way: they use the same copy of a feature descriptor for different positions of the image (weight sharing). In such way, the parameters to be learned are largely reduced, and their number is independent from the input size.

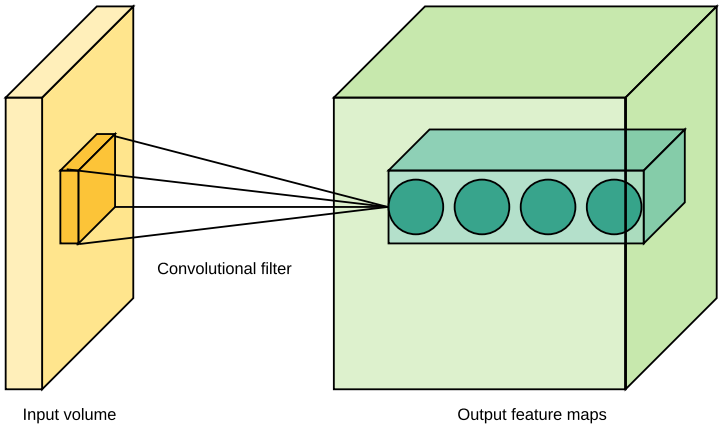


Figure 7.1: An example of convolutional layer. Given an input volume, each node of the layer is connected to a local region of the input module along the input depth. Each local region is connected to multiple neurons (4 in the example above), one for each output feature map. The neurons of each depth slice (i.e., the neurons forming the same output feature map) are connected to their corresponding local region through the same weights (taking the shape of a convolutional filter). The figure is inspired by [76].

The neurons of a CNN are three-dimensional, where the third dimension defines the number of different representations that must be learned for each image patch. Note that the usage of replicated feature detectors does not make the neuron activations invariant to translation, but it makes them equivariant with respect to the input (a translation in the input amounts to an equivalent translation of the activation).

In the following subsections we will shortly introduce the main layers which constitute a typical CNN. For a detailed survey of different convolutional deep learning architectures, and their performance and spatio-temporal efficiency, we refer the reader to the comprehensive survey in [15].

7.3.1 Convolutional layers

The convolutional layers are the basic bricks for any CNN model. The weights of a convolutional layer are sets of, usually small, filters. The dimension of such weights is W × H × F, where W and H are the width and the height of the filter (along the image dimensions), and F is the number of filters. The input image is processed, during the forward pass, by performing a convolution operation with each filter, thus F feature maps are obtained as output (see Figure 7.1). The extent of the image patch which is connected to the filter of a convolutional layer, is also called *receptive field*, i.e. the region of the input that a particular feature is looking at. Successive convolutions have the effect of enlarging such receptive field, thus capturing, at deeper stages, more complex features which depend on the global context.

Each convolutional layer has three important hyperparameters which have to be set: the number of filters (also referred as *depth*), the **stride**, and the **zero-padding**. The stride is the step at which the filter slides. Regular convolutions have a stride equal to one, but larger strides can be used, and this has the effect of reducing the size of the output feature maps. The zero-padding is used as a convenience mean in order to adjust the input size in order to control the spatial size of the output. For example, when one wants to maintain the same dimensionality from input to output, one can adjust the padding as a function of the other two hyperparameters in order to reach the scope.

Convolutional layers apply linear transformations to the input, thus, in order to introduce some non linearity, it is common practice to combine them with some activation function. There are several activation functions proposed in literature, e.g. sigmoid, tanh, and the Rectified Linear Unit (ReLU) and its variations. In practice, ReLU variants are used as activations, since they have a great impact into accelerating the convergence of stochastic gradient descent [84], and due to the

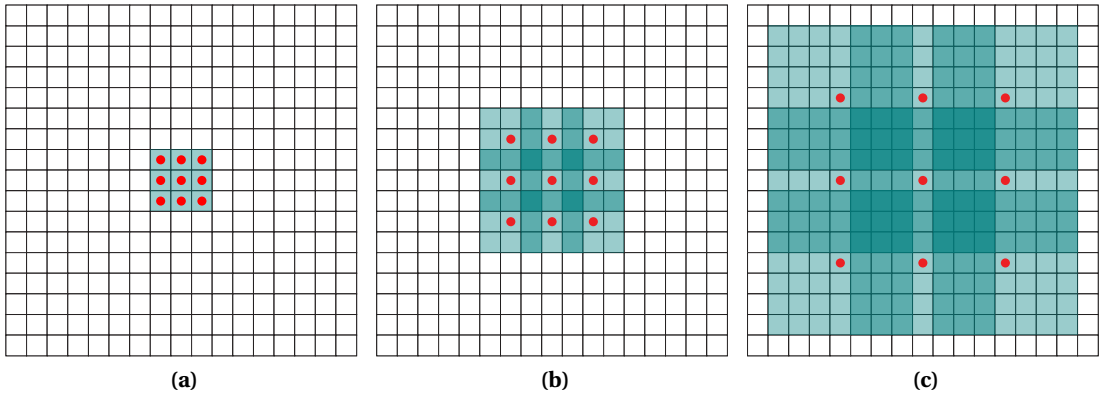


Figure 7.2: Example of how the use of dilated convolutions can exponentially enlarge the effective receptive field, while linearly increasing the number of parameters. Red dots specify the cells where the filter is applied, while green cells highlight the receptive field. Let us call F_0 the set of input elements. The receptive field of an element p in F_i is the set of elements of F_0 which contribute to modify the value of $F_i(p)$ [180]. (a) F_1 after a 1-dilated convolution of F_0 (3×3 receptive field). (b) F_2 after a 2-dilated convolution of F_1 (7×7 receptive field). (b) F_3 after a 4-dilated convolution of F_2 (15×15 receptive field). Image taken from [180].

drawbacks of other functions (e.g. vanishing gradient problem for sigmoid and tanh).

The combination of convolutional layer and activation one can be further enriched by the addition of some normalization layer, e.g. batch normalization [70], which is usually applied between the output of the convolutional layer and the input of the activation.

Dilated convolutions

Recent works [180] have proposed to introduce a fourth hyperparameter to the convolutional layer structure called **dilation**. The dilation allows us to have filters that are applied to spaced input cells. The dilation corresponds to the amount of spacing between cells (we will refer to dilation equal to one for contiguous input patches).

Dilated convolutions have been proposed with image segmentation in mind, in order to be able to aggressively enlarge the receptive field with fewer layers. In fact, when applying successive dilated convolutions, one can exponentially enlarge the effective receptive field without loosing resolution (e.g., with downsampling layers), thus aggregating faster the contextual information.

7.3.2 Pooling layers

The pooling layers have the primary objective to reduce the spatial size of the feature maps, in order to decrease the computational load on the network, thus allowing for more feature maps in the successive layers.

The pooling layer operates on each feature map independently and reduces each feature slice to a single value. The spatial extent (size of the slice to pool) and the stride are hyperparameters of the layer. For example, a pooling layer with both spatial extent and stride equal to two will produce output feature maps which are halved on each dimension, thus having one fourth of the input size.

Pooling can use different operations in order to perform the reduction (max, average, etc...), with the max operation being the most common, since it has been shown to work better in practice.

Pooling can be a problematic operation in case when clear spatial relationships between different parts of the input must be kept. In fact, several pooling operations can destroy the information about the precise localization of such parts.

Having presented very succinctly these preliminary notions and building blocks used extensively in deep learning nowadays, we can proceed in the following chapter with presenting the specific adaptations for our architecture.

Chapter 8

Pedestrian map computation with Convolutional Neural Networks

Contents

8.1 Problem formulation

We turn the problem of detecting heads in a dense crowd into a semantic segmentation problem, which in our case may be defined as follows. Given an input image, we aim to estimate a binary map of the same size as the input, where pixels belonging to the heads are labeled as foreground.

Such specific problem is positioned halfway between two main applications of image segmentation: natural images segmentation (e.g., for urban scenes understanding) and medical image segmentation (e.g., for finding cell nuclei). On one side the images represent real world scenery, so the background is often rich in terms of texture, while on the other side the objects to detect are small and with poor texture information, as it happens for nuclei in medical images. Thus, a network for segmentation should learn how to distinguish cluttered objects in a rich environment.

8.2 Data acquisition and augmentation

Let us consider an ideal video-safety scenario, where cameras are deployed in order to perform real time analysis of a sensible area during an event. Each urban environment setting could lead to a different distribution to learn with respect to others, thus needing to acquire data in place for full model training or transfer learning. However, access constraints to the site could prevent deploying the system a long time before the event, and, even if so, using data of the scene when too few people are present could provide insufficient or unsuitable data. Moreover, again due to access constraints, when checking the system most (or all) of the data could come from a single

camera, and it could be difficult in general to train the entire system with data from multiple views. Finally, the data labeling process has a cost which is significant in terms of human effort, which is always a factor which needs to be taken into account.

For such reasons, the network choice and the training preparation has to be done with the idea in mind that few training data may be available which represent the distribution that one wants to learn for the analysis of the event. Thus one has to exploit the available images as efficiently as possible in order to perform a robust training while preventing overfitting.

The semantic segmentation task differs from the classification task (assigning a unique label to the entire image) in the sense that each pixel of the image carries a piece of information for the class it belongs to, while in classification the image as a whole carries information on the object class. Thus higher the spatial extent of each image, the more information is carried out. However, severe imbalance of class labels must be taken into account. Consider an image with a single head present and the rest background. Such image carries imbalanced information because the network is trained with much more negative than positive pixels. Thus, the crowd density at the training stage has to be used as an indicator in order to take into account such imbalance.

A powerful technique for coping with small training datasets and to prevent network overfitting is called *data augmentation*, which consists in creating artificial new data starting from the available one. Such technique is so successful into improving the invariance and robustness of the network to various conditions that it is always used in combination with even large datasets. The main idea under data augmentation is that the available data has been acquired under a limited variety of conditions, for example orientation, illumination, noise, scale etc. Thus we synthetically modify the available data in order to account for such variability, thus making the network more invariant when detecting objects in a different context than the ones encountered during training. The specific kind of augmentation to apply is dependent on the task of the network, since any augmentation should be coherent with meaningful changes in the actual data (e.g., it is not worth to flip vertically a crowd image). Some augmentations, for example, have been created for the specific problem under analysis, for example landmark perturbation in the face recognition problem. For our problem, we augment the data with horizontal flips, Gaussian noise, salt and pepper noise, brightness and contrast change. Note that augmentation sources are applied randomly at each epoch for a specific image, so only a subset (or maybe none) of them are performed together. Since we are dealing with a semantic segmentation problem, any data augmentation which perform an affine transform or a flip of the original image must be applied in parallel to the ground truth output map.

8.3 Learning with soft labels

In the context of a segmentation task, the ground truth maps are usually labeled by assigning the correct class pixel by pixel. The contour of the objects is clearly defined and such information is crucial in order to have a high fidelity segmentation. For the specific task of head detection in dense crowds such precise labeling is often impossible. The reason is that due to the extreme clutter and occlusion it is sometimes unfeasible for a human operator to clearly distinguish the contour of a head with respect to the background. Moreover, thousands of heads per image should be labeled, making the process cumbersome to complete.

For such reason, we inspect the problem of estimating head maps with partially labeled data, where only the center of the head is annotated (see Figure 8.1, and a prior knowledge on the average size of a head in pixels (or the size of each head with respect to its location in the image, in case of strong perspective variance) is available. While this labeling is much more efficient and avoids a clear definition of the borders, it can still be imprecise since it could be difficult for the human operator to precisely detect the head center. Ground truth errors of one or two pixels are always present in practice for any application, but for head detection the effective error can be important since the size of the head can be extremely small (e.g. 2 pixels displacement over a head of a 10 pixel diameter).

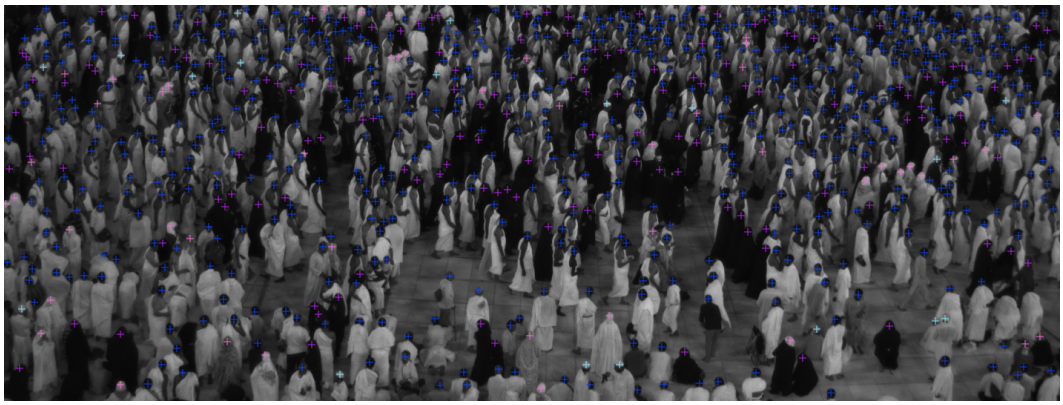


Figure 8.1: An example of the manual ground truth labeling on the *Mecca* dataset. Soft labels are used in order to annotate the presence of pedestrian heads. The different colors of the labels reflect the capability of the interface to be trained for multi-class problems, where pedestrians are distinguished in e.g., men and women. For the purpose of this study, all the annotations are considered belonging to the same class, thus making the problem binary.

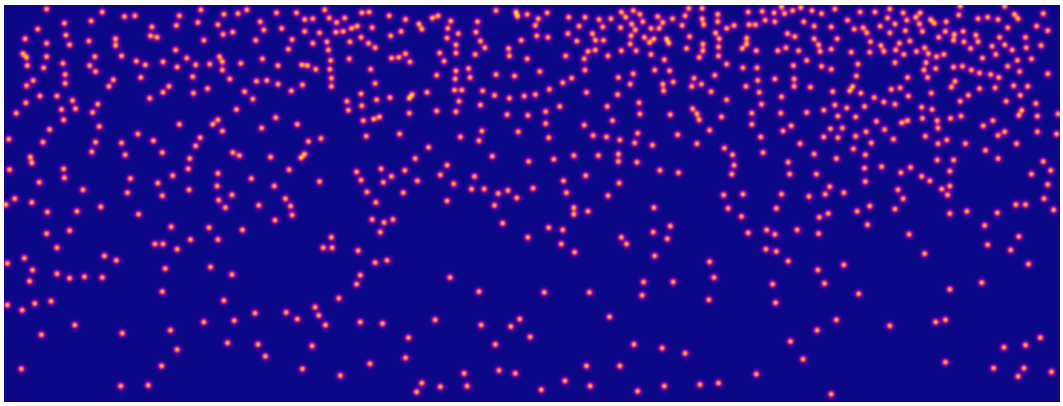


Figure 8.2: Pedestrian ground truth map as a sum of Gaussian distributions, one for each head. The score associated to each pixel is the sum of the contribution of each Gaussian at the given location (higher score from blue to yellow).

Let us consider the ground truth map for a single head. In the classical definition of segmentation, the head map would be a circular blob where pixels belonging to such blob would take a value of one. Such labeling expresses a strong discontinuity at the borders between the two labels, and thus requires a good knowledge of the head contour.

Conversely to this formulation, we consider that a head ground truth map has a distribution which expresses the probability of a pixel to be at the center of such head. Starting from the labeled head center location (x_c, y_c), the ground truth map for such head is expressed in terms of a Gaussian distribution as

$$(x, y) \sim w\mathcal{N}\left((x_c, y_c), \sigma_h\right), \tag{8.1}$$

where 2σ is the expected head radius, and w is a scaling factor, which will be clarified further in the section. The choice of a Gaussian distribution is justified by the fact that it is an infinitely differentiable function with the property of having tails which vanish at infinity, thus well modeling the uncertainty on the precise border location.

The final ground truth map will be the sum of Gaussian distributions, one for each head. Thus the map will not be a probability distribution by itself, and transforming it into a distribution via a global normalization would make the score associated to each pixel depend on the number of heads in the image. By avoiding the normalization, the score associated to each pixel represents the sum of probabilities that any head, occluded or not, is located at that position (the sum is due to the superposition of two or more possibly close heads). Figure 8.2 shows an example of ground

map from *Mecca* dataset soft labels.

The parameter w in Equation 8.1 is tuned in order to solve the imbalance problem between pixels belonging to heads and to background. Higher the w, higher the impact that each single pixel belonging to a head has in the loss function. Thus, it is equivalent to weight the loss for the positive class (as it is done in weighted cross entropy loss for the classification problem).

Regarding the choice of the loss function, a standard semantic segmentation definition allows us to turn the problem into a classification problem, where the class for each pixel has to be estimated, thus making it ideal for the use of a cross entropy loss function for training the CNN. However, in our case, the pixels are not labeled with their class, but with real values resulting from the combination of the head distributions. Thus, a Mean Square Error (MSE) loss, or L2 loss, is used, as a straightforward estimate of the distance between two 2D maps.

8.4 CNN architectures

In this section we will describe the different architectures which have been adapted for the head semantic segmentation task.

8.4.1 UNet

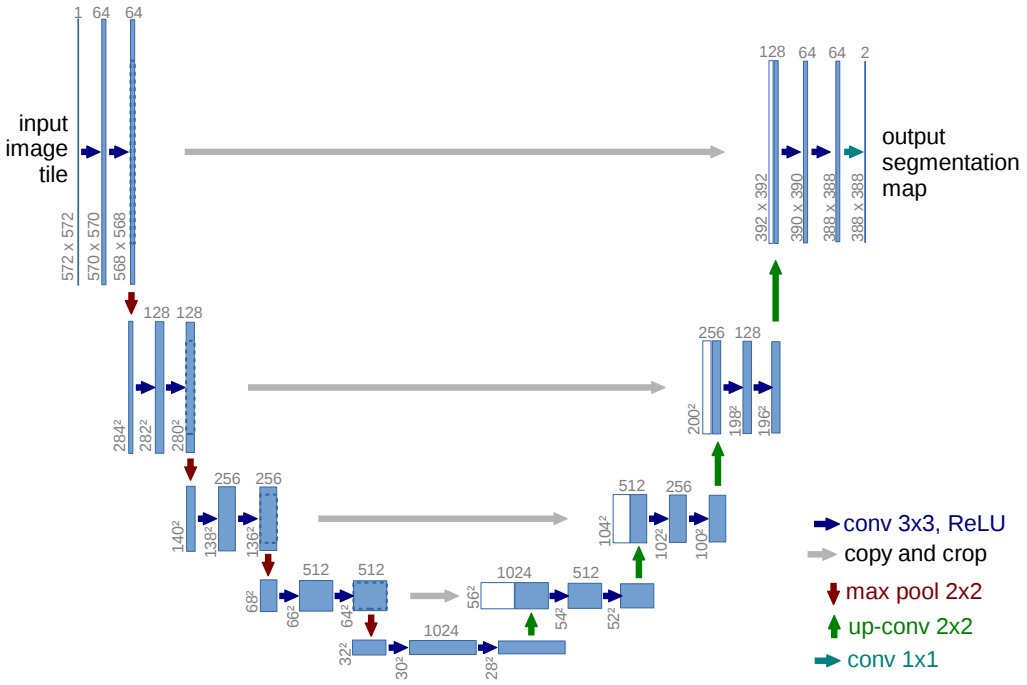


Figure 8.3: UNet architecture. The U-shape is given by a descending phase (encoding) for context extraction, and an upsampling phase (decoding) for output map reconstruction. The grey arrows represent the shortcut connections which result into the combination of upsampled reconstructions and feature maps.

The UNet architecture [137] is a state-of-the-art segmentation network which has been originally introduced for biomedical images. The network, coupled with aggressive data augmentation, makes an efficient use of the available training data, so it is capable to achieve good performance on small datasets, while being fast at test time.

The UNet extends the well founded fully convolutional network (FCN) for segmentation [99], which has been for long time the state-of-the-art architecture for semantic segmentation. The original FCN introduces the idea of using upsampling (bilinear interpolation or deconvolutional layers) of the lower resolution (because of pooling) feature maps, in order to reconstruct the output map. The authors propose also to insert shortcut connections between higher resolution features and the reconstructed map in order to improve localization.

The main property of UNet is to have a downsampling part for context extraction, and a symmetric upsampling part for localization which has a high number of feature channels, so that the context can propagate progressively to layers at high resolution. Figure 8.3 shows the U-shaped architecture of the network, with a contracting and an expansive path. Each downsampling (pooling) in the contracting path corresponds to a mirrored upsampling in the expansive path. After each upsampling, the interpolated maps are combined with the corresponding feature maps at the same resolution in the descending path. Such operation is essential to avoid producing too coarse output maps.

8.4.2 Using dilated convolution for segmenting small heads

The authors of [55] highlight a fundamental problem of semantic segmentation when the objects to detect are very small, and they are densely located. The use of pooling layers can gradually destroy the resolution, so details of small objects can be missed, and even the addition of shortcuts (as in the UNet architecture) could be not enough to recover the structure of the objects.

	Layers
Front end	Conv 3×3, F = 16, D = 1
	Conv 3×3, F = 32, D = 1
	Conv 3×3, F = 32, D = 2
	Conv 3×3, F = 64, D = 2
	Conv 3×3, F = 64, D = 3
LFE	Conv 3×3, F = 64, D = 2
	Conv 3×3, F = 64, D = 2
	Conv 3×3, F = 64, D = 1
	Conv 3×3, F = 64, D = 1
	Conv 1×1, F = 1, D = 1

Table 8.1: Detailed architecture of our adapted network, inspired by [55]. The parameter F indicates the number of filters, while D expresses the dilation factor.

In order to enlarge the receptive field without decreasing in return the resolution of the output, dilated convolutions can be exploited. If the dilation factor is linearly increased as we go deep in the network, the effective receptive field will exponentially grow, thus capturing a large context. However, when dealing with small objects, one cannot apply progressively larger dilations in a straightforward way, because, since dilation causes weights to skip information between cells, this could prevent the network from modeling well locally the head structure. For such reason, in [55] the authors propose a deep network without pooling layers which has a pyramidal application of the dilation factor: increasing dilation, as any straightforward use of dilation, and then a so called Local Feature Extraction (LFE) module, with a decreasing dilation factor.

The architecture used by the authors in [55] is a VGG front end module (enriched with increasing dilations), augmented with a LFE module, which keeps invariant the number of filters and the kernel size, and linearly decreases the dilation factor to one. The drawback of such kind of architecture, however, is that it needs much more memory than, e.g., the UNet, because the feature maps at each step have always the same size of the original input. The absence of max pooling makes the quantity of memory needed for the backward and forward passes much larger for the same number of parameters. Thus, in order to keep the memory use manageable, the number of filters at each convolutional layer has to be greatly reduced in order to resize the parameter space. For this reason, the original architecture proposed by [55] has been modified in order to fulfill hardware constraints (8 GB of video memory). Moreover, batch normalization is added on top of each convolutional layer of the network. Table 8.1 shows the details of the modified architecture.

8.5 Implementation details

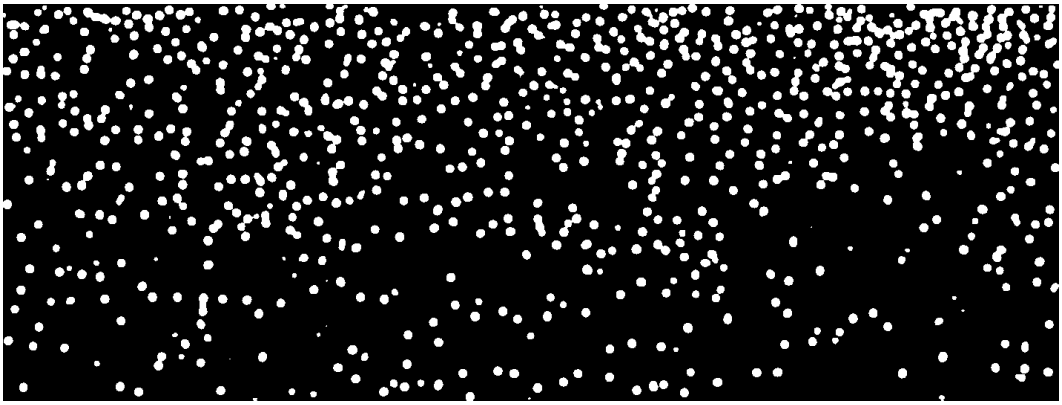
The two networks are trained on two different datasets: *Mecca* and *Regent's Park Dense*. The *Mecca* dataset consists of 35 training images (with approximately 300 heads per image) of the *Mecca* pilgrim crowd at an extremely high density. The *Regent's Park Dense* dataset, introduced in Chapter 4.3, consists of 140 training images (with approximately 40 heads per image). The training and validation sets (and, thus the separate test set) are sampled at a constant rate from a video stream at very different initial time frames. As regards crowd density, the *Mecca* dataset is much more denser than *Regent's Park Dense*, while the latter exhibits stronger variation in density in different locations of the image. This difference in density reflects into a higher data imbalance for *Regent's Park Dense*, which is solved by setting a higher weight factor w into the ground truth distributions.

The weights of the convolutional layers are initialized with the Kaiming He method [64], which is targeted for deep networks with ReLU activations. The network is trained by using an Adam stochastic optimizer [81] with a learning rate of 10^{-2} for UNet, and 7×10^{-3} for the network with pyramidal dilation (from now on, we will refer to such network as *FE+LFE*, since the architecture combine a front end structure with the LFE module). Early stopping with a patience of 20 epochs is used in order to terminate the learning process as soon as the network error stops improving on the validation set.

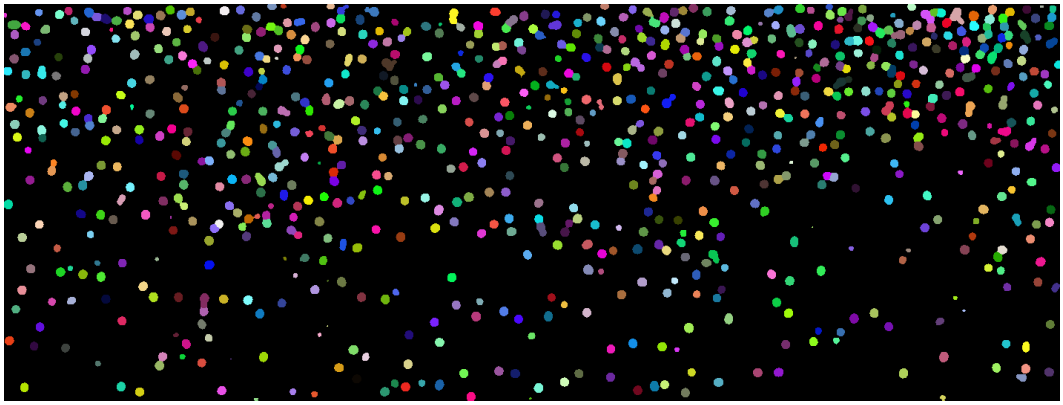
8.6 From semantic segmentation to instance segmentation

While the aforementioned networks solve a semantic segmentation problem, in order to detect heads having a uniform map of head occupation it is not enough. In fact, we want to distinguish pixels belonging to each individual head, thus turning the problem in an instance segmentation problem, where, beside estimating the mask, one estimates also the location of all the instances of the same class. While methods like Mask R-CNN [63] perform jointly mask segmentation and bounding box regression of each target, networks like UNet only provide the mask as an output. Some workarounds for this problem exist (for example perform joint estimation of the mask, and of the borders, by treating them as a different class), but the particular shape of the proposed ground truth extraction procedure gives a straightforward solution.

By estimating this cumulative map of Gaussian distributions, one has for free the division between the different heads. The computationally simpler tecnhique is to get the locations of the maxima of such distributions in the estimated map, and then to apply a watershed algorithm, originating from the mathematical morphology community [10], by using such maxima as seed. Each pixel score, negated, can be treated as an elevation, and thus the algorithm floods basins starting from each seed and, thus, assigns pixels to such basin until the edge of another basin is met.



(a)



(b)

Figure 8.4: (a) Example of pedestrian semantic segmentation inferred on a test image of the *Mecca* dataset. The pixels are either labeled as background (black) or head (white). (b) Instance segmentation derived from the semantic segmentation, by using the watershed algorithm on the peaks of the estimated head distributions. The different colors represent unique labels for each independent head.

8.7 Method evaluation

The various architectures are evaluated for the two datasets on their test subparts. Since the objective is to detect as many (and as precisely) pedestrians as possible, we use an object level metric for object detection for evaluation, rather than pixel level metrics for segmentation.

First of all, one has to recover the detections and then to check if it is either a false positive (FP) or a true positive (TP). Since the network estimates a distribution which is defined on all the domain, the extent of a detection can be indefinitely large, comprising pixels with arbitrary small values. In order to prevent this, each detection is constrained such that each pixel has a minimum predefined score η. Such operation is symmetric to, in a binary classification problem, one-hot-encoding the estimated classes by checking whether the output probability is higher or lower than 0.5. The value of such predefined score is obtained as

$$\eta = w\mathcal{N}_{(0,\sigma_h)}(2\sigma_h),$$

which corresponds to the ground truth score associated to any pixel head which is located at a distance equal to the target head radius from the labeled center. Thus, only for the scope of the evaluation, any pixel value, both in the ground truth map and in the estimated one, is set to zero if it is lower than η.

Given the truncated maps, instance segmented masks are obtained by applying the watershed method explained in Section 8.6. At this point, each detection will be represented by a connected set of pixels. The detections are assigned with a score value, which expresses the likelihood of being a real head. In our context, we assign as score value the value of the maximum of the weighted distribution.

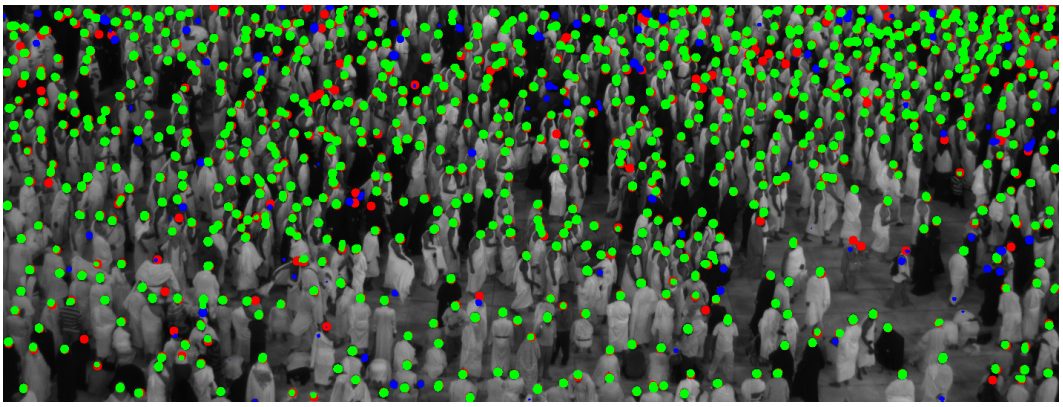
The standard approach for deciding whether an estimated detection is a TP or a FP is to calculate the Intersection over Union (IoU) score. Given a ground truth blob and a detection, the IoU is the ratio between the number of pixels which belong both to the ground truth and the detection, and the number of pixels which belong to union of the ground truth with the detection. A detection is then considered a TP if its IoU value is over a predefined value $\mathscr{I}$. If multiple detections are positive for a given ground truth target, only the one with the highest score is labeled as TP, while the others are FP.

As accuracy metric we use the *mean Average Precision* (mAP), which is the standard metric for measuring the performance of object detectors. It serves as an alternative representation of the area under the precision-recall curve. In order to evaluate the mAP value, the detections are ranked by descending score. The recall at rank r is defined as the ratio between the number of TP detections ranked r or higher and the total number of ground truth positives. The precision at r is defined as the proportion of all the examples ranked r or higher which are TP. The mAP score is then the integral of the approximated curve built by taking, for any distinct target value of recall, the maximum precision which occurs for a recall higher or equal than the target.

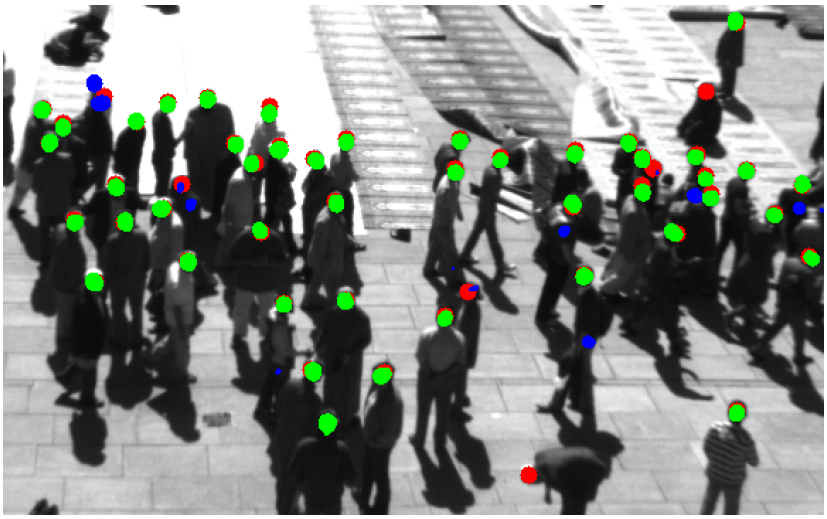
In the following we will call $\text{mAP}_{\text{I}=\mathscr{I}}$ as the mAP calculated with a IoU threshold of $\mathscr{I}$.

8.8 Results

Figure 8.5 shows qualitative results of the pedestrian detection with the UNet for both *Mecca* and *Regent's Park Dense* datasets. Table 8.2 and 8.3 show the mAP scores for both UNet and FE+LFE for *Mecca* and *Regent's Park Dense* datasets respectively. With regard to the choice of the IoU threhsold, 0.3 and 0.5 values have been used. The value $\mathscr{I} = 0.5$ is the standard value for evaluating object detectors in the PASCAL VOC 2012 dataset [38]. However, when considering our specific problem, such constraint could be too strict into evaluating the goodness of the detector. The reason is that since there is imprecision in the head center labeling, some bias in the ground truth could have a noticeable impact in the calculated IoU, since the head blobs are small. Thus, we relax the IoU threshold to a value of 0.3 in order to allow for some small misplacement of the estimated location of the head, and we provide results for both thresholds.



(a)



(b)

Figure 8.5: Detections on the (a) *Mecca* dataset and on the (b) *Regent's Park Dense* dataset. The pixel blobs are colored in according the the following convention. Red blobs are ground truth heads, green blobs are true positive detections, and blue blobs are false positive detections.

Mecca	$mAP_{I=0.3}$	$mAP_{I=0.5}$
UNet	0.86	0.74
FE+LFE	0.88	0.74

Table 8.2: Quantitative results of the two different architectures presented in Section 8.4 on the *Mecca* dataset in terms of mAP, for 0.3 and 0.5 IoU thresholds.

The two different architectures provide comparable performance in both datasets, with FE+LFE outperforming by a small margin the UNet for $\mathscr{I} = 0.3$ in the *Mecca* dataset and $\mathscr{I} = 0.5$ in the *Regent's Park Dense* dataset. We can also notice that the overall performance of the method in the two datasets remains constant, even if the number of training images and the crowd density vary, which is a desirable property.

Regent's Park	$mAP_{I=0.3}$	$mAP_{I=0.5}$
UNet	0.88	0.76
FE+LFE	0.88	0.78

Table 8.3: Quantitative results of the two different architectures presented in Section 8.4 on the *Regent's Park Dense* dataset in terms of mAP, for 0.3 and 0.5 IoU thresholds.

8.9 Conclusion

In this chapter we have proposed an interpretation of the problem of detecting pedestrian heads in a dense crowd as a semantic segmentation problem. We have highlighted the potential difficulties of training with few data and on imprecise labels. We have highlighted an original ground truth definition which takes into account the irreducible uncertainty of the input labels, and we have adapted state-of-the-art architectures to learn on such data. The quality of the networks have been assessed by using standard evaluation metrics used in the object detection community.

The models presented in this section can be used further in order to infer pedestrian maps in multiple cameras, leading the way for data fusion on the common ground plane. When N cameras are available, one can obtain N pedestrian maps which serve as independent sources estimating the crowd occupation. Such maps do not have any cue of the 3D world, but they can still be projected into the ground plane by setting the height of each head as unknown. Moreover, for the final fusion problem, these maps will provide orthogonal information with respect to the geometric detector of Part II.

Chapter 9

Fusion of Appearance and Geometry Information on the Ground Plane

Contents

9.1 Motivation

In the previous chapter we have presented a method for obtaining pedestrian detection maps in the image space. However, one detector alone cannot capture all the characteristics of the crowd, because an important percentage of pedestrians could be occluded at a given time instant when observed from one camera, while in some cases the homogeneity between two very close pedestrians might be too high to detect them individually in a reliable manner. For this reason, it is beneficial to perform the pedestrian detection map inference on multiple views, if available. The network must thus show perspective invariance properties when detecting pedestrians from different locations, and thus being able to well represent heads at any angle. In our specific scenario the detector, for the *Regent's Park Dense* dataset, has been trained on data acquired by the central camera only, since in many cases data from a single view might be readily available in advance for training, due to placement constraints. In most settings this can cause an imbalance on the number of examples of heads for each orientation, since usually the infrastructure of the scene forces the crowd to follow a common direction of movement, thus making the problem of perspective invariance more challenging.

As the maps from different views are acquired, one has to propose a robust method in order to combine them on the ground plane. This will give both localization information of the target (with a precision dependent on the number of sources with consensus) and can be exploited to easily remove false positives. Once a ground plane detection location proposal is obtained from single view maps, this can be further combined with the estimation provided by the geometric detector of Part II.

9.2 From pedestrian maps to ground plane detections

In order to have a common frame for fruther combination of detections, starting from a blob of one of the pedestrian maps, its bounding box is extracted. This operation is performed in such a way as to minimize the number of successive geometric projections. Each vertex of the bounding box can be thus projected on the ground plane, given an hypothesis for its height. The projection makes use of the variable height homographies estimated in Part II. Considering a height interval between 1.4 and 2 meters, the top corners are projected at 1.4 meters, and the bottom ones at 2. The corresponding points in the ground form a new box which encloses all the possible locations of the head in the ground plane conditioned by the height interval.

Each camera source input will now consist of a set of ground plane boxes. All the sources have to be combined together, needing data association in order to associate boxes of the same target. Data association between N sources is tackled as $(N-1)$ data association problems in cascade, from the left-most (or right-most) camera to the right-most (or left-most). Since there are two possible directions of association leading to different outputs, the results of the two combinations will be further fused together in a second phase.

Let us consider the problem of associating the first two sources. For data association, in addition to the possible ground location represented by the box intersection, the height assumption needs to be considered. Indeed two boxes could intersect in such a way that the height of the target estimated from one view would be totally different from the height estimated by the other. Thus the cost of data association also depends on the agreement on the interval of possible heights estimated by the two sources.

Figure 9.1 describes how the cost between two boxes belonging to two different sources is evaluated. First the intersection of the boxes is extracted as a list of vertexes $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$. For a given vertex X_i, the following re-projection scheme is performed. Let us consider a set of heights ranging from 1.4 to 2 meters, with a step of 2.5 centimeters. The point in the ground plane X_i is projected on the original image of camera j for each possible height. All the projected points in the image form a discretized segment. The range of heights which are coherent with the original bounding box detection is the one for which the corresponding projected points are in the interior of the box. The set of possible heights for X_i on camera j is then expressed as:

$$H^{X_i,j} = \left[h_{min}^{X_i,j}, h_{max}^{X_i,j} \right]$$

The set of possible heights for camera j is then calculated by performing the union of the intervals for each intersection vertex:

$$H^j = \bigcup_{i=1\ldots n} H^{X_i,j}$$

We now have the locus of coherent heights for each of the two cameras, i.e. the intervals H^{j_1} and H^{j_2}. The cost of the association of the two boxes is then evaluated as:

$$C = -\log\left(\frac{| H^{j_1} \cap H^{j_2} |}{\min\left(| H^{j_1} |, | H^{j_2} | \right)} \right)$$

The cost penalizes a low intersection over minimum metric, which is an alternative interval comparison to intersection over union, where an unbalanced size of the intervals penalizes the score. In this specific case, a difference in the interval size corresponds to a difference in the size of the original detections in the image space. We do not aim at penalizing such difference because it could be caused just by a partial occlusion in one of the views.

Given the cost definition, data association is performed by using a Hungarian algorithm [73, 116]. Let us consider the data association between camera 0 and camera 1. A new set of boxes is computed, containing the intersections of associated detections, and the original boxes of each camera which have not been associated. The next data association will be performed between this new set of boxes and the boxes of camera 2. No box is discarded during the process, since all the non-associations are propagated as they are.

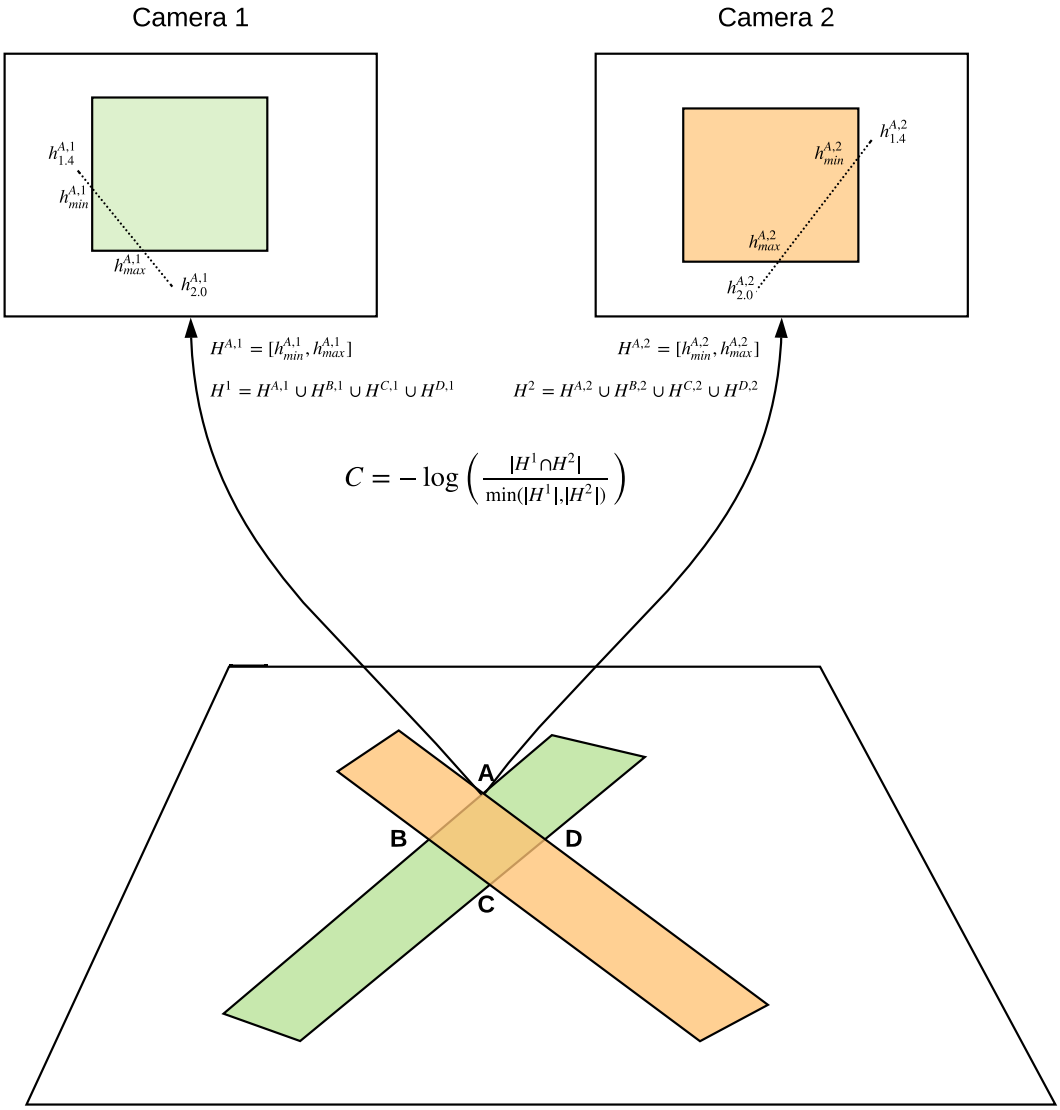
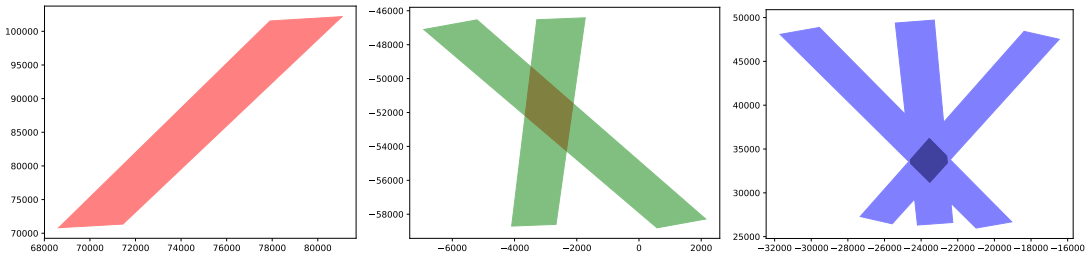


Figure 9.1: Data association cost computation for a pair of boxes. The intersection between the boxes is computed. Each vertex of the intersection is re-projected in the original images at varying heights (mapping to a segment). The interval $H^{X,j}$ of possible heights for that vertex X on camera j is extracted as the portion of the re-projected segment which intersects the original bounding box. The overall interval of plausible heights H^j for camera j is then computed as the union of the intervals of all the vertexes of the intersection. The cost C is then computed as the negative logarithm of the intersection over the minimum of the height intervals of the two cameras.

9.3 BBA construction

Each polygon resulting from cascade data association is transformed into a separate BBA. Such BBA is consonant and consists of two focal elements (one if no association occurred), the internal corresponding to the intersection itself, and the external being given by the disjunction of all the boxes which have generated such intersection. The second focal element models the possible locations of the head given that some source may be unreliable.



(a) **(b)** **(c)**

Figure 9.2: Examples of BBA construction from supervised detection associated boxes on the ground plane. (a) No association; (b) Two intersecting boxes; (c) Three intersecting boxes.

Figure 9.2 shows an example for each type of association, namely no association and association between two or three boxes. The two focal elements are assigned with masses 0.51 (internal) and 0.49 (external) for symmetry breaking.

9.4 Combination of one-directional data associations

As introduced in Section 9.2, data association can be performed in one of two directions, starting from the first or the last camera. The direction of association will produce in general different results, and no one is clearly better than the other. For these reasons, after having constructed the BBA for each direction, the two estimations are combined conjunctively. Such combination introduces a new data association problem. The idea is that if one box is connected in one direction to a different set of boxes of the other, this information would be combined in a unique BBA containing that box. Thus an ad-hoc constraint for data association of two BBAs is that at least one of the generating original bounding boxes is in common. This rule avoids the circular problem of possibly finding intersecting boxes needing another height interval evaluation. Together with this pre-filtering stage, the conflict-based data association cost in Section 6.5.3 of Chapter III.6 is used.

Once the data association has been completed, the matched BBAs are fused by using a conjunctive rule. Since the BBAs are not independent (they comes from the same sources), Denoeux' cautious rule [29] would have been more relevant, at least theoretically. However, in this study, it has not a significant impact on the results and we use the classical conjunctive rule. Prior to fusion, a discounting may be applied. If one BBA has been generated by an intersection deriving from less boxes than the other, the first source is then considered less reliable and thus discounted by a fixed value, according to global discounting (Equation 5.8 of Chapter III.5). Let us consider the case of a BBA like the one in Figure 9.2a and another BBA like the one in Figure 9.2c. Without any discounting, the single box would force the other two uncertainty directions to disappear. However, as a single detection, such information should not have such power and is thus discounted. The single box will have now merely the function of increasing the belief in that direction.

Figure 9.3 shows an example of final BBA map after the combination of the single direction associations.

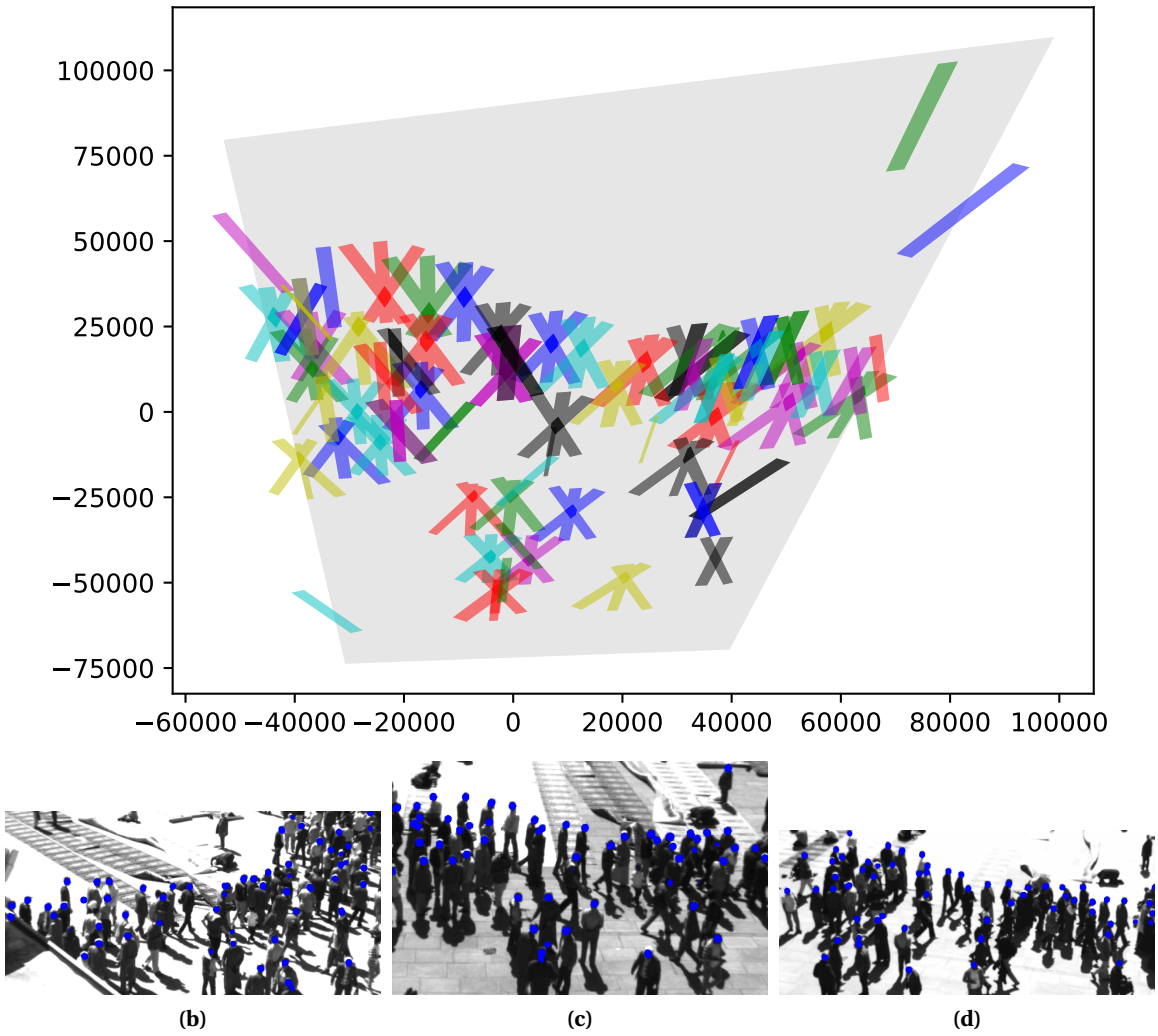


Figure 9.3: Example of supervised detection BBAs after combination of one-directional data association estimations. (a) In grey the region of interest, as a projected crop from the central camera. Each BBA is depicted by a different color. In (b)-(d), as a reference, the input single view detections in the image space are shown.

9.5 Combination with the geometry detector

The BBAs of the appearance-based supervised detector can be further combined with BBAs constructed from the geometry detections. The BBA construction for such unsupervised clusters is the same as the one presented in Section 6.5.3 of Chapter III.6. Data association of the geometric and appearance-based detectors works as follows. Two BBAs intersecting such that the inner focal element of the geometric detection (the one embedding the height hypothesis) has a non empty intersection with the appearance-based detector BBA have to satisfy a height coherence test in order to be a candidate match. The resulting intersection between the inner focal element of the geometric detection and the appearance-based detector BBA, by using the same re-projection technique as presented in Section 9.2, can be used in order to estimate the possible heights of the person from each of the cameras. The two BBAs are matchable if the height estimation of the geometric centroid is included inside each of the intervals for the various cameras. If such condition holds, conflict-based cost is used for data association (Section 6.5.3 of Chapter III.6).

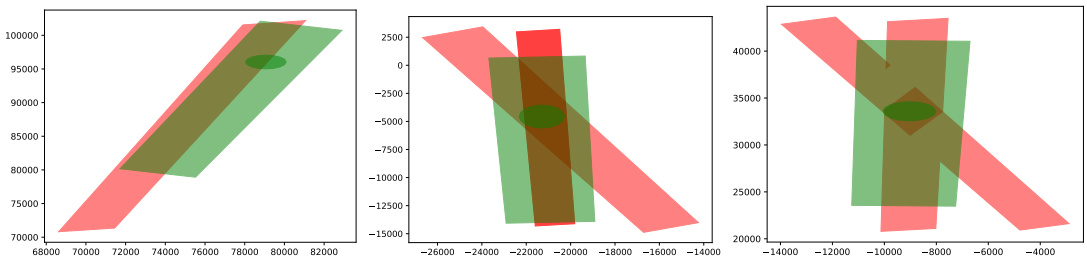
The benefits of the combination with geometry can be summarized as follows:

1. Reinforce single camera detections with no associations. When boxes are not associated, they are less reliable and they gather no knowledge on the possible height of the target. For such reason, as shown later, they would be removed from the candidate detections. A com-

bination with the geometry, however, would not only enforce the hypothesis, but also provide a punctual height inference (see Figure 9.4a).

2. Increase localization precision of matched boxes. Especially when few sources agree on the target location (e.g. 2), the intersection area, and thus the possible location of the target, may be of an important size. Geometry can naturally reduce the area of search (see Figure 9.4b).

3. Solve ambiguities in localization. Sometimes the decision making could select as target region for the true location one which is fragmented in multiple areas. The estimation of the person location would thus be carried randomly between the fragments. Geometry information can help to solve such issue by increasing the belief on one of the fragments (see Figure 9.4c).



(a) **(b)** **(c)**

Figure 9.4: Benefits introduced by geometric detection. (a) Reinforce single box detections; (b) Increase localization precision; (c) Solve ambiguities in localization.

9.6 Estimation of pedestrian location

The results of data fusion in the 3D space are evaluated on instantaneous detections. At each time stamp pedestrian locations are extracted as barycenters of the polygons maximizing the plausibility of the singletons (*contour function*). Such approach has been favored, for this task , to BetP maximization since the BetP tends to favor regions with lower cardinality as maximizers. However, a small intersection in our context does not mean always that the localization is more precise, but it could be due to a very small intersection overlap between bounding boxes, and thus could be less reliable than large intersections. In general, excluding occlusion, the size of a good intersection in the ground plane should be almost constant if the head size in the image space is almost constant. For this reason, plausibility maximization is preferred being completely independent from the cardinality of the maximizer. The algorithm for maximizing the contour function is exactly the same as the one for BetP maximization presented in Section 6.4 of Chapter III.6, thus requiring evaluation on maximal intersections, by changing only the objective function to maximize. See Figure 9.5 for an example on pedestrian location estimation at a given time frame.

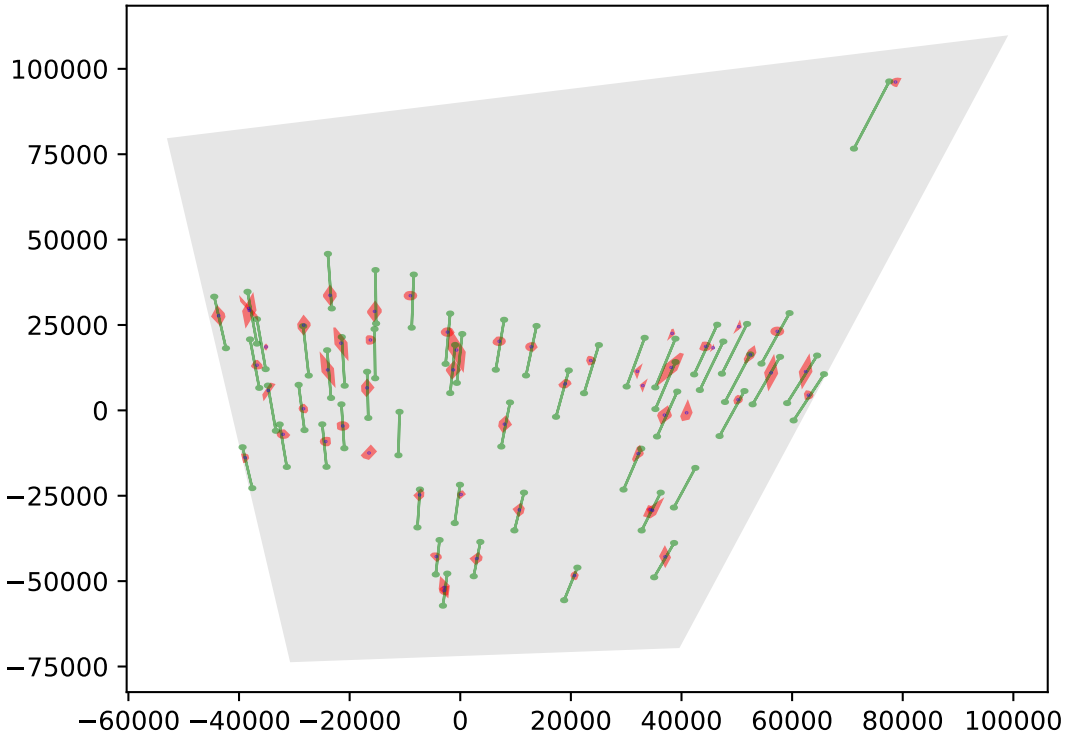
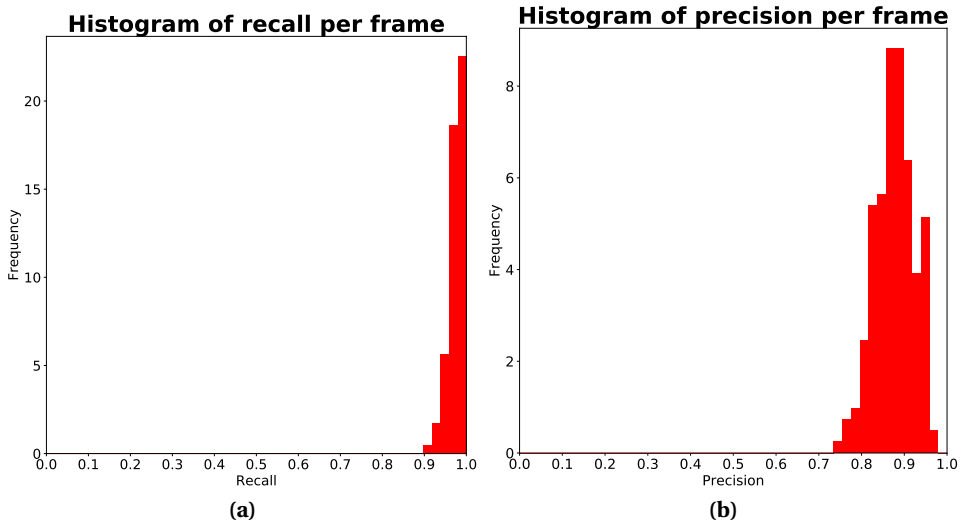


Figure 9.5: Example of pedestrian location extraction through contour function maximization. In grey the region of interest, in green the ground truth segments, in red the maximizers for each BBA. The blue dots are the barycenters of such areas, and thus the final pedestrian locations.



(a) **(b)**

Figure 9.6: Histograms of detection recall and precision computed at each frame independently (200 frames, bin size = 0.02).

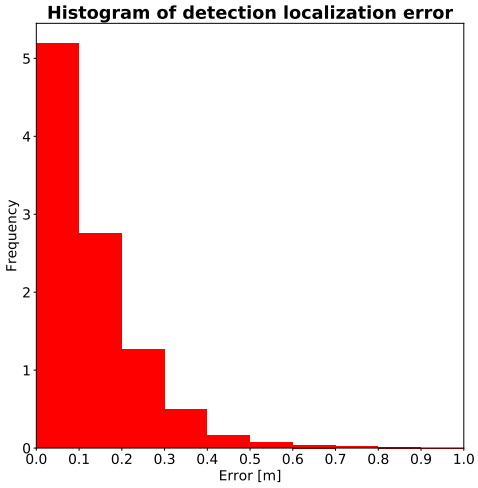


Figure 9.7: Normalized histogram of detection localization error (200 frames, bin size = 0.1).

9.7 Results

The algorithm is tested on the 200 test frames of the *Regent's Park Dense* dataset. The precision and recall are evaluated against the ground truth segment in such a way that each location is associated to it by ignoring the height component, thus calculating a point-to-line distance to the ground truth segment. This assumption aims to split in two different evaluations the occupancy estimation (whether a human is present or not) from the localization task (how close the detection is to the real person). The distance between the estimated location and the ground truth has to be lower than 20 centimeters in order to be considered as a true positive, taken as an average head-to-shoulder distance. The recall and precision values in all the sequence are 97.89% and 87.31% respectively, reflecting excellent performance from the combination of the various sources. Figure 9.6 shows the histogram of recall and precision calculated at each frame. Such information allows us to understand if the values maintain low variance during the sequence. Recall values never go under 90% for all the frames, while the precision follows an approximately normal distribution with a minimum value of recall around 75%.

We evaluate also the instantaneous localization error of the detections prior to tracking. The localization error of a detection is the distance between the detection itself and the point in the ground truth segment which corresponds to the estimated height of the pedestrian. Figure 9.7 shows the histogram of localization errors on the whole sequence. The average localization error is 13.12 centimeters, being extremely precise for the given task.

9.8 Conclusion

In this Chapter we have presented a method to combine supervised detections from single cameras together in a common reference plane and perform the fusion of such sources with the unsupervised geometric detector. We highlight different data association schemes for the different sources, by integrating the height information as an additional information to the 2D discernment frame. We demonstrate our approach in the *Regent's Park Dense* dataset, which presents area at high densities, by showing competitive results in term of accuracy.

A straightforward extension of this approach is to feed the detection BBA extracted at every frame into a tracking framework. At first glance, the performance of the pedestrian tracking of Section 6.5.3 of Chapter III.6 have to be evaluated at this higher level of density, since ambiguities in the track-detection matching would be always more frequent, while a large branch of tracking research can be carried out specifically for handling the dense case.

An interesting alternative approach to the one presented would try to overcome the fundamental limitation of handling the height data as an external variable to the discernment frame. This causes the height to be used as a filtering gate for 2D operations. A more elegant solution would be to handle 3D BBAs, thus including the height in the discernment frame. This would allow for seamless combination and conjunction of detections from different sources. Handling 3D BBAs would implicitly imply an extension of the 2CoBel framework. Such extension would be limited, however, only to the geometric representation part, since the graph representation (and the related algorithms which are based on it) does not depend on the number of dimensions of the discernment frame. Thus 2CoBel should define methods for operator computation (intersection, union, etc...) which generalize the 2D clipping methods.

Conclusion and future work

Conclusion

In this thesis we have addressed the problem of pedestrian detection in high-density crowds. The main contribution of this thesis is to propose a complete framework for detection and tracking of pedestrians in difficult crowded scenes, in which classical computer vision fails, by using multiple calibrated cameras. The work has been carried out in several parts with a strong link with each other. Each of these parts, spanning from camera calibration to data fusion, targets a specific task in the detection framework, and tackles novel problems introduced by dense crowds.

In Part I we have presented the problem of estimating the relative pose of two cameras in urban scenes. We have highlighted the difficulties of such scenarios (constrained locations of cameras, scarcity of matching parts, ambiguities), and we have shown that state-of-the-art estimators on image pairs fail under these conditions. We have thus proposed an iterative method which uses an entire video stream in order to estimate the relative pose, guided by the assumption that gathering information from multiple image pairs can solve the problem of few data available at each instant. On the other side, since we deal with highly unreliable information, we have proposed to guide the acquisition of new data on the basis of the estimated confidence of the new solution in each part of the image space. Such approach has given high quality estimates of relative pose for our dataset, and has demonstrated to be applicable also to refine some already available pose. The output of this Part is the extrinsic calibration of each camera pair of the system, together with the good matched points which have been used to estimate and refine the model.

In Part II such calibration is used in order to metrically relate the system to a common ground plane reference. Thus, one can freely project back and forth points from the image space to the 3D space, knowing the metric distance of such point from the ground plane. This automatic registration represents a first contribution, since the state-of-the-art multiple camera approaches need extensive manual intervention. The main objective of Part II is then to propose an unsupervised pedestrian detector which is based on the estimated geometry. The use of multiple cameras at the same time for the detection can automatically handle occlusion, while giving a complementary insight on the scene, with respect to single view supervised detectors. We tackle this as a stereo matching problem, where pixels in one view are projected in the others at variable possible heights over the ground. Thus, the stereo matching is formulated in terms of a height hidden variable, which allows to define novel discontinuity functions among neighboring pixels. The output of this Part is a joint estimation of the presence and height of pedestrians, which results in a 3D pedestrian occupation map with respect to the registered ground plane.

In Part III we handle the problem of information fusion in 2D spaces, since we aim to apply fusion between multiple sources and at subsequent time steps in the ground plane. We realize that current approaches for 2D data fusion make use of representations which are too computationally inefficient in order to be used in large scale scenes. We then move to the theoretical problem of proposing an efficient representation in order to handle information fusion efficiently at any scale. Our contribution consists into two complementary representations, one geometrical and one structural, which provide together the foundation for the definition of primitive operators, combination rules, decision making methods and several other algorithms of Belief Function Theory. We have demonstrated that such new formulation is scale invariant, and thus can be

used to perform fusion at the finest level of detail. We have demonstrated its effectiveness by performing evidential tracking of the detections extracted in Part II over a long period of time, with competitive performance in terms of localization error.

In Part IV we focus on the problem of supervised pedestrian detection with single cameras. These detectors represent additional important sources of information that can be exploited for information fusion, in order to obtain a global robust detector by combining them with the one proposed previously. We highlight the problems related to data acquisition and labeling for video-safety applications, which in turn reflect on the design of the architectures and of the training phase. We underline a new ground truth definition scheme, which copes with soft labeling of the pedestrian heads and with possible errors in the manual ground truth acquisition process, which may harm the network efficiency. We then adapt state-of-the-art semantic segmentation networks for our problem, and we show that they achieve competitive performance independently from the difficulty of the images (in terms of crowd density). The output of this Part is a pedestrian map for each independent view, where each head is represented by a segmented blob of pixels.

At this point we have the unsupervised detection from the algorithm in Part II, the multiple supervised detection maps from the network of Part IV, and the fusion framework which may combine different sources in the ground plane III. We finally exploit our proposed representation in order to perform spatial information fusion between the unsupervised detections, which carry knowledge on the 3D location of the pedestrian, and the independent supervised maps, which miss the information on the height of each detection over the ground plane. We demonstrate the efficiency of our approach on a challenging scenario with high and not uniform density distribution.

Future work

Many issues addressed in the different parts of the thesis present challenges which are still open and may be further investigated in order to improve the quality of the final output of the framework.

In terms of the calibration performed in Part I, the iterative approach for the estimation has the desirable property of being convergent towards a stable solution for any choice of the parameters of the method. Moreover, the quality of such solution is not sensitive to small variations of such parameters. However, some hyperparameters of such algorithm, for example the maximum allowed error for the actual solution when no information is available, can vary greatly from one dataset to the other. In fact, one dataset can provide much scarcer and less qualitative matches than another, presenting estimation errors on a different scale, and thus for a much larger maximum drift of the solution at each iteration. We believe that, instead of setting manually such prior information, some critical hyperparameters could be learned. This would imply training the process on a vast amount of different datasets and camera configurations, thus requiring an important effort in terms of data acquisition.

In terms of the detection part, in this work we have split the multiple camera detection and the appearance-based detection in two independent problems, by subsequently performing the fusion among them at the last stage. An alternative approach that is worth exploring is to integrate the appearance information in the data term of the pedestrian height map optimization that is done with all the cameras at the same time, in order to have a lower level data fusion which could allow for more robust decision making. Moreover, such method could be trainable end-to-end, from the images to the height map. In fact, some works have succeeded into proposing optimization techniques on Conditional Random Fields (CRFs) by using Recurrent Neural Networks (RNNs) [187]. Thus one could feed to such network the output of the semantic segmentation networks as data term, and the proposed discontinuity function as smoothness term, by then optimizing the pixel height labeling with the additional advantage that the regularization strength would be also learnt.

A natural extension of the work, which would lead to an important research topic, is to propose

a robust tracker for the ground plane tracking problem. While the tracking employed in Part III serves as a baseline example of the potential of the work, more sophisticated tracking techniques exist, such as evidential box particle filters [160]. The problem introduces novel information with respect to regular 2D tracking, because each target has indeed a 3D state, with an estimation of the height interval at which the head is located. Thus exploiting this additional information could greatly improve the robustness of the tracker.

Bibliography

[1] Ákos Kiss and Szirányi, T. (2013). Localizing people in multi-view environment using height map reconstruction in real-time. *Pattern Recognition Letters*, 34(16):2135 – 2143. 44

[2] Alahi, A., Jacques, L., Boursier, Y., and Vandergheynst, P. (2011). Sparsity driven people localization with a heterogeneous network of cameras. *Journal of Mathematical Imaging and Vision*, 41(1-2):39–58. 44

[3] Alahi, A., Ramanathan, V., and Fei-Fei, L. (2017). Tracking millions of humans in crowded spaces. In *Group and Crowd Behavior for Computer Vision*, pages 115–135. Elsevier. 103

[4] Alan Gray, E. (2017). Best practice guide-gpgpu. xv, 58

[5] André, C., Le Hégarat-Mascle, S., and Reynaud, R. (2015). Evidential framework for data fusion in a multi-sensor surveillance system. *Engineering Applications of Artificial Intelligence*, 43:166 – 180. 74, 75, 90, 96, 98, 99

[6] Ataer-Cansizoglu, E., Taguchi, Y., Ramalingam, S., and Miki, Y. (2014). Calibration of non-overlapping cameras using an external slam system. In *3D Vision (3DV), 2014 2nd International Conference on*, volume 1, pages 509–516. IEEE. 16

[7] Ayaz, S. M., Kim, M. Y., and Park, J. (2017). Survey on zoom-lens calibration methods and techniques. *Machine Vision and Applications*, 28(8):803–818. 13

[8] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359. 7

[9] Benfold, B. and Reid, I. (2011). Stable multi-target tracking in real-time surveillance video. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3457–3464. IEEE. 104

[10] Beucher, S. and Meyer, F. (1992). The morphological approach to segmentation: the watershed transformation. *Optical Engineering-New York-Marcel Dekker Incorporated-*, 34:433–433. 112

[11] Bloch, I. (1996). Some aspects of Dempster-Shafer evidence theory for classification of multi-modality medical images taking partial volume effect into account. *Pattern Recognition Letters*, 17(8):905–919. 73

[12] Boost (2015). Boost C++ Libraries. `http://www.boost.org/`. Last accessed 2018-03-13. 84

[13] Boutros, N., Shortis, M. R., and Harvey, E. S. (2015). A comparison of calibration methods and system configurations of underwater stereo-video systems for applications in marine ecology. *Limnology and Oceanography: Methods*, 13(5):224–236. 13

[14] Brückner, M., Bajramovic, F., and Denzler, J. (2014). Intrinsic and extrinsic active self-calibration of multi-camera systems. *Machine vision and applications*, 25(2):389–403. 13

[15] Canziani, A., Paszke, A., and Culurciello, E. (2016). An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*. 105

[16] Caspi, Y., Simakov, D., and Irani, M. (2006). Feature-based sequence-to-sequence matching. *Int. J. Comp. Vis.*, 68(1):53–64. 17

[17] Chang, M.-C., Krahnstoever, N., and Ge, W. (2011). Probabilistic group-level motion analysis and scenario recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 747–754. IEEE. 45

[18] Chaquet, J. M., Carmona, E. J., and Fernández-Caballero, A. (2013). A survey of video datasets for human action and activity recognition. *Computer Vision and Image Understanding*, 117(6):633–659. 29

[19] Chavez-Garcia, R. O. and Aycard, O. (2016). Multiple sensor fusion and classification for moving object detection and tracking. *IEEE Transactions on Intelligent Transportation Systems*, 17(2):525–534. 73

[20] Chum, O. and Matas, J. (2005). Matching with prosac-progressive sample consensus. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 220–226. IEEE. 8, 25

[21] Conte, D., Foggia, P., Percannella, G., and Vento, M. (2013). Counting moving persons in crowded scenes. *Machine vision and applications*, 24(5):1029–1042. 29

[22] Criminisi, A., Reid, I., and Zisserman, A. (1999). Single view metrology. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 434–441. IEEE. xiv, 42, 43, 51

[23] Csurka, G., Csurka, G., Zeller, C., Zeller, C., Zhang, Z., Zhang, Z., Faugeras, O., Faugeras, O., and Robotvis, P. (1995). Characterizing the uncertainty of the fundamental matrix. *Computer Vision and Image Understanding*, 68:18–36. 12

[24] Cuzzolin, F. (2008). A geometric approach to the theory of evidence. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(4):522–534. 74

[25] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE. 104

[26] Damen, D. and Hogg, D. (2012). Detecting carried objects from sequences of walking pedestrians. *IEEE transactions on pattern analysis and machine intelligence*, 34(6):1056–1067. 44

[27] Dang, T., Hoffmann, C., and Stiller, C. (2009). Continuous stereo self-calibration by camera parameter tracking. *IEEE Transactions on Image Processing*, 18(7):1536–1550. 24

[28] Dehghan, A., Idrees, H., Zamir, A. R., and Shah, M. (2014). Automatic detection and tracking of pedestrians in videos with various crowd densities. In *Pedestrian and Evacuation Dynamics 2012*, pages 3–19. Springer. 104

[29] Denœux, T. (2008). Conjunctive and disjunctive combination of belief functions induced by nondistinct bodies of evidence. *Artificial Intelligence*, 172(2-3):234–264. 70, 71, 84, 120

[30] Denoeux, T., El Zoghby, N., Cherfaoui, V., and Jouglet, A. (2014). Optimal object association in the Dempster–Shafer framework. *IEEE transactions on cybernetics*, 44(12):2521–2531. 73

[31] Denœux, T. and Yaghlane, A. B. (2002). Approximating the combination of belief functions using the fast moebius transform in a coarsened frame. *International Journal of Approximate Reasoning*, 31(1-2):77–101. 84

[32] Devarajan, D., Radke, R. J., and Chung, H. (2006). Distributed metric calibration of ad hoc camera networks. *ACM Transactions on Sensor Networks (TOSN)*, 2(3):380–403. 13

[33] Dubois, D. and Prade, H. (1988). Representation and combination of uncertainty with belief functions and possibility measures. *Computational intelligence*, 4(3):244–264. 78

[34] Dubois, D. and Prade, H. (1991). Measuring and updating information. *Inf. Sci.*, 57-58:181–195. 84

[35] Dubuisson, S. and Gonzales, C. (2016). A survey of datasets for visual tracking. *Machine Vision and Applications*, 27(1):23–52. 29

[36] Eshel, R. and Moses, Y. (2010). Tracking in a dense crowd using multiple cameras. *International Journal of Computer Vision*, 88(1):129–143. xiv, 29, 44, 45, 47, 48, 56

[37] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231. 21

[38] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2012). The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 114

[39] Felzenszwalb, P. F. and Huttenlocher, D. P. (2006). Efficient belief propagation for early vision. *International journal of computer vision*, 70(1):41–54. 61

[40] Ferryman, J. and Shahrokni, A. (2009). Pets2009: Dataset and challenge. In *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on*, pages 1–6. IEEE. 17, 25, 29

[41] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24(6):381–395. 7, 8, 50

[42] Fleuret, F., Berclaz, J., Lengagne, R., and Fua, P. (2008). Multi-camera people tracking with a probabilistic occupancy map. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(2):267–282. 44

[43] Foroughi, H., Ray, N., and Zhang, H. (2015). Robust people counting using sparse representation and random projection. *Pattern Recognition*, 48(10):3038–3052. 29

[44] Fortin, B., Hachour, S., and Delmotte, F. (2017). Multi-target PHD tracking and classification using imprecise likelihoods. *International Journal of Approximate Reasoning*, 90:17–36. 73

[45] Fradi, H., Luvison, B., and Pham, Q. C. (2017). Crowd behavior analysis using local mid-level visual descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(3):589–602. 29

[46] Fraundorfer, F., Tanskanen, P., and Pollefeys, M. (2010). A minimal case solution to the calibrated relative pose problem for the case of two known orientation angles. *Computer Vision–ECCV 2010*, pages 269–282. 16

[47] Ge, W. and Collins, R. T. (2009). Marked point processes for crowd counting. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2913–2920. IEEE. 44, 103

[48] Gebru, I. D., Alameda-Pineda, X., Forbes, F., and Horaud, R. (2016). Em algorithms for weighted-data clustering with application to audio-visual scene analysis. *IEEE Trans. on pattern analysis and machine intelligence*, 38(12):2402–2415. xv, 50

[49] Gemeiner, P., Micusik, B., and Pflugfelder, R. (2015). *Calibration Methodology for Distant Surveillance Cameras*, pages 162–173. Springer International Publishing, Cham. 16

[50] Goldman, Y., Rivlin, E., and Shimshoni, I. (2017). Robust epipolar geometry estimation using noisy pose priors. *Image and Vision Computing*, 67:16–28. 16

[51] Guan, L., Franco, J.-S., and Pollefeys, M. (2010). Multi-view occlusion reasoning for probabilistic silhouette-based dynamic scene reconstruction. *International journal of computer vision*, 90(3):283–303. 44

[52] Guo, X. and Cao, X. (2010). Triangle-constraint for finding more good features. In *Pattern Recognition (ICPR), Int. Conf. on*, pages 1393–1396. 15

[53] Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., and Alahi, A. (2018). Social gan: Socially acceptable trajectories with generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, number CONF. 104

[54] Hachour, S., Delmotte, F., Mercier, D., and Lefèvre, E. (2014). Object tracking and credal classification with kinematic data in a multi-target context. *Information Fusion*, 20:174–188. 73

[55] Hamaguchi, R., Fujita, A., Nemoto, K., Imaizumi, T., and Hikosaka, S. (2018). Effective use of dilated convolutions for segmenting small object instances in remote sensing imagery. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1442–1450. IEEE. xix, 111

[56] Hamid, R., Kumar, R., Hodgins, J., and Essa, I. (2014). A visualization framework for team sports captured using multiple static cameras. *Computer Vision and Image Understanding*, 118:171–183. 44

[57] Han, T. D. and Abdelrahman, T. S. (2011). Reducing branch divergence in gpu programs. In *Fourth Workshop on General Purpose Processing on GPU*, page 3. ACM. 59

[58] Hansen, P., Alismail, H., Rander, P., and Browning, B. (2012). Online continuous stereo extrinsic parameter estimation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1059–1066. IEEE. 13, 24

[59] Harris, M. (2007). Optimizing cuda. *SC07: High Performance Computing With CUDA*. 60

[60] Harris, M., Sengupta, S., and Owens, J. D. (2007). Parallel prefix sum (scan) with cuda. *GPU gems*, 3(39):851–876. 59

[61] Hartley, R. I. (1997). In defense of the eight-point algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(6):580–593. 6, 7

[62] Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition. xiii, 3, 4, 5, 6, 11, 14, 20, 25

[63] He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE. 112

[64] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034. 112

[65] Helbing, D., Johansson, A., and Al-Abideen, H. Z. (2007). Dynamics of crowd disasters: An empirical study. *Physical review E*, 75(4):046109. vii

[66] Hough, P. V. (1962). Method and means for recognizing complex patterns. US Patent 3,069,654. 90

[67] Hu, P. and Ramanan, D. (2017). Finding tiny faces. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 1522–1530. IEEE. 104

[68] Hu, R., Wang, R., Shan, S., and Chen, X. (2014). Robust head-shoulder detection using a two-stage cascade framework. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 2796–2801. IEEE. 104

[69] Idrees, H., Soomro, K., and Shah, M. (2015). Detecting humans in dense crowds using locally-consistent scale prior and global occlusion reasoning. *IEEE transactions on pattern analysis and machine intelligence*, 37(10):1986–1998. 103

[70] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*. 106

[71] Jin, Z., An, L., and Bhanu, B. (2016). Group structure preserving pedestrian tracking in a multi-camera video network. *IEEE Transactions on Circuits and Systems for Video Technology*. 45

[72] Johnson, A. (2014). Clipper - an open source freeware library for clipping and offsetting lines and polygons. `http://www.angusj.com/delphi/clipper.php`. Last accessed 2018-03-13. 75

[73] Jonker, R. and Volgenant, A. (1987). A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340. 96, 118

[74] Jousselme, A.-L., Grenier, D., and Bossé, É. (2001). A new distance between two bodies of evidence. *Information fusion*, 2(2):91–101. 96

[75] Kanjanatarakul, O., Denoeux, T., and Sriboonchitta, S. (2016). Prediction of future observations using belief functions: a likelihood-based approach. *International Journal of Approximate Reasoning*, 72:71–94. 21

[76] Karpathy, A. (2018). Cs231n: Convolutional neural networks for visual recognition. Last accessed 2018-10-24. xvi, 105

[77] Kasten, Y., Ben-Artzi, G., Peleg, S., and Werman, M. (2016). Fundamental matrices from moving objects using line motion barcodes. In *European Conference on Computer Vision*, pages 220–228. Springer. 17, 29

[78] Kennes, R. and Smets, P. (1990). Computational aspects of the Mobius transformation. In *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, pages 401–416. Elsevier Science Inc. 85

[79] Khan, S. M. and Shah, M. (2009). Tracking multiple occluding people by localizing on multiple scene planes. *IEEE transactions on pattern analysis and machine intelligence*, 31(3):505–519. 29, 44, 45, 56, 103

[80] Kim, K. and Davis, L. S. (2006). Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering. In *European Conference on Computer Vision*, pages 98–109. Springer. 44

[81] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. 112

[82] Kneip, L., Chli, M., and Siegwart, R. Y. (2011). Robust real-time visual odometry with a single camera and an imu. In *Proceedings of the British Machine Vision Conference 2011*. British Machine Vision Association. 16

[83] Kolmogorov, V. (2006). Convergent tree-reweighted message passing for energy minimization. *IEEE transactions on pattern analysis and machine intelligence*, 28(10):1568–1583. 55

[84] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105. 105

[85] Kumar, P., Mittal, A., and Kumar, P. (2010). Addressing uncertainty in multi-modal fusion for improved object detection in dynamic environment. *Information Fusion*, 11(4):311–324. 73

[86] Kurdej, M. (2014). BFT - Belief Functions Theory library. `https://github.com/mkurdej/bft`. Last accessed 2018-03-13. 74

[87] Kurdej, M., Moras, J., Cherfaoui, V., and Bonnifait, P. (2014). Controlling remanence in evidential grids using geodata for dynamic scene perception. *International Journal of Approximate Reasoning*, 55(1):355–375. 73, 74

[88] Labayrade, R., Gruyer, D., Royere, C., Perrollaz, M., and Aubert, D. (2007). *Obstacle detection based on fusion between stereovision and 2d laser scanner*. Pro Literatur Verlag. 73

[89] Le Hégarat-Mascle, S. and Seltz, R. (2004). Automatic change detection by evidential fusion of change indices. *Remote Sensing of Environment*, 91(3-4):390–404. 73

[90] Lezama, J., Grompone von Gioi, R., Randall, G., and Morel, J.-M. (2014). Finding vanishing points via point alignments in image primal and dual domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 509–515. xv, 49

[91] Li, M., Bao, S., Dong, W., Wang, Y., and Su, Z. (2013). Head-shoulder based gender recognition. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 2753–2756. IEEE. 104

[92] Li, T., Chang, H., Wang, M., Ni, B., Hong, R., and Yan, S. (2015). Crowded scene analysis: A survey. *IEEE transactions on circuits and systems for video technology*, 25(3):367–386. viii

[93] Liem, M. C. and Gavrila, D. M. (2014). Joint multi-person detection and tracking from overlapping cameras. *Computer Vision and Image Understanding*, 128:36–50. 44

[94] Lin, B., Johnson, A., Qian, X., Sanchez, J., and Sun, Y. (2013). Simultaneous tracking, 3d reconstruction and deforming point detection for stereoscope guided surgery. In *Augmented Reality Environments for Medical Imaging and Computer-Assisted Interventions*, pages 35–44. Springer. 31

[95] Lin, W.-Y., Cheong, L.-F., Tan, P., Dong, G., and Liu, S. (2012). Simultaneous camera pose and correspondence estimation with motion coherence. *International journal of computer vision*, 96(2):145–161. 15

[96] Lin, W.-Y., Liu, S., Jiang, N., Do, M. N., Tan, P., and Lu, J. (2016). Repmatch: Robust feature matching and pose for reconstructing modern cities. In *European Conference on Computer Vision*, pages 562–579. Springer. 15

[97] Ling, Y. and Shen, S. (2016). High-precision online markerless stereo extrinsic calibration. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 1771–1778. IEEE. 24

[98] Liu, Z., Monasse, P., and Marlet, R. (2014). Match selection and refinement for highly accurate two-view structure from motion. In *European Conference on Computer Vision*, pages 818–833. Springer. 15

[99] Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440. 110

[100] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comp. Vis.*, 60(2):91–110. 7, 18, 19

[101] Luitjens, J. (2014). Faster parallel reductions on kepler. 59

[102] Madrigal, F., Hayet, J.-B., and Rivera, M. (2015). Motion priors for multiple target visual tracking. *Machine Vision and Applications*, 26(2-3):141–160. 29

[103] Mahmoud, N., Hostettler, A., Collins, T., Soler, L., Doignon, C., and Montiel, J. M. M. (2017). SLAM based quasi dense reconstruction for minimally invasive surgery scenes. *ICRA 2017 workshop C4 Surgical Robots: Compliant, Continuum, Cognitive, and Collaborative*. 31

[104] Maier-Hein, L., Groch, A., Bartoli, A., Bodenstedt, S., Boissonnat, G., Chang, P.-L., Clancy, N., Elson, D. S., Haase, S., Heim, E., et al. (2014). Comparative validation of single-shot optical techniques for laparoscopic 3-d surface reconstruction. *IEEE transactions on medical imaging*, 33(10):1913–1930. 31

[105] Martin, A. (2014). Matlab toolbox for belief functions. `http://www.arnaud.martin.free.fr/Doc`. Last accessed 2018-03-13. 74

[106] Martinec, D. and Pajdla, T. (2007). Robust rotation and translation estimation in multiview reconstruction. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE. 14, 15

[107] Mavrinac, A. and Chen, X. (2013). Modeling coverage in camera networks: A survey. *International journal of computer vision*, 101(1):205–226. 13

[108] Mehmood, M. O., Ambellouis, S., and Achard, C. (2014). Ghost pruning for people localization in overlapping multicamera systems. In *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*, volume 2, pages 632–639. IEEE. 29

[109] Mercier, D., Quost, B., and Denoeux, T. (2008). Refined modeling of sensor reliability in the belief function framework using contextual discounting. *Information Fusion*, 9(2):246–258. 69

[110] Milan, A., Roth, S., and Schindler, K. (2014). Continuous energy minimization for multitarget tracking. *IEEE transactions on pattern analysis and machine intelligence*, 36(1):58–72. 29

[111] Mittal, A. and Davis, L. S. (2003). M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *Int. Journal of Computer Vision*, 51(3):189–203. 44

[112] Moisan, L. and Stival, B. (2004). A probabilistic criterion to detect rigid point matches between two images and estimate the fundamental matrix. *Int. J. Comp. Vis.*, 57(3):201–218. 8, 9, 14, 21, 25

[113] Mountney, P., Stoyanov, D., and Yang, G.-Z. (2010). Three-dimensional tissue deformation recovery and tracking. *IEEE Signal Processing Magazine*, 27(4):14–24. 25, 31

[114] Mountney, P. and Yang, G.-Z. (2009). Dynamic view expansion for minimally invasive surgery using simultaneous localization and mapping. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 1184–1187. IEEE. 31

[115] Mueller, G. R. and Wuensche, H.-J. (2016). Continuous extrinsic online calibration for stereo cameras. In *Intelligent Vehicles Symposium (IV), 2016 IEEE*, pages 966–971. IEEE. 24

[116] Munkres, J. (1957). Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38. 96, 118

[117] Nguyen, H. T. (2006). *An introduction to random sets*. Chapman and Hall/CRC. 67

[118] Ochoa, B. and Belongie, S. (2006). Covariance propagation for guided matching. In *Workshop on Statistical Methods in Multi-Image and Video Processing*. 15, 25, 27

[119] Ojala, T., Pietikäinen, M., and Harwood, D. (1996). A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59. 104

[120] Pellicano, N., Aldea, E., and Le Hégarat-Mascle, S. (2016). Robust wide baseline pose estimation from video. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 3820–3825. IEEE. xiv, 21, 27, 28, 33

[121] Pellicanò, N., Aldea, E., and Le Hegarat-Mascle, S. (2017). Geometry-Based Multiple Camera Head Detection in Dense Crowds. In *BMVC - 5th Activity Monitoring by Multiple Distributed Sensing Workshop*. xvi, 94, 97

[122] Pellicano, N., Aldea, E., and Le Hegarat-Mascle, S. (2017). Geometry-based multiple camera head detection in dense crowds. In *Proceedings of the 28th British Machine Vision Conference (BMVC) - 5th Activity Monitoring by Multiple Distributed Sensing Workshop*. 29

[123] Peng, P., Tian, Y., Wang, Y., Li, J., and Huang, T. (2015). Robust multiple cameras pedestrian detection with multi-view bayesian network. *Pattern Recognition*, 48(5):1760–1772. 29, 45

[124] Pichon, F., Destercke, S., and Burger, T. (2015). A consistency-specificity trade-off to select source behavior in information fusion. *IEEE transactions on cybernetics*, 45(4):598–609. 71, 84

[125] Pichon, F., Dubois, D., and Denoeux, T. (2012). Relevance and truthfulness in information correction and fusion. *Int. J. Approx. Reasoning*, 53(2):159–175. 69

[126] Pollefeys, M., Koch, R., and Van Gool, L. (1999). Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *International Journal of Computer Vision*, 32(1):7–25. 13

[127] Pollok, T. and Monari, E. (2016). A visual slam-based approach for calibration of distributed camera networks. In *13th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2016, Colorado Springs, CO, USA, August 23-26, 2016*, pages 429–437. 16

[128] Possegger, H., Sternig, S., Mauthner, T., Roth, P. M., and Bischof, H. (2013). Robust real-time tracking of multiple objects by volumetric mass densities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2395–2402. 45

[129] Puig, L. and Daniilidis, K. (2016). Monocular 3d tracking of deformable surfaces. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 580–586. IEEE. 31

[130] Radke, R. J. (2010). A survey of distributed computer vision algorithms. *Handbook of Ambient Intelligence and Smart Environments*, pages 35–55. 13

[131] Raguram, R., Chum, O., Pollefeys, M., Matas, J., and Frahm, J.-M. (2013). Usac: a universal framework for random sample consensus. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):2022–2038. 25

[132] Ravichandran, A. and Vidal, R. (2011). Video registration using dynamic textures. *Patt. Anal. Mach. Intell.*, 33(1):158–171. 17

[133] Reineking, T. (2014). Dempster-Shafer theory library. `https://pypi.python.org/pypi/py_dempster_shafer/0.7`. Last accessed 2018-03-13. 74

[134] Rekik, W., Le Hégarat-Mascle, S., Reynaud, R., Kallel, A., and Ben Hamida, A. (2016). Dynamic object construction using belief function theory. *Information Sciences*, 345:129 – 142. 73, 74, 75

[135] Remondino, F. and Fraser, C. (2006). Digital camera calibration methods: considerations and comparisons. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(5):266–272. 13

[136] Ristic, B. and Smets, P. (2006). The TBM global distance measure for the association of uncertain combat id declarations. *Information fusion*, 7(3):276–284. 96

[137] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer. 110

[138] SanMiguel, J. C., Micheloni, C., Shoop, K., Foresti, G. L., and Cavallaro, A. (2014). Self-reconfigurable smart camera networks. *IEEE Computer*, 47(5):67–73. 13

[139] Sekii, T. (2016). Robust, real-time 3d tracking of multiple objects with similar appearances. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4275–4283. 29, 45

[140] Shafer, G. (1976). *A mathematical theory of evidence*, volume 42. Princeton university press. 67, 69, 84

[141] Sheather, S. J. and Jones, M. C. (1991). A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 683–690. 29

[142] Sindagi, V. A. and Patel, V. M. (2017). Generating high-quality crowd density maps using contextual pyramid cnns. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1879–1888. IEEE. 104

[143] Smets, P. (1995). The canonical decomposition of a weighted belief. In *IJCAI*, volume 95, pages 1896–1901. 70

[144] Smets, P. (2005). Belief functions on real numbers. *International journal of approximate reasoning*, 40(3):181–223. 71

[145] Smets, P. (2008). Belief functions: the disjunctive rule of combination and the generalized bayesian theorem. In *Classic Works of the Dempster-Shafer Theory of Belief Functions*, pages 633–664. Springer. 69

[146] Smeulders, A. W., Chu, D. M., Cucchiara, R., Calderara, S., Dehghan, A., and Shah, M. (2014). Visual tracking: An experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1442–1468. 29

[147] Snavely, N., Seitz, S. M., and Szeliski, R. (2008). Modeling the world from internet photo collections. *Int. J. Comp. Vis.*, 80(2):189–210. 14, 15

[148] STEREOLABS (2018). *ZED Stereo Camera*. 24

[149] Sternig, S., Mauthner, T., Irschara, A., Roth, P. M., and Bischof, H. (2011). Multi-camera multi-object tracking by robust hough-based homography projections. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1689–1696. IEEE. 44

[150] Sui, L., Feissel, P., and Denœux, T. (2018). Identification of elastic properties in the belief function framework. *International Journal of Approximate Reasoning*. 75

[151] Sur, F., Noury, N., and Berger, M.-O. (2008). Computing the uncertainty of the 8 point algorithm for fundamental matrix estimation. In *19th British Machine Vision Conference-BMVC 2008*, page 10. 20

[152] Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., and Rother, C. (2008). A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE transactions on pattern analysis and machine intelligence*, 30(6):1068–1080. 55

[153] Tan, X., Sun, C., Sirault, X., Furbank, R., and Pham, T. D. (2015). Feature matching in stereo images encouraging uniform spatial distribution. *Pattern Recognition*, 48(8):2530–2542. 15, 26

[154] Tang, N. C., Lin, Y.-Y., Weng, M.-F., and Liao, H.-Y. M. (2015). Cross-camera knowledge transfer for multiview people counting. *IEEE Transactions on image processing*, 24(1):80–93. 29

[155] Tang, S., Andriluka, M., Milan, A., Schindler, K., Roth, S., and Schiele, B. (2013). Learning people detectors for tracking in crowded scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1049–1056. 29

[156] Tanzmeister, G., Thomas, J., Wollherr, D., and Buss, M. (2014). Grid-based mapping and tracking in dynamic environments using a uniform evidential environment representation. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 6090–6095. IEEE. 73

[157] Tola, E., Lepetit, V., and Fua, P. (2010). Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE transactions on pattern analysis and machine intelligence*, 32(5):815–830. 52

[158] Torr, P. H. and Zisserman, A. (2000). Mlesac: A new robust estimator with application to estimating image geometry. *Computer vision and image understanding*, 78(1):138–156. 8

[159] Totz, J., Mountney, P., Stoyanov, D., and Yang, G.-Z. (2011). Dense surface reconstruction for enhanced navigation in mis. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2011*, pages 89–96. 31

[160] Tran, T. A., Jauberthie, C., Le Gall, F., and Travé-Massuyès, L. (2018). Evidential box particle filter using belief function theory. *International Journal of Approximate Reasoning*, 93:40–58. 73, 99, 129

[161] Tsai, R. Y. (1987). A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *Robotics and Automation, IEEE Journal of*, 3(4):323–344. 18, 29

[162] Utasi, A. and Benedek, C. (2011). A 3-d marked point process model for multi-view people detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3385–3392. IEEE. 44

[163] Utasi, Á. and Benedek, C. (2013). A bayesian approach on people localization in multicamera systems. *IEEE transactions on circuits and systems for video technology*, 23(1):105–115. 29

[164] Vandoni, J., Le Hégarat-Mascle, S., and Aldea, E. (2018). Belief function definition for ensemble methods-application to pedestrian detection in dense crowds. In *21st International Conference on Information Fusion (FUSION)*. 104

[165] Vatti, B. R. (1992). A generic solution to polygon clipping. *Commun. ACM*, 35(7):56–63. 75

[166] Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE. 104

[167] Visentini-Scarzanella, M., Stoyanov, D., and Yang, G.-Z. (2012). Metric depth recovery from monocular images using shape-from-shading and specularities. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pages 25–28. IEEE. 31

[168] Vogiatzis, G., Esteban, C. H., Torr, P. H., and Cipolla, R. (2007). Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2241–2246. 53

[169] Walley, P. (1991). Statistical reasoning with imprecise probabilities. 67

[170] Wang, B., Wang, G., Chan, K. L., and Wang, L. (2017). Tracklet association by online target-specific metric learning and coherent dynamics estimation. *IEEE transactions on pattern analysis and machine intelligence*, 39(3):589–602. 29

[171] Wu, S., Wong, H.-S., and Yu, Z. (2014). A bayesian model for crowd escape behavior detection. *IEEE transactions on circuits and systems for video technology*, 24(1):85–98. 29

[172] Wu, Y., Lim, J., and Yang, M.-H. (2015). Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1834–1848. 29

[173] Xiao, C.-B., Feng, D.-Z., and Yuan, M.-D. (2016). An efficient fundamental matrix estimation method for wide baseline images. *Pattern Analysis and Applications*, pages 1–10. 15

[174] Xiong, F., Shi, X., and Yeung, D.-Y. (2017). Spatiotemporal modeling for crowd counting in videos. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 5161–5169. IEEE. 104

[175] Xu, P., Davoine, F., and Denoeux, T. (2014). Evidential combination of pedestrian detectors. In *British Machine Vision Conference*, pages 1–14. 104

[176] Xu, P., Davoine, F., Zha, H., and Denoeux, T. (2016). Evidential calibration of binary svm classifiers. *International Journal of Approximate Reasoning*, 72:55–70. 21

[177] Ye, M., Giannarou, S., Meining, A., and Yang, G.-Z. (2016). Online tracking and retargeting with applications to optical biopsy in gastrointestinal endoscopic examinations. *Medical image analysis*, 30:144–157. 31

[178] Ye, M., Giannarou, S., Patel, N., Teare, J., and Yang, G.-Z. (2013). Pathological site retargeting under tissue deformation using geometrical association and tracking. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 67–74. Springer. 31

[179] Yedidia, J. S., Freeman, W. T., Weiss, Y., et al. (2000). Generalized belief propagation. In *NIPS*, volume 13, pages 689–695. 55

[180] Yu, F. and Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*. xvi, 106

[181] Zadeh, L. A. (1999). Fuzzy sets as a basis for a theory of possibility. *Fuzzy sets and systems*, 100(1):9–34. 67, 69

[182] Zagoruyko, S. and Komodakis, N. (2015). Learning to compare image patches via convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 4353–4361. IEEE. 7

[183] Zair, S. and Le Hégarat-Mascle, S. (2017). Evidential framework for robust localization using raw GNSS data. *Engineering Applications of Artificial Intelligence*, 61:126 – 135. 73, 74, 75, 84, 90

[184] Zamir, A. R., Dehghan, A., and Shah, M. (2012). Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *Computer Vision–ECCV 2012*, pages 343–356. Springer. 29

[185] Zhang, Z. (1998a). Determining the epipolar geometry and its uncertainty: A review. *Int. J. Comput. Vision*, 27(2):161–195. 10, 11

[186] Zhang, Z. (1998b). Determining the epipolar geometry and its uncertainty: A review. *Int. J. Comp. Vis.*, 27(2):161–195. 20, 21

[187] Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., and Torr, P. H. (2015). Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1529–1537. 128

Titre : Une approche réaliste de la détection de piétons multi-vues et multi-représentations pour des scènes extérieures.

Mots clés : Multi-vues, fusion de données, fonctions de croyance, analyse des foules, détection de têtes

Résumé : La détection et le suivi de piétons sont devenus des thèmes phares en recherche en Vision Artificielle, car ils sont impliqués dans de nombreuses applications. La détection de piétons dans des foules très denses est une extension naturelle de ce domaine de recherche, et l'intérêt croissant pour ce problème est lié aux évènements de grande envergure qui sont, de nos jours, des scenarios à risque d'un point de vue de la sûreté publique. Par ailleurs, les foules très denses soulèvent des problèmes inédits pour la tâche de détection. De par le fait que les caméras ont le champ de vision le plus grand possible pour couvrir au mieux la foule les têtes sont généralement très petites et non texturées. Dans ce manuscrit nous présentons un système complet pour traiter les problèmes de détection et de suivi en présence des difficultés spécifiques à ce contexte. Ce système utilise plusieurs caméras, pour gérer les problèmes de forte occultation. Nous proposons une méthode robuste pour l'estimation de la position relative entre plusieurs caméras dans le cas des environnements requérant une surveillance. Ces environnements soulèvent des problèmes comme la grande distance entre les caméras, le fort changement de perspective, et la pénurie d'information en commun. Nous avons alors proposé d'exploiter le flot vidéo pour effectuer la calibration, avec l'objectif d'obtenir une solution globale de bonne qualité. Nous proposons aussi une mèthode non supervisée pour la détection des piétons avec plusieurs caméras, qui exploite la consistance visuelle des pixels à partir des différents points de vue, ce qui nous permet d'effectuer la projection de l'ensemble des détections sur le plan du sol, et donc de passer à un suivi 3D. Dans une troisième partie, nous revenons sur la détection supervisée des piétons dans chaque caméra indépendamment en vue de l'améliorer. L'objectif est alors d'effectuer la segmentation des piétons dans la scène en partant d'une labélisation imprécise des données d'apprentissage, avec des architectures de réseaux profonds. Comme dernière contribution, nous proposons un cadre formel original pour une fusion de données efficace dans des espaces 2D. L'objectif est d'effectuer la fusion entre différents capteurs (détecteurs supervisés en chaque caméra et détecteur non supervisé en multi-vues) sur le plan du sol, qui représente notre cadre de discernement. nous avons proposé une représentation efficace des hypothèses composées qui est invariante au changement de résolution de l'espace de recherche. Avec cette représentation, nous sommes capables de définir des opérateurs de base et des règles de combinaison efficaces pour combiner les fonctions de croyance. Enfin, notre approche de fusion de données a été évaluée à la fois au niveau spatial, c'est à dire en combinant des détecteurs de nature différente, et au niveau temporel, en faisant du suivi évidentiel de piétons sur de scènes à grande échelle dans des conditions de densité variable

Titre : Tackling pedestrian detection in large scenes with multiple views and representations.

Mots clés : Multiple sensors, data fusion, belief functions, crowd analysis, head detection

Résumé : Pedestrian detection and tracking have become important fields in Computer Vision research, due to their implications for many applications, e.g. surveillance, autonomous cars, robotics. Pedestrian detection in high density crowds is a natural extension of such research body, and has a growing interest since large scale events are, nowadays, critical scenarios from a safety point of view. High density crowds introduce novel problems to the detection task. First, clutter and occlusion problems are taken to the extreme, so that only heads are visible, and they are not easily separable from the moving background. Second, heads are usually small and with little or no textures. This comes out from two independent constraints, the need of one camera to have a field of view as high as possible, and the need of anonymization. In this work we develop a complete framework in order to handle the pedestrian detection and tracking problems by using multiple cameras. As a first contribution, we propose a robust method for camera pose estimation in surveillance environments. We handle problems as high distances between cameras, large perspective variations, and scarcity of matching information, by exploiting an entire video stream to perform the calibration, in such a way that it exhibits fast convergence to a good solution. As a second contribution, we propose an unsupervised multiple camera detection method which exploits the visual consistency of pixels between multiple views in order to estimate the presence of a pedestrian. One is capable of jointly estimating the presence of a pedestrian and its height, allowing for the projection of detections on a common ground plane, and thus allowing for 3D tracking. In the third part, we study different methods in order to perform supervised pedestrian detection on single views. We aim to build a dense pedestrian segmentation of the scene starting from spatially imprecise labeling of data, since their extraction is unfeasible in a dense crowd. Most notably, deep architectures for semantic segmentation are studied and adapted to the problem of small head detection in cluttered environments. As last but not least contribution, we propose a novel framework in order to perform efficient information fusion in 2D spaces. The final aim is to perform multiple sensor fusion (supervised detectors on each view, and an unsupervised detector on multiple views) at ground plane level, that is, thus, our discernment frame. Since the space complexity of such discernment frame is very large, we propose an efficient compound hypothesis representation which has been shown to be invariant to the scale of the search space. Through such representation, we are capable of defining efficient basic operators and combination rules of Belief Function Theory. Finally, we demonstrate our information fusion approach both at a spatial level, i.e. between detectors of different natures, and at a temporal level, by performing evidential tracking of pedestrians on real large scale scenes in sparse and dense conditions.