Capacités audiovisuelles en robot humanoïde NAO

Jordi Sanchez-Riera

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **IMAGERIE, VISION ET ROBOTIQUE**

Arrêté ministérial :

Présentée par

Jordi Sanchez-Riera

Thèse dirigée par **Radu Horaud**

préparée au sein **Laboratoire Jean Kuntzman(LJK) - INRIA Rhône Alpes** et de **Mathématiques, Sciences et Technologies de l'Information, Informatique**

Developing Audio-Visual capabilities of humanoid robot NAO

Thèse soutenue publiquement le **14 Juin 2013**,
devant le jury composé de :

Dr., Peter Sturm
INRIA, Président
Dr., Crisitian Sminchisescu
Lund University, Rapporteur
Dr., Vaclav Hlavac
CTU Prague, Rapporteur
Dr., Rodolphe Gelin
Aldebaran Robotics, Examinateur
Dr., Radu Horaud
INRIA, Directeur de thèse

Abstract

Humanoid robots are becoming more and more important in our daily lives due the high potential they have to help persons in different situations. To be able to aid, a human-robot interaction is essential and to this end, it is important to use as well as possible, the external information collected by the different sensors of the robot. Usually most relevant sensors for perception are cameras and microphones, which provide very rich information about the world. In this thesis, we plan to develop applications towards human-robot interaction and to achieve a more natural communication when interacting with the robot. Taking advantage of the information provided by the cameras and microphones of NAO humanoid robot, we present new algorithms and applications using these sensors. With the visual information we introduce two different stereo algorithms, that will serve as a basis to design other applications. The first stereo algorithm is designed to avoid problems with textureless regions using information from images in different temporal instances. The second stereo algorithm, sceneflow, is designed to provide a more complete understanding of a scene, adding optical flow information in the computation of disparity. Indeed, position and velocity vector is available for each pixel. This provides a basis to start developing more high-level applications to a certain extent of interaction. Using the sceneflow algorithm, a descriptor is designed for action recognition. As a result, action recognition benefits from richer information in opposition to traditional monocular approaches, giving robustness to background clutter and disambiguating depth actions like 'punch'. To complement and improve the performance in action recognition, auditory information is added. It is well known that auditory data is complementary to the visual data and can be helpful in situations where objects are occluded or simply are not there. Finally, a last application developed towards a better human-robot interaction is a speaker detector. This can be used, for example, to center camera images to the speaking person (person of interest) and collect more reliable information. Here data from video and audio is also used, but the principle is completely different: from the visual and auditory features used to the way that these features are combined.

Résumé

Les robots humanoïdes sont de plus en plus important dans nos vies quotidiennes en raison du fort potentiel qu'ils ont pour aider les personnes. Pour être en mesure d'aider, il est nécessaire que le robot peut communiquer avec les humains, et pour cela, il est l'information importante du monde collectées par les capteurs intégrés au robot. Dans notre cas particulier, le rellevant la plupart sont des caméras et des micros, qui peuvent fournir une description assez complète de l'environnement du robot. Dans cette thèse, nous avons l'intention d'utiliser les informations fournies par les caméras et les micros de robot humanoïde Nao de développer des applications qui permettent une interaction homme-robot. Avec l'information visuelle deux algorithmes différents stéréo, qui serviront de base pour concevoir d'autres applications, sont présentés. La première utilise des informations provenant framse temporelle différente de surmonter certains problmes avec les régions sans texture, tandis que la deuxième chaîne hi-fi et le flux optique sont recherchées en même temps afin d'avoir plus d'informations sur la scène. Dans les vecteurs de béton, de position et de vitesse pour chaque pixel. Est le dernier algorithme que le descripteur est conçu pour la reconnaissance d'actions avec des données stéréo. Le but de cela est de tirer parti de l'information supplémentaire qui peut fournir l'stéréo comme en face de traditionnels algorithmes monoculaires qui existent à ce jour. Pour compléter et améliorer le taux de reconnaissance moyen de la reconnaissance d'actions, l'information auditive est également utilisé. Il est bien connu que les données provenant visuelle et capteurs auditifs est complémentaire et peut aider dans des situations où des objets sont caché ou ne sont tout simplement pas là. Enfin, une dernière application vers une meilleure interaction entre l'humain et le robot est un détecteur de haut-parleur. en ce cas, les données des deux modalités est également utilisé, mais il en diffère sur la manière dont les informations sont combinées, ainsi que les informations extraites de capteurs visuels et auditifs. Presque la totalité des applications sont mises en œuvre et exécuter en robot humanoïde NAO.

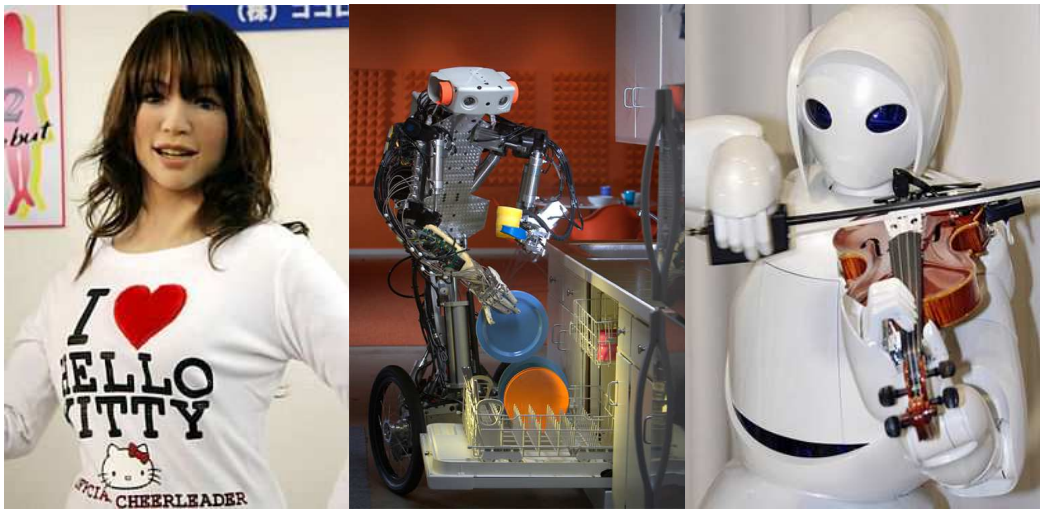
Contents

Chapter 1

Introduction

Robots are becoming more and more popular. They have been used widely in the past for industrial purposes, but new advances are facilitating to introduce them in other fields, like medical environments such as surgery room. Nowadays, several projects to build robots that can help and collaborate with humans are being developed. Exist the belief that robots can be of a great help when a good communication can become a fact. Social Robotics is a field that tries to address the interaction between persons and propose methodologies for a proper interaction. Here we can include what is called human-robot interaction. Probably the first the first idea that comes to our head is a humanoid robot, but other shape exists as well. For example, and "owl" like robot, or a driving assistant robot can be considered into this category.

One of the purposes of this thesis is to work with humanoid robot NAO (presented in more detail later) and to provide it with social skills. This is linked directly with the social robotics. Which skills we should implement?, what it should be able to do?, what it would be a good interaction? etc. are questions that we will try to address in this manuscript. One of the basis of a humanoid robot is that resembles to a human, not only in shape but also in perception abilities. Starting from visual and hearing sensors, such RGB cameras and microphones, the main goal is to develop algorithms for the robot that will provide perception capabilities in order to achieve some level of interaction.



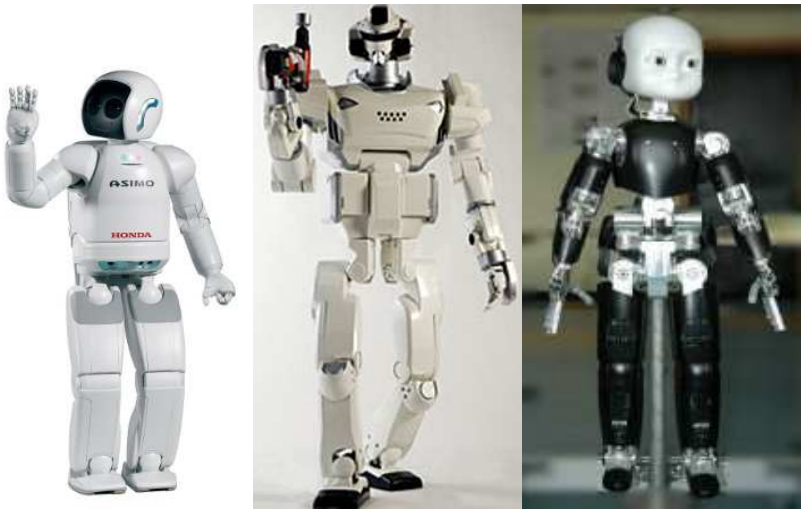
(a) Actroid (b) (c)

Figure 1.1: Different models of humanoid robots.

1.1 Humanoid Robots

Humanoid Robots seems the best to interact with humans because we tend to feel more relaxed and ease in front of something with familiar characteristics and resemblance to us. The potentialities to have a similar behavior is another factor. Several humanoid robots have been developed for different purposes. From help the astronauts in the international space station to the ones that can play instruments. For this, it is important that humans and robots can work together, in a complex situations where extreme natural environments exists or as a partners as they need to interact.

The robots can be designed a bit different giving them different features depending on the application they are supposed to tackle. For example, in Figure 1.1 we see different examples of specialization. The actroid has highly develop facial expression to seem more human and have more empathy with the interacting people, while the robot that brings the dishes to the dishwasher has more develop functional hands. Other case is when playing the violin, where high precision is needed on the hand to be on tune and produce the several tones. Note that in this example, legs are not fundamental for any of the cases.



(a) ASIMO-Honda (b) HRP-Kawada (c) iCub

Figure 1.2: Different models of humanoid robots.

Not only specialized robots exists, but also more general purpose robots. Some examples are shown in Figure 1.2. The famous Asimo from Honda is a robot that can walk, climb stairs, etc. Not only is mechanically advanced, but also it incorporates cameras and microphones for perception. Is the same case of HRP robot from Kawada Industries, designed to help humans with advanced mechanics and a vision system. The last example is the iCub, created by a consortium of different countries, more intended to explore the cognitive system part. It is equipped with visual sensors and microphones for interaction with persons, where mechanics are not as much advanced as in the other robots. However, it is also able to grasp objects.

Smaller and accessible (in terms of cost) robots are also developed to work in mechanical aspects or in perception aspects or both. Some of them are shown in Figure 1.3. These robots present similar characteristics at perception level with cameras and microphones, but also at mechanical level.

Robots existing from different companies have different hardware, even within several models of the same companies also have different hardware. This means that the software developed for one model, can not be ported to an other model,

(a) HOAP (b) DARWIN (c) NAO

Figure 1.3: Different models of humanoid robots.

no to talk to another robot from other company. Is for this reason that usually code is not developed for a concrete version of hardware and a middle-ware software is commonly used. A middle-ware can be understood as a software tool that abstracts the hardware level avoiding to have to work directly with it. Most known middle-ware is Robot Operating System (ROS)[1] but we can find others such as Robotics Service Bus (RSB)[2].

Other efforts have been made in the compatibility software direction for interaction with robots. For example, the Behavior toolkit, that includes *perception* (fusion data between several sensors), *cognition* (handles internal and external information to activate some triggers) and *memory* (stores behavioral specifications from social-scientific literature) features. Some applications using that toolkit can be found in Huang & Mutlu (2012).

[1] www.ros.org
[2] https://code.cor-lab.de/projects/rsb

1.2 HUMAVIPS FP7 European Project

Imagine a robot in a museum that is guiding a group of people. The robot is explaining some paintings and if the middle of the explanation using verbal, vision, and auditory cues, proposes a question to the group. From the replies the robot is able to identify a visitor to engage a different conversation or continue with the explanation. This scenario is implemented in Yamazaki *et al.* (2012). The goal of HUMAVIPS project is that a robot can interact with people in a complex environment similarly to the before mentioned scenario. To this end several ways to extract information from the scene are needed. Most important could be vision, hearing and speaking, but others as tactile abilities could also be considered. NAO robot, is a humanoid robot developed by Aldebaran[1] and it incorporates two cameras, four microphones, two speakers, an inertial measurement unit (gyro-meter, accelerometer), eight force-sensing resistors and two bumpers. From all these sensors, most useful for interaction and perception are the cameras and microphones. However, the cameras are located in a way that no overlap exist between images. Stereo is an important cue for navigation and other multiple applications and this is why the version of NAO head is modified. More details on this modified head are given in Chapter 7.

1.2.1 Developing Audio-Visual capabilities of NAO

The perception part of the robot (vision and audition) will center the focus of study. Because a lot of work exists with vision only (pose detection, action and object recognition, etc.), it can seem reasonable to concentrate only on the vision part, but we can soon realize that auditory information can be also rich and complement with visual information. Moreover, there are some applications, e.g. sound recognition that obviously, auditory information is very important. Using the information available from NAO sensors, methods and algorithms that will be used later on for interaction, are presented.

[1] http://www.aldebaran-robotics.com/en/

Using both cameras, two stereo algorithms are developed: *temporal stereo* and *Sceneflow*. Both methods are based on a seed growing algorithm which is a compromise between a global and a local method. The *temporal stereo*, explained in Chapter 2, is a convenient method to have stereo when texture-less regions appears in some frames for short amount of time. The principle is that some frames can have no texture because this would be compensated by other frames that contains texture. Unfortunately, match spatio-temporal information is not as easy as it seems, due the movement of an object is seen differently by both cameras. A method is proposed to overcome these difficulties.

A disparity map contains only static information, but usually scenes are not static. In this sense a method that estimates jointly disparity and optical flow is important to estimate also the movement. Since disparity and optical flow constrains each other, now the seed growing algorithm is modified as explained in Chapter 3, to obtain the *Sceneflow*.

This last algorithm serve as a start point to introduce higher level algorithms and start to get some real interaction. In Chapter 4, a descriptor for action recognition that uses the information provided by *Sceneflow* is described. This kind of information gives us a double advantage: from one side we have an easy segmentation of the scene, on the other side, on those actions that depth is involved, e.g. punch, are more easy to identify, contrary to the monocular images where ambiguities are not possible to solve.

Not only stereo is useful to solve ambiguities. Auditory information can be used too. To have better results on action recognition both modalities (vision and auditory) are used together. Due the complementary nature of sound and video, the results of the combination highly improve those that only use one of the modalities. How the fusion is carried out is explained in Chapter 5.

A first application towards the robot human interaction is just explained. However, multiple ways of combine auditory and visual information as well as other applications can be thought. This is the case of Chapter 6. Typically in action recognition the person is just in the center of the image, unfortunately this is not always true. This chapter presents an application to detect visible speaking persons, that at the same time the robot moves the head to put the speaking

person in the center. This could be exploited to center a person when performing some action, but it is also interesting enough as stand alone application for a more interactive communication towards the robot.

Finally, in Chapter 7, is explained how all these algorithms are actually translated to run in NAO. A description of the hardware used as well as the software to interface with the robot is described. Then a stereo method based on seed growing, the audio-visual action recognition and the audio-visual speaker detection is presented describing the modular structure used that allow to reuse each of the components for other applications.

Chapter 2

Temporal Stereo

2.1 Introduction

Stereo vision is an important ingredient of robot navigation, path planning, obstacle avoidance, or object grasping, because it provides depth, and photometric information of the real world captured with a pair of cameras. For example Talukder & Matthies (2004) proposes a real-time dense stereo combined with optical-flow to get a faithful representation of the scene and to yield a comprehensive dynamic scene analysis which, combined with egomotion estimation, provides accurate information in order to navigate in an unknown environment. Similarly, Ess *et al.* (2009) combines a stereo algorithm with machine learning techniques to support path planning algorithms for the avoidance of dynamic obstacles. Another representative example can be find in Marks *et al.* (2008) where stereo is used for visual SLAM. Also when grasping an object with an articulated arm the perception of depth is fundamental, *e.g.* Leeper *et al.* (2010).

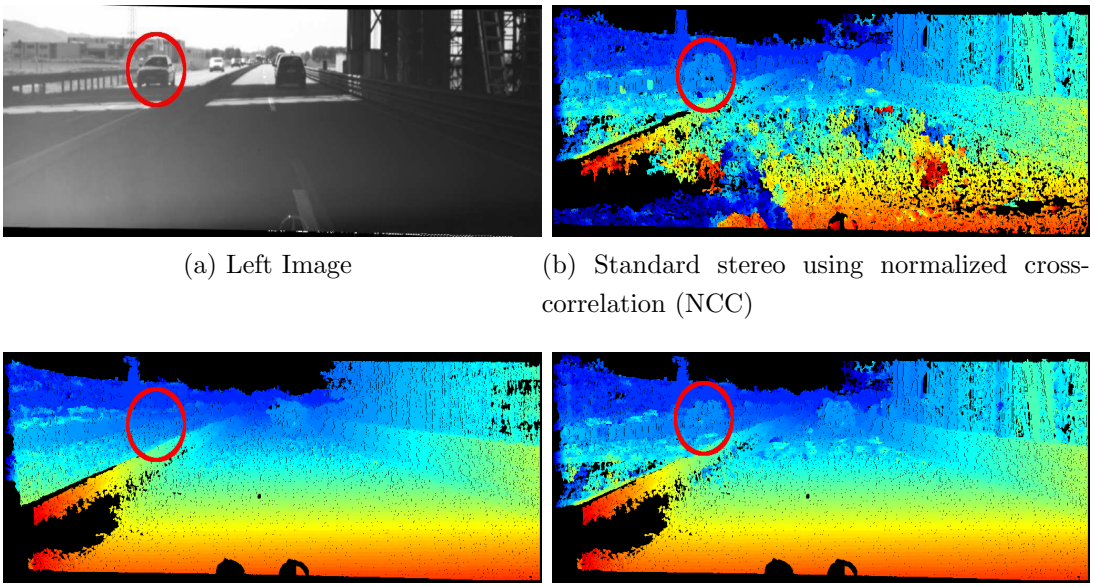
Therefore, many robotic applications use stereo to extract reliable information of the real world. However, the stereoscopic matching itself is a difficult ill-posed problem. This is especially true in unconstrained outdoor environments where devices like time-of-flight (TOF) cameras or other active sensors based on structured light (like Kinect™) cannot operate. The major difficulty of stereo is an intrinsic ambiguity when matching pixels. This is due to a weak or repetitive

texture in the scene, low or unstable illumination, or various kind of noise present in the images. To this end, the use of temporal information processing a video sequence of stereo images (as opposed to processing stereo images frame-by-frame independently) can be valuable to mitigate the matching ambiguity. In literature, there are several approaches to integrate the extra temporal information into stereo.

In a first category there are algorithms that compute the *scene flow*, namely the simultaneously estimation of depth and motion. The formulation of the coupled estimation of disparity (between a stereo pair) and of optical flow (between consecutive frames) mutually constrain each other. This task is traditionally solved by variational methods Basha *et al.* (2010); Huguet & Devernay (2007), by MRF methods Isard & MacCormick (2006); Liu & Philomin (2009), or by seed-growing methods Čech *et al.* (2011), to cite just a few.

In a second category, there are methods which rely on *independent motion estimates* to improve the stereo matching. A straightforward algorithm Stankiewicz & Wegner (2010) enforces the temporal consistency by detecting moving regions via background subtraction. The disparities in the static regions are averaged over time by a linear low-pass filter, while the disparities of the moving are estimated frame-by-frame. In Bleyer & Gelautz (2009), the disparity maps are filtered by a median filter along pixel trajectories obtained by an external optical flow module. Independent optical flow is also used in Zhu *et al.* (2010), where the authors propose an MRF framework with an extra term which penalizes discrepancies in photo-consistency of the (optical flow related) neighbourhood in the current-, previous-, and next frames. Reference Larsen *et al.* (2007) proposes a similar MRF formulation, but additionally they dynamically disconnect the edges to prevent over-smoothing in case of large motion and failure of the optical flow estimates.

A third category is composed of methods of *spatiotemporal stereo* that do not estimate motion explicitly, but exploit a spatiotemporal neighbourhood $(2D + t)$ around an image location in order to increase the discriminability of the similarity statistics. For example Davis *et al.* (2005) projects an artificial pattern varying over time, onto the scene. Temporal aggregation of the statistics significantly



(a) Left Image

(b) Standard stereo using normalized cross-correlation (NCC)

(c) Spatiotemporal stereo (TNCC)

(d) Robust spatiotemporal stereo (RTNCC)

Figure 2.1: Disparity maps from the DAGM 2011 Exposure Challenge dataset. (a) Left input image. (b) Standard stereo matching based on normalized cross-correlation (NCC) without any temporal aggregation; notice that there are large errors in the road regions corresponding to bad illumination conditions. (c) Trivially averaged NCC over several time frames (TNCC); the car (circled) was filtered out. (d) Robust spatiotemporal stereo matching method proposed in this paper (RTNCC); notice that the disparity was correctly estimated both around the car and on the road. Warmer colours are closer to the camera. Black encodes unmatched pixels.

disambiguates the matching in weakly textured regions. However, this is an active system assuming a static camera and a static scene. The similarity statistic (based on bilateral filtering) is temporally aggregated also in Richardt *et al.* (2010), such that adjacent frames are weighted by a Gaussian kernel to make the central frame the most informative one and to reduce the influence of more distant frames to cope with a small motion. In Zhang *et al.* (2003) the authors give an insight into how spatio-temporal windows are deformed due to surface slant and motion and propose an optimization framework to find the distortion parameters and construct similarity statistics invariant to a small motion. Alternatively, the same insensitivity is achieved in Sizintsev & Wildes (2009) by representing the image using Gabor filter responses and the similarity statistic is computed in a closed form without iterative optimization. However, all these methods assume that the disparity map between frames changes only slowly. In reality this assumption is not valid near object boundaries, for rapidly moving objects, which cause serious artifacts.

The first approach to improve the stereo is a new spatiotemporal stereo matching method which benefits from aggregating the similarity statistic over a $2D + t$ window. Unlike previous work on spatiotemporal stereo algorithms, where dealing with rapidly moving objects is problematic because of the difficulties to match $2D + t$ patterns, the proposed method is robust to abrupt temporal changes in disparity due to large motions and at the same time benefits from the extra information provided by temporal information whenever it is possible. The main idea of our algorithm is to automatically detect the image regions corresponding to this phenomena (large change in disparity in time), such that the aggregating of the similarity statistic over the time window is disconnected for these regions in order to prevent typical artifacts in moving part of the scene, *e.g.* blurred contours or even missing the objects completely.

In Figure 2.1, there are resulting disparity maps computed using three different similarity statistics on a frame from DAGM 2011 Challenge Exposure Changes dataset, Figure 2.1a. In Figure 2.1b, no temporal information is used (the matching algorithm works with a single stereo-pair of images). We can observe that the output contains serious errors in weakly illuminated road. In Figure 2.1c,

a similarity statistic was trivially averaged over time. Notice that in that case the errors on the road are strongly reduced, but the car going on the opposite lane was filtered out completely. Finally in Figure 2.1d, there is an output generated using the proposed similarity statistic, where errors on the road are reduced significantly and the car going in the opposite direction is not missed.

2.2 Robust Spatiotemporal Stereo

We consider a sequence of stereoscopic images captured by calibrated and synchronized cameras. We assume the images epipolar rectified $\mathbf{I_l}(x, y, t)$ and $\mathbf{I_r}(x, y, t)$, where (x, y) is the horizontal-vertical location of a pixel and t is a time parameter (a frame). The sequences are related by an unknown disparity function $d(x, y, t)$ which assigns the correspondences between pixels in the left and right image

$$\mathbf{I_l}(x, y, t) \approx \mathbf{I_r}(x + d(x, y, t), y, t). \tag{2.1}$$

A matching algorithm must compute a certain similarity statistic between potentially corresponding pixels to measure how the corresponding image locations are photometrically consistent and consequently how likely their matching is.

In the next subsection 2.2.1 we discuss various similarity statistics defined over a spatiotemporal neighbourhoods, and present the proposed robust similarity statistic. Then, in the subsection 2.2.2 we describe how the similarity statistic is integrated into a matching algorithm which finally assigns the correspondences and output the disparity map. In our case, this is an efficient seed growing algorithm adopted from Čech *et al.* (2010).

2.2.1 Similarity statistic

The simplest image similarity statistic is a difference of pixel intensities, which is however ambiguous. More discriminable statistics use a small neighbourhood (a window) around potentially corresponding pixels in the images. Then, these algorithms locally approximate the disparity function in (2.1). For instance,

paper Zhang *et al.* (2003) uses a linear approximation by the first order Taylor expansion. In a small spatiotemporal neighbourhood $\mathcal{N}$ around location (x_0, y_0, t_0), *e.g.* a 3D window of 5×5 pixels over 3 frames, the disparity function is $d(x, y, t) \approx \hat{d}(d_i, d_0, d_1, d_2, d_t) = d_i + d_0 + d_1 \cdot (x - x_0) + d_2 \cdot (y - y_0) + d_t \cdot (t - t_0)$. Then they use an optimized statistic to measure a photometric consistency of the potential correspondence for candidate (integer) disparities d_i

$$\mathrm{TSSD}(x_0, y_0, t_0, d_i) = \min_{d_0, d_1, d_2, d_t} \sum_{(x,y,t) \in \mathcal{N}} \Big(\mathbf{I_l}(x, y, t) - \mathbf{I_r}(x + \hat{d}(d_i, d_0, d_1, d_2, d_t), y, t) \Big)^2 \tag{2.2}$$

to compensate the distortion which occurs due to sub-pixel displacement d_0, surface slant d_1, d_2, and temporal disparity change d_t.

However, there are several sources of errors in this approach: (i) Tendency to get stuck in a local extrema; (ii) Not a significant gain in discriminability[1] over the case where $d_0 = d_1 = d_2 = d_t = 0$, since the statistic is improved by the optimization for both correct and incorrect matches; (iii) When the assumption on the linearity of the disparity function within the local spatiotemporal neighbourhood is violated (*e.g.* abrupt change in disparity), the method fails dramatically.

Therefore we adopted a simpler approach which assumes a fronto-parallel surface undergoing a motion that preserves the constant disparity, however our proposed statistic is fairly insensitive to small violation of this assumption. We will show that the discriminability is comparable or even higher over the optimization framework (2.2). Similar 'over-fitting' effect of the discriminability loss of too complex model has been reported in *e.g.* Shi & Tomasi (1994). When the constant disparity assumption is violated, our proposed statistic automatically switches off the temporal aggregation and avoids the artifacts.

As an elementary similarity statistic, we use Moravec normalized cross correlation Moravec (1977). It has several favorable properties compared to the sum

[1] The discriminability of the similarity statistic is proportional to a probability that the statistic has better response for the true correspondence than for the incorrect ones.

of squared differences. It is defined as

$$\mathrm{NCC}(x_0, y_0, t_0, d_i) \quad = \quad \frac{2\ \mathrm{cov}\big(\mathbf{W_l}(x_0, y_0, t_0), \mathbf{W_r}(x_0 + d_i, y_0, t_0)\big)}{\mathrm{var}\big(\mathbf{W_l}(x_0, y_0, t_0)\big) + \mathrm{var}\big(\mathbf{W_r}(x_0 + d_i, y_0, t_0)\big) + \epsilon}, \quad (2.3)$$

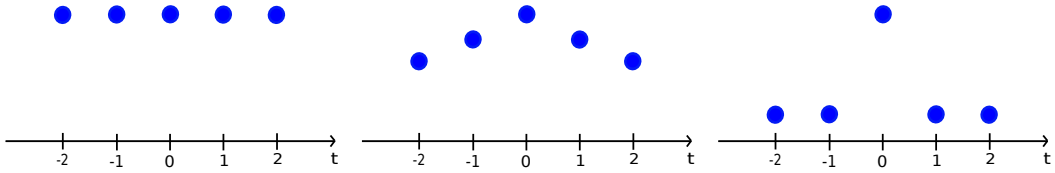
where $\mathbf{W_l}(x_0, y_0, t_0) = \mathbf{I_l}(x_0 - N : x_0 + N, y_0 - N : y_0 + N, t_0)$ is a spatial window (a sub-image) of $(2N + 1) \times (2N + 1)$ pixels centered at position (x_0, y_0) of the frame t_0 of the left image sequence. Window $\mathbf{W_r}$ is defined similarly, and ϵ is a machine epsilon to prevent instability of the statistic in case of low intensity variance. The statistic has consequently low response in textureless regions. Its range is limited in $[-1, +1]$. Unlike a standard correlation coefficient, this statistic is not completely invariant to affine transformations of data, but insensitive only. It is also reported quite insensitive to a small surface slant when the window is small Čech *et al.* (2010).

Then the NCC statistic is aggregated over a symmetric time window of $2T+1$ frames, such that

$$\mathrm{TNCC}(x_0, y_0, t_0, d_i) = \frac{1}{2T + 1} \sum_{t=t_0-T}^{t_0+T} \mathrm{NCC}(x_0, y_0, t, d_i). \quad (2.4)$$

Apparently, the TNCC statistic is decayed when the disparity changes significantly within the temporal window. Notice that the motion in general is not harmful, but the motion changing the disparity is. In Figure 2.2, we illustrate a typical distribution of $\mathrm{NCC}(t)$ statistic over the time t of the window for a correct match $(x_0, y_0, t = 0, d_i)$. If the disparity is constant over time, all per-frame correlations for $t = \{-T, \ldots, T\}$ are high, Figure 2.2a. If the disparity changes slowly, the correlation is slightly lower more faraway from the central frame, Figure 2.2b. However, when the disparity changes rapidly, the correlations off the central frame drop quickly, Figure 2.2c, since the other correlations measure a photometric consistency of locations which are not corresponding any more.

On the other hand, a potential mismatch (*i.e.* wrong correspondence) has the distribution of per-frame correlations over the time window such that the correlations are low, but due to random fluctuations or texture self-similarity there may be high responses for any frame of the temporal window. The temporal



(a) constant disparity (b) small change in disparity (c) large change in disparity

Figure 2.2: Distributions of per-frame correlations over time. The match is correct at time $t = 0$. (a) If disparity is constant over time, all per-frame correlations are high. (b) When there is a small temporal change in disparity, all per-frame correlations around the central one ($t = 0$) have a lower value. (c) When there is a large change in disparity, the correlations around the central frame tend close to zero.

aggregation in (2.4) averages out these excesses and decreases their correlations and hereby increases the discriminability.

However, it is important to detect phenomena in Figure 2.2c corresponding to large changes in disparity and in these cases to use the central correlation only without any aggregation which would cause artifacts. Therefore, we propose a robust temporal normalized cross correlation

$$\mathrm{RTNCC}(x_0, y_0, t_0, d_i) = \begin{cases} \mathrm{NCC}(x_0, y_0, t_0, d_i) & \text{if } \big(\mathrm{NCC}(t_0) - \mathrm{NCC}(t_0 \pm 1)\big) \geq \alpha, \\ \mathrm{TNCC}(x_0, y_0, t_0, d_i) & \text{otherwise.} \end{cases} \tag{2.5}$$

This means that RTNCC uses the correlation (2.3) of the central frame $\mathrm{NCC}(t_0)$, if it is higher than correlations of adjacent frames $\mathrm{NCC}(t_0 + 1)$ and $\mathrm{NCC}(t_0 - 1)$ by threshold α. For simplicity of notation we omitted all other indexes x_0, y_0, d_i. Otherwise, RTNCC uses the average correlation TNCC over the entire temporal window (2.4). In this way, the RTNCC statistic achieves high discriminability.

2.2.2 Matching algorithm

To establish the matching between stereo images, the proposed RTNCC statistic is integrated in a seed growing procedure Čech *et al.* (2010) that is sketched in

pseudo-code in Alg. 1.

The input is the sequence of $2T+1$ image pairs, set of initial correspondences (the seeds) $\mathcal{S}$ which are obtained by matching Harris points Harris & Stephens (1988) between the images of the central frame t_0, and a parameter τ which directly controls a trade-off between accuracy and density of the matching. The output is disparity map $\mathbf{D}$ which relates pixel correspondences between the images of the central frame $\mathbf{I_l}^{t_0}, \mathbf{I_r}^{t_0}$.

Similarity statistic RTNCC(s), defined in (2.5), is computed for all seeds $s = (x, y, t_0, d) \in \mathcal{S}$, Step 1. For each seed, the algorithm searches other correspondences in the surroundings of the seeds by maximizing the similarity statistic. This is done in a 4-neighbourhood $\{\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3.\mathcal{N}_4\}$ of the pixel correspondence, such that in each respective direction (left, right, up, down) the algorithm searches the disparity in a range ± 1 pixel from the disparity of the seed, Step 5. If the similarity statistic of a candidate exceeds threshold τ, then a new correspondence is found, Step 7. It becomes a new seed, and output disparity map $\mathbf{D}$ is updated. The process repeats until there are no more seeds to be grown. For more details on the growing algorithm, we refer a reader to Čech *et al.* (2010).

Besides low computational complexity, and generally good results, the advantage of the algorithm in our context is the ability to accept the seeds as an input. Namely, we observed the condition in (2.5) of RTNCC is reliable in textured regions only. The decision does not work well for weakly textured areas or in the presence of strong noise. Nevertheless, the seed correspondences are points with the Harris property and for them the decision works well. Therefore, we propose to take this decision for the seeds only. Each seed then propagates a flag indicating whether the aggregation in RTNCC is used or not and this flag is inherited by its 'offspring' seeds in the growing process. This integration of the RTNCC statistic into the seed growing algorithm produces high quality results, which we show in the experiments.

Algorithm 1 Robust Spatiotemporal Matching

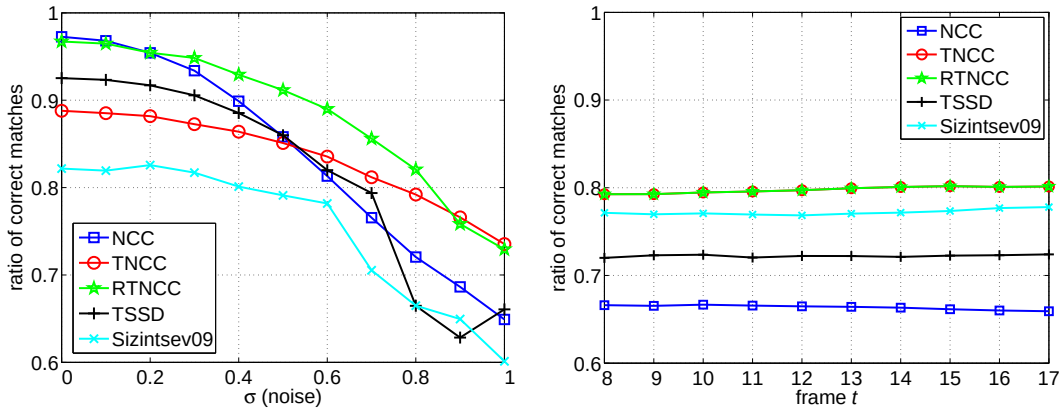
Require: Rectified images $(\mathbf{I_l}^{t_0-T}, \mathbf{I_r}^{t_0-T}), \ldots, (\mathbf{I_l}^{t_0}, \mathbf{I_r}^{t_0})$, $\ldots, (\mathbf{I_l}^{t_0+T}, \mathbf{I_r}^{t_0+T})$, initial correspondence seeds $\mathcal{S}$, image similarity threshold τ.

Compute RTNCC(s) statistic for every seed $s \in \mathcal{S}$.
Initialize empty matching disparity map $\mathbf{D}$ of size $\mathbf{I_l}^{t_0}$.
repeat
  Draw seed $s \in \mathcal{S}$ of the best RTNCC(s) value.
  for each of the four best neighbours
  $q_i^* = (x, y, t_0, d) = \underset{q \in \mathcal{N}_i(s)}{\operatorname{argmax}}\ \mathrm{RTNCC}(q),\ i \in \{1, 2, 3, 4\}$ **do**
    $c := \mathrm{RTNCC}(q_i^*)$
    if $c \geq \tau$ **and** pixels not matched yet **then**
      Update the seed queue $\mathcal{S} := \mathcal{S} \cup \{q_i^*\}$.
      Update the output map $\mathbf{D}(x, y) = d$.
    end if
  end for
until $\mathcal{S}$ is empty
return disparity map $\mathbf{D}$ for frame t_0.

2.3 Experiments

We performed a set of experiments to demonstrate that the proposed algorithm can cope with weak or ambiguous data and images corrupted by noise, without introducing artifacts of smoothing boundaries of rapidly moving objects. We compare the proposed algorithm (RTNCC) with two baseline instances of the growing algorithm: (i) the algorithm which uses the spatial neighbourhood only for matching (NCC), and (ii) the algorithm which trivially uses the spatio-temporal neighbourhood, such that all per-frame correlations are averaged (TNCC).

The other comparisons are with two state-of-the art spatio-temporal methods: (i) temporal SSD optimization Zhang *et al.* (2003) integrated in the growing algorithm (TSSD), and (ii) the stequel matching algorithm Sizintsev & Wildes (2009) (Sizintsev09).



(a) Synthetic Scene (b) York University Scene

Figure 2.3: Quantitative evaluation. Ratio of correct matches for: (a) synthetic data perturbed by noise with increasing standard deviation σ, (b) real data per frame of the sequence. In (a), note that RTNCC statistic performs the best for all noise levels. The NCC (non-temporal) statistic has the same performance without noise, then it drops quickly. The other tested statistics have lower performance (even without noise), because of the artifacts in the dynamic part of the scene. In (b), note that the due to slow motion, there are only small changes in disparity, therefore the performance of TNCC and RTNCC statistic is equal.

For all experiments, we used 5×5 pixel windows as the spatial neighbourhood of all statistics, parameter α in RTNCC (2.5) was empirically set to 0.8. For the short synthetic sequence, we set temporal window half-size to $T = 2$, while for all real data sequences it was set to $T = 7$.

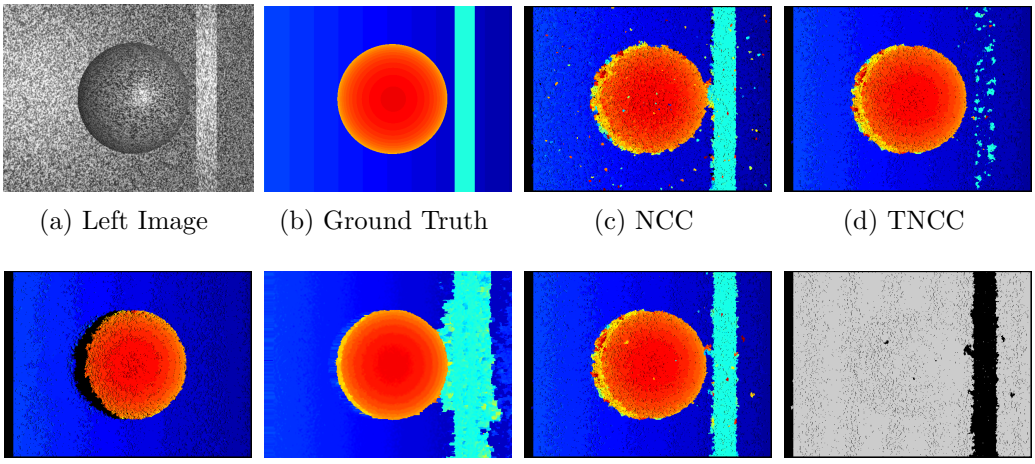
All disparity maps we show in this paper are colour coded, see e.g. Figure 2.5. Warmer colours correspond to higher disparities (closer to the camera), colder colours to lower disparities (further away from the camera). Black colour denotes unmatched pixels.

2.3.1 Ground-truth experiments

To quantitatively evaluate and compare the different algorithms, we tested on two stereo sequences with ground-truth disparity maps associated to each frame. The first sequence is a synthetic scene also used in Chapter B. It consists of three objects: a plane, a sphere, and a thin bar. The slanted background plane moves slowly towards the cameras. The sphere slowly rotates and slightly moves to the right and away from the cameras. Finally, the thin vertical bar moves rapidly (about 30 pixels per frame) from right to left crossing the entire scene. It is textured randomly with a white noise. How this ground truth was generated is explained in Appendix B. The other sequence used in Sizintsev & Wildes (2009) is a laboratory scene captured by real cameras, see Figure 2.5a. The scene is composed with multiple objects. Cameras very slowly move towards the scene, while part of the scene undergoes a small translation motion. The challenge here is that several objects have a weak texture, or a texture where the pattern is aligned with epipolar lines.

For all the experiments, we measured ratio of correctly matched pixels in non-occluded pixels, i.e. number of all pixels without mismatches (error ≥ 1 pixel) and unmatched pixels divided by the total number of pixels. This error statistic allows us to compare algorithms which differ in the density of their results. However, since the mismatches count the same as gaps, the algorithms are set to give maximum density.

In Figure 2.3, we plot the ratio of correct matches for experiments on two datasets. For the synthetic scene, we perturbed the input images with zero mean additive Gaussian noise with successively increasing standard deviation σ. The noise has equal variance as the signal for $\sigma = 1$. In Figure 2.3a, we can see the algorithm using NCC performs very well without noise. This is because the texture is optimal and hence the correlation is very high and unambiguous. However, it degrades rapidly with noise, see Figure 2.4c, producing mismatches and becoming more unstable as small spatial only image windows do not correlate well. The TNCC degradates slowly with increasing level of noise, however the ratio of correct matches is lower since it tends to completely miss the rapidly



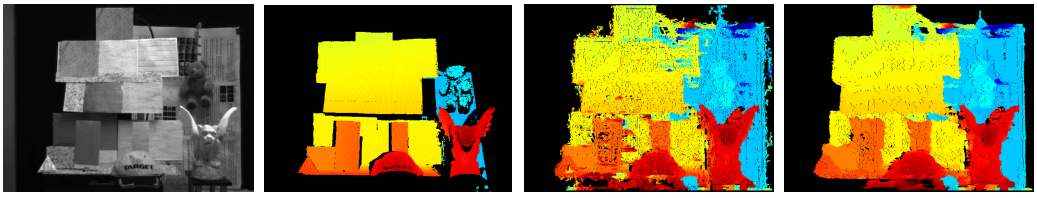
(a) Left Image (b) Ground Truth (c) NCC (d) TNCC

(e) TSSD (f) Sizintsev09 (g) **RTNCC** (h) RTNCC aggreg.

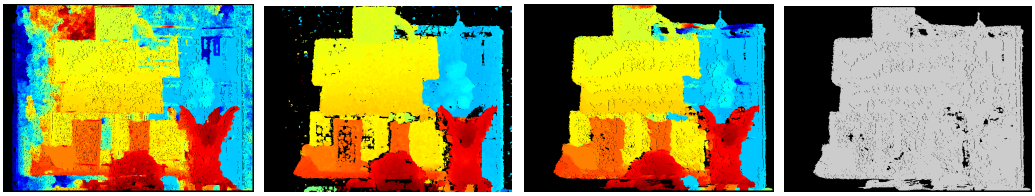
Figure 2.4: Synthetic dataset. Frame 6, noise level $\sigma = 0.5$. Disparity maps of different methods. The proposed method (g) is significantly less affected by noise than per-frame method (c), and there are no serious artifacts around rapidly moving bar, as illusory disappearing in (d), (e), or blurring (f). Notice the region of rapidly moving bar correspond well with the map, where the temporal aggregation was automatically switched off, in black (h).

moving bar, see Figure 2.4d. The temporal aggregation helps to filter out the noise in slowly moving regions, but the aggregation is harmful for the bar where the disparity changes abruptly over time, since for this region TNCC of the false background wins over TNCC of the true bar. Similarly, the other two methods Sizintsev09 and TSSD perform well filtering the noise but both of them have serious problems with the rapidly moving bar where the disparity changes abruptly over time, see Figure 2.4e, 2.4f. Our proposed method RTNCC performs the best. It is as good as the NCC for low noise, and it is always superior to other methods with increasing levels of noise. It has fewer mismatches in slowly moving regions, but at the same time it preserves the rapidly moving bar without artifacts, see Figure 2.4g. The reason is that it correctly aggregates over time using TNCC in regions where it helps, and for other regions it uses the spatial statistic NCC. The map in Figure 2.4h shows which case in (2.5) was used in results of RTNCC. Pixels matched using the temporal aggregation are indicated by gray colour, pixels matched by spatial statistic by black colour. We can see, it correctly used the spatial statistic NCC for the region of the rapidly moving bar, while for other pixels, it correctly used the temporal aggregation TNCC.

For the real scene, we did not perturb the input data, we show the ratio of correct matches per frame, Figure 2.3b. We can see the low performance of NCC. It is caused by many mismatches, since the texture is weak and ambiguous, see Figure 2.5c. Methods TNCC and RTNCC perform the best and exactly the same for this scene. Their plots coincide. Mismatches are nicely filtered out, see Figure 2.5d, 2.5g. The reason is the small motion does not change the disparity much and even a trivial temporal aggregation significantly helps improving the results over the spatial statistic NCC. Notice, that the RTNCC statistic used full temporal aggregation for practically all pixels, see Figure 2.5h, which makes it equivalent to TNCC in this case. Results of Sizintsev09, Figure 2.5f, are comparable or slightly inferior to ours, since the motion is very small and these data of authors of the algorithm are probably optimal. Method TSSD does not perform so well, probably due to possible overfitting and loss of discriminability in this kind of texture, as discussed before.



(a) Left Image (b) Ground Truth (c) NCC (d) TNCC



(e) TSSD (f) Sizintsev09 (g) **RTNCC** (h) RTNCC aggreg.

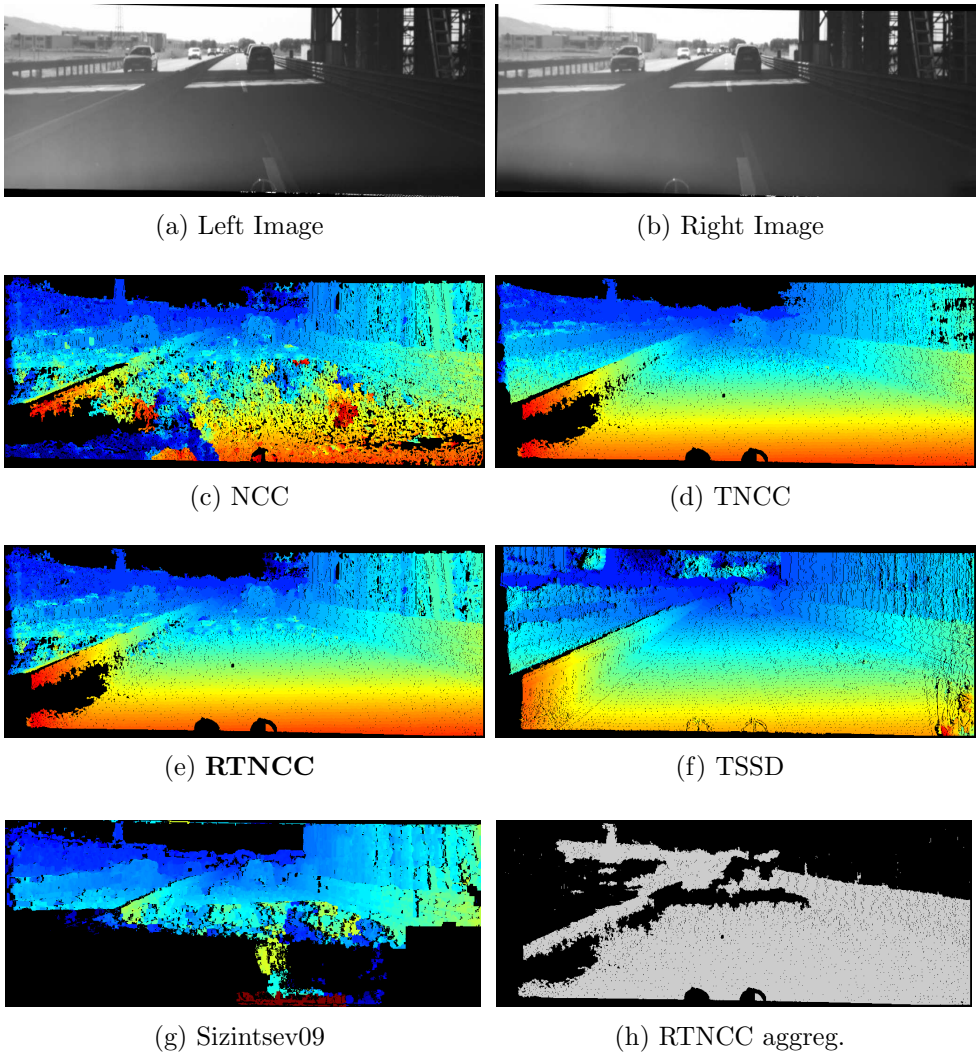
Figure 2.5: York University dataset of Sizintsev & Wildes (2009), frame 8. Disparity maps are fairly similar since the scene moves really slowly and the disparity does not change much over time. Almost the entire scene is matched by the temporally aggregated statistic, see (h).

2.3.2 Real outdoor scenes

To show the validity of the proposed algorithm on real outdoor scenes we tested under two different stereo datasets. The DAGM Challenge Exposure Changes dataset[1] (DAGM), and the ETHZ dataset[2] (ETHZ). The DAGM dataset is recorded by a stereo camera mounted in a car driving in a highway quite rapidly in difficult lighting conditions, sudden changes in the exposure and sharp shadows. Cars going in the opposite lane moves very fast, see Figure 2.6a. The ETHZ dataset was recorded by a stereo camera mounted on a pram and strolled in the street. It is a complex scene with multiple pedestrians moving typically forward the camera, see Figure 2.7a.

For DAGM dataset, we show results for the frame, where the car is passing under the bridge, where the lighting conditions are very bad. The texture of the road almost disappears. It causes the spatial statistic NCC to fail, producing

[1]http://www.dagm2011.org/adverse-vision-conditions-challenge.html
[2]http://www.vision.ee.ethz.ch/ aess/dataset/



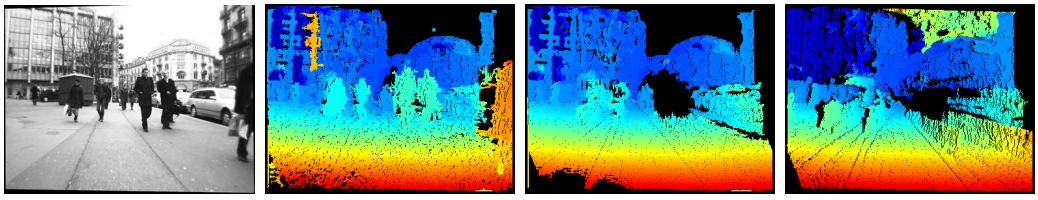
(a) Left Image (b) Right Image

(c) NCC (d) TNCC

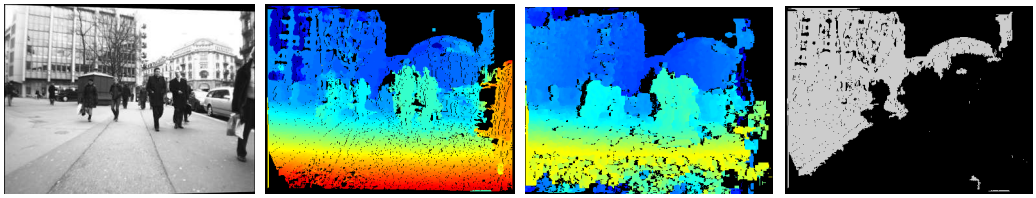
(e) **RTNCC** (f) TSSD

(g) Sizintsev09 (h) RTNCC aggreg.

Figure 2.6: DAGM Exposure Challenge dataset. Disparity maps.



(a) Left Image (b) NCC (c) TNCC (d) TSSD



(e) Right Image (f) **RTNCC** (g) Sizintsev09 (h) RTNCC aggreg.

Figure 2.7: ETHZ dataset. Disparity maps.

many mismatches, as in Figure 2.6c. The spatio-temporal version TNCC works better, see Figure 2.6d. Much information is retained due to the temporal aggregation. Notice that the disparity of the road remains constant over time and this is also the case of the car going in the same direction, since the distance to it is more or less constant. However, the problem is, that the car going in the opposite direction, whose relative velocity is very high, is missed by the TNCC. This is the same effect as the case of the rapidly moving bar in Figure 2.4d. TSSD has similar difficulties there, Figure 2.6f. Surprisingly, algorithm Sizintsev09 has severe problems with all rapidly moving pixels in the scene, including those where the disparity remains constant. It produces large artifacts in regions near the camera. The proposed RTNCC works well, Figure 2.6e. It is significantly superior to NCC and all objects, including the car in the opposite direction, are preserved.

For the ETHZ dataset, we can observe similar behaviour of the methods. Spatial NCC is already quite good, but there is a clear mismatch in the repetitive structure of the building, Figure 2.7b. Temporal aggregation in TNCC removes this artifact, however it misses three pedestrians who walk towards the camera, Figure 2.7c. Their disparity in the location of the middle frame changes abruptly, which causes the same effect as above. Method TSSD suffers from similar arti-

facts, Figure 2.7d. Algorithm Sizintsev09 similarly as above produces mismatches in the region of the fast motion near the cameras, besides missing the closest pedestrian, see Figure 2.7g. The proposed RTNCC works well, Figure 2.6e. Due to temporal aggregation, it is able to remove the mismatch of spatial NCC while preserving all the pedestrians.

We can also observe small imperfections in our disparity maps in Figure 2.6e, 2.7f. Small mismatches are probably caused by insufficient temporal aggregation. Looking at the map indicating the decision on the aggregation of RTNCC in Figure 2.6h, 2.7h, we can see, it aggregates correctly in the region of temporally constant disparity, but these regions are not complete, e.g. the road or the pavement are not aggregated completely. This is due to a conservative choice of α in (2.5) in order to perform at least as good as the spatial statistic without introducing artifacts by incorrect temporal aggregation.

Algorithm Complexity and Implementation Notes. The proposed algorithm inherits the low computational complexity from the growing procedure in Čech *et al.* (2010). The complexity is given by the size of the disparity search space, *i.e.* by the number of correlation statistics which have to be computed. Assuming the images of size n^2, any algorithm which searches the disparity space exhaustively is of complexity $\mathcal{O}(n^3)$ However, the growing algorithm, due to the limited local search in the vicinity of the seeds, reduces the complexity to $\mathcal{O}(n^2)$, Čech *et al.* (2011).

Practically, we used a non-optimized (combined Matlab and C) implementation of the growing algorithm using NCC which takes about 1 second per frame of 640×480 pixel images. There also exists an implementation of this algorithm running in real time on standard CPU Dobiaš & Šára (2011). Using a primitive implementation, the cost for temporal aggregation (RTNCC, TNCC) scales the CPU time with a factor of the temporal window size. However, this can be highly reduced by reusing correlations computed for previous frames such that this extra cost becomes negligible. The TSSD statistic takes about 10 seconds (due to gradient descent optimization), and Sizintsev09 about 30 seconds per frame.

2.4 Discussion

We presented a spatiotemporal correlation statistic that increases the discriminability by aggregating over time and hereby produces higher quality matching results. The proposed method is robust to a rapid motion in the scene, which is a situation where the state-of-the-art algorithms are prone to produce artifacts. We performed experiments demonstrating the validity of the method on two scenes with the ground-truth (synthetic and real datasets), and on two real outdoor challenging datasets.

We obtained promising results already, despite the simplicity of the method, namely the heuristic decision rule on the aggregation of RTNCC. We demonstrate that we are able to deal with extremely challenging situations in dynamic outdoor scenes where stereo algorithms have more difficulties and alternative devices like time-of-flight (TOF) cameras are unable to operate.

Chapter 3

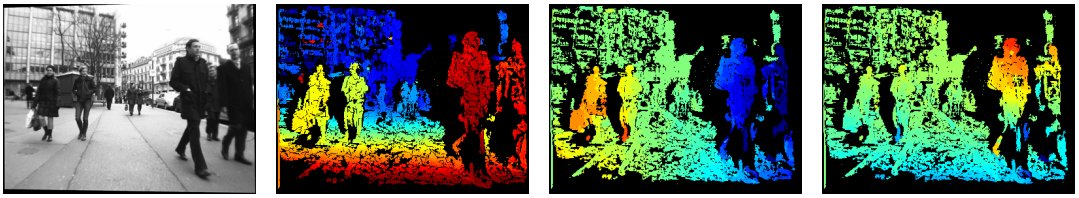
Scene Flow

3.1 Introduction

A sequence of image pairs gathered with calibrated and synchronized cameras contains more information to estimate depth and 3D motion than a single stereo-pair or a single image sequence. There are approaches Richardt *et al.* (2010); Sizintsev & Wildes (2009); Zhang *et al.* (2003) which exploit the extra temporal information to estimate disparity maps, but do not estimate the motion explicitly, we call them a *spatiotemporal stereo*.

Other methods estimate a complete scene flow benefiting from a coupled stereo and optical flow correspondence problem. *Scene flow* was introduced in Vedula *et al.* (1999) as a dense 3D motion field. It can be estimated with: (1) variational methods Basha *et al.* (2010); Huguet & Devernay (2007); Pons *et al.* (2003), which are usually well suited for simple scenes with a dominant surface; (2) discrete MRF formulations Isard & MacCormick (2006); Liu & Philomin (2009), which involve expensive discrete optimization, and (3) local methods finding the correspondences greedily, which are efficient Gong (2009) but not so accurate.

We propose a seed growing algorithm to estimate the scene flow in a binocular-video setup. A basic principle of the seed growing methods is that correspondences are found in a small neighborhood around an initial set of seed correspondences. This idea has been adopted in stereo Čech & Šára (2007); Čech *et al.*



(a) (b) disparity (c) horizontal OF (d) vertical OF

Figure 3.1: Output of the proposed algorithm on ETH dataset as color coded maps. For disparity, warmer colors are closer to the camera. In optical flow (OF), green color is zero motion, warmer colors is left and up motion, colder colors is right and down motion respectively. Black color denotes unmatched pixels.

(2010); Kannala & Brandt (2007); Lhuillier & Quan (2002), but to the best of our knowledge, it has not been used for scene flow. The advantage of such approaches is a fast performance compared to global variational and MRF methods, and a good accuracy compared to purely local methods, since neighboring pixel relations are not ignored completely.

Our proposed algorithm can simultaneously estimate accurate temporally-coherent disparity and optical flow maps of a scene with a rich 3D structure and large motion between time instances. Small local variations of disparity and flows are captured by the growing process while large displacement are found due to the seeds. Boundaries between objects and different motions are naturally well preserved without smoothing artifacts. Nevertheless, the algorithm produces semi-dense (unambiguous) results only, but they are dense enough for many potential applications, see Figure 3.1.

3.2 Algorithm Description

The proposed algorithm for growing correspondences of scene flow in a sequence of stereo images (GCSFs) is summarized in Figure 3.2. At each time instance t, it takes as input two epipolarly rectified image pairs, a pair $\mathbf{I}_l^0, \mathbf{I}_r^0$ for time $t-1$ (last frame), and the consecutive pair $\mathbf{I}_l^1, \mathbf{I}_r^1$ for time t (current frame). The output at

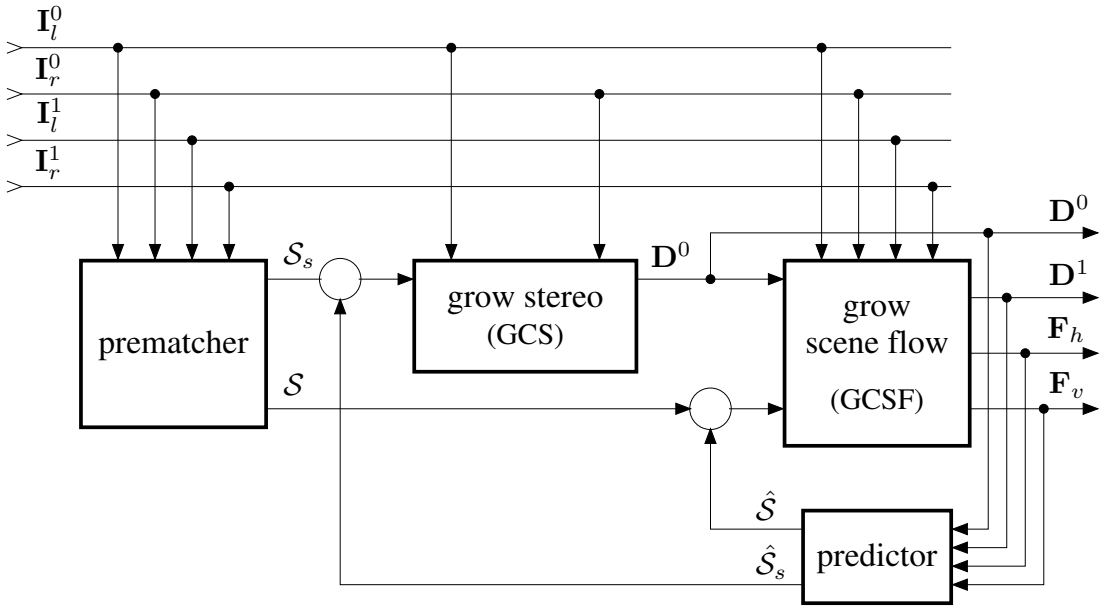


Figure 3.2: Overview of the proposed algorithm (GCSFs).

each time instance is a disparity map $\mathbf{D}^0$ holding the stereo correspondences from the last frame $t-1$, disparity map $\mathbf{D}^1$ holding correspondences found between $\mathbf{I}_l^1$ and $\mathbf{I}_r^1$, and horizontal and vertical optical flow maps $\mathbf{F}_h$ and $\mathbf{F}_v$ respectively, encoding the correspondences between consecutive images $\mathbf{I}_l^0$ and $\mathbf{I}_l^1$.

Notice that having full camera calibration, this representation fully determines the scene flow, since $\mathbf{D}^0$ gives a reconstruction of 3D points $\mathcal{X}^0$, $\mathbf{D}^1$ a reconstruction of 3D points $\mathcal{X}^1$ (after the motion), and $\mathbf{F}_h, \mathbf{F}_v$ gives the mapping between these two sets.

First, a prematcher is run to deliver initial correspondences, the seeds. They are used in subsequent growing processes. The prematcher finds sparse correspondences of interest points between left and right images and between consecutive images. Each seed $\mathbf{s} = (x_l^0, x_r^0, y^0, x_l^1, x_r^1, y^1) \in \mathcal{S}$ represents a correspondence of 4 pixels, *i.e.* projections of a 3D point $X^0 \in \mathcal{X}^0$ into $\mathbf{I}_l^0, \mathbf{I}_r^0$ and the same 3D point after the motion $X^1 \in \mathcal{X}^1$ into $\mathbf{I}_l^1, \mathbf{I}_r^1$. The seed encapsulates both stereo and optical flow correspondences, see Figure 3.3. Beside the set of these scene flow seeds, the prematcher also output the stereo seeds $\mathbf{s}_s = (x_l^0, x_r^0, y^0) \in \mathcal{S}_s$ which is a set of two-pixel correspondences between $\mathbf{I}_l^0$ and $\mathbf{I}_r^0$.

Then, the stereo seeds $\mathcal{S}_s$ are grown by a stereo algorithm (GCS), which computes a disparity map $\mathbf{D}^0$ between $\mathbf{I}_l^0$ and $\mathbf{I}_r^0$. Disparity map $\mathbf{D}^0$ together with seeds $\mathcal{S}$ and the input images are an input of the subsequent algorithm (GCSF), which jointly grows disparity map $\mathbf{D}^1$, and the optical flow maps $\mathbf{F}_h$, $\mathbf{F}_v$.

The solution at time t contains lots of information about the solution at time $t+1$, *i.e.* when a new frame is available. This information, is exploited in the proposed algorithm by predicting the seeds for the growing processes in the next time instance. Considering the motion of pixels from previous solution, the predictor estimates new correspondence seeds $\hat{\mathcal{S}}$ and $\hat{\mathcal{S}}_s$. These seeds are unified with current seeds given by the prematcher. It means, that starting from the second frame, the growing processes work with larger and richer set of seeds. The prematcher remains connected for all frames in order to capture the dynamic scene events in which objects suddenly appears. This process is repeated with each subsequent frame.

Details of the algorithm are described below. First, we describe in detail the procedure for growing the scene flow, since it is the essential part. Afterward, we give further details on the rest of the algorithm.

3.2.1 Growing scene flow (GCSF)

The algorithm is presented in pseudo-code as Algorithm 2. It takes as input two rectified image pairs $\mathbf{I}_l^0, \mathbf{I}_r^0$ and the consecutive pair $\mathbf{I}_l^1, \mathbf{I}_r^1$, a set of initial correspondence seeds $\mathcal{S}$, a disparity map $\mathbf{D}^0$ for a previous frame $t-1$, and the parameters α (temporal consistency enforcement), β (optical flow regularization), and τ (growing threshold). The output are maps of disparity $\mathbf{D}^1$ and optical flows $\mathbf{F}_h$, $\mathbf{F}_v$.

First, the algorithm computes a photometric consistency statistic of the 4-pixel correspondence by average correlation

$$\mathrm{corr}(\mathbf{s}) = \frac{\mathrm{c}_{lr}^{11}(x_l^1, y_l^1, x_r^1, y_l^1) + \mathrm{c}_{ll}^{01}(x_l^0, y_l^0, x_l^1, y_l^1) + \mathrm{c}_{rr}^{01}(x_r^0, y_r^0, x_r^1, y_r^1)}{3}. \tag{3.1}$$

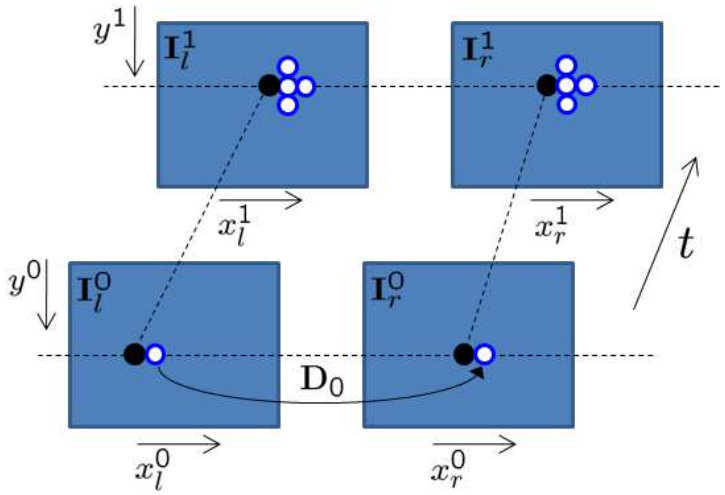


Figure 3.3: A sequence of consecutive epipolarly rectified stereo images. A seed correspondence $\mathbf{s}$ sketched by filled circles, its right neighborhood $\mathcal{N}_1$ by empty circles.

Left-right correlation c_{lr}^{11} is between small windows centered at pixels $\mathbf{I}_l^1(x_l^1, y_l^1)$ and $\mathbf{I}_r^1(x_l^1, y_l^1)$. Similarly the correlations c_{ll}^{01} and c_{rr}^{01} are between consecutive images in the left and right sequences. All the correlations are MNCC statistics Moravec (1977) on 5×5 pixel widows. Seed correlation $\mathbf{s}.c$ is enhanced by a small positive α to enforce temporal consistency, Step 1. The set $\mathcal{S}$ is organized as a correlation priority queue. The seed $\mathbf{s} \in \mathcal{S}$ is removed from the top of the queue, Step 3. If its consistency exceeds threshold τ in Step. 4, output maps are updated by

$$\begin{aligned} \mathbf{D}^1(x_l^1, y^1) &= x_l^1 - x_r^1, \\ \mathbf{F}_h(x_l^1, y^1) &= x_l^1 - x_l^0, \qquad & \mathbf{F}_v(x_l^1, y^1) = y^1 - y^0. \end{aligned} \tag{3.2}$$

For all four neighbors (right, left, up, down) of seed $\mathbf{s}$, the best correlating candidate in $\mathcal{N}_i(\mathbf{s}|\mathbf{D}^0)$ is found, Step 5. For instance

$$\begin{aligned} \mathcal{N}_1(\mathbf{s}|\mathbf{D}^0) = \Big\{ \bigcup_{\mathbf{k}\in\mathcal{L}} & (x_l^0 + 1, x_l^0 + 1 - \mathbf{D}^0(x_l^0 + 1, y^0), y^0, \\ & x_l^1 + 1, x_r^1 + 1, y^1) + (0, 0, 0, \mathbf{k}) \Big\}, \end{aligned} \tag{3.3}$$

where $\mathcal{L} = \{(0,0,0), (\pm 1,0,0), (0,\pm 1,0), (0,0,\pm 1)\}$ is a set of seven local search vectors having the stereo or temporal disparity less or equal to one, see Figure 3.3.

Algorithm 2 Growing the scene flow (GCSF)

Require: rectified images $\mathbf{I}_l^0$, $\mathbf{I}_r^0$, $\mathbf{I}_l^1$, $\mathbf{I}_r^1$,
initial correspondence seeds $\mathcal{S}$,
disparity map $\mathbf{D}^0$,
parameters α, β, τ.

1: Compute similarity $\mathbf{s}.c = \mathrm{corr}(\mathbf{s}) + \alpha$ for all seeds $\mathbf{s} \in \mathcal{S}$.
2: **repeat**
3: Draw the seed $\mathbf{s} \in \mathcal{S}$ of the best similarity $\mathbf{s}.c$.
4: **if** $\mathbf{s}.c \geq \tau$ **then** Update output maps. **endif**
5: **for** each of the four best neighbors $i \in \{1, 2, 3, 4\}$
$$\mathbf{t}_i^* = (x_l^0, x_r^0, y^0, x_l^1, x_r^1, y^1) = \operatorname*{argmax}_{\mathbf{t} \in \mathcal{N}_i(\mathbf{s}|\mathbf{D}^0)} \mathrm{corr}_{\mathbf{s}}^{\beta}(\mathbf{t}),$$
do
6: $\mathbf{t}_i.c = \mathrm{corr}_{\mathbf{s}}^{\beta}(\mathbf{t}_i^*)$,
7: **if** $\mathbf{t}_i.c \geq \tau$ **and** all pixels in $\mathbf{t}$ not matched yet **then**
8: Update output maps.
9: Update the seed queue $\mathcal{S} = \mathcal{S} \cup \{\mathbf{t}_i^*\}$.
10: **end if**
11: **end for**
12: **until** $\mathcal{S}$ is empty.
13: **return** disparity map $\mathbf{D}^1$, flow maps $\mathbf{F}_h$, $\mathbf{F}_v$.

Notice the candidates depend on the previous disparity $\mathbf{D}^0$. The other neighbors $\mathcal{N}_2, \mathcal{N}_3, \mathcal{N}_4$ are defined similarly.

The optical flow generally suffers from a well known aperture problem. This is not completely avoided in a joint stereo setup. Therefore we regularize assuming the seed has a correct flow, new candidates having a different flow are penalized by lower correlation

$$\mathrm{corr}(\mathbf{t})_{\mathbf{s}}^{\beta} = \mathrm{corr}(\mathbf{t}) - \beta ||\mathbf{s}.f - \mathbf{t}.f||_1, \tag{3.4}$$

where notation $.f = (x_l^1 - x_l^0, x_r^1 - x_r^0, y^1 - y^0)$ means a vector of optical flows of respective seeds $\mathbf{s}$ and $\mathbf{t}$, where β is a small positive constant.

If the highest correlation exceeds a threshold τ and any of the pixels in $\mathbf{t}$ is unmatched so far, then a new match is found, Step 7. Output maps are updated

by (3.2) in Step 8, and the found match becomes a new seed, Step 9. Up to four seeds are created in each growing step. The process continues until there are no seeds in the queue, Step 12.

Default values of algorithm parameters were found empirically and set to $\alpha = \beta = 0.05$, $\tau = 0.6$ in all our real-data experiments. The value of temporal consistency parameter α in Step 1 is a trade-off between a temporal coherence of the results and an ability to capture fast changes in the motion. We observed that for $\alpha = 0$, the results are not so temporally coherent, certain matches in the 3D surface were randomly disappearing and reappearing due to noise or various degradations in the image sequence. Small $\alpha > 0$ causes that already matched points have a better position in the priority queue and higher chance to be matched. On the other hand, when α is too high, we observed matching errors in sudden changes of object's motion, since wrong (incorrectly predicted) seeds were accepted in Step 4.

Parameter β in (3.4) regularizes the growing process to handle the aperture problem. When $\beta = 0$, we observed artifacts of the optical flow estimation in edge-like structures. Growing process finds the matches based on local maxima of correlation, which need not necessarily correspond to the correct solution due to various noise in the images. Very small $\beta > 0$ helps. However, when β is too large, the solution is biased towards seeds and locally flat around them.

The last parameter τ directly controls the trade-off between the density of the solution and mismatch rate.

Note that MNCC statistic in (3.1) is not invariant to deformation of local image neighborhoods between corresponding pixels related by optical flow, which occurs due to camera or scene motion. A general assumption, which is hardly preserved, is a fronto-parallel surface undergoing a fronto-parallel motion Zhang *et al.* (2003). Nevertheless the statistic is insensitive enough to violations of this assumption. We show in the experiments that the algorithm works well under non-trivial motion and non-planar or slanted surfaces. In cases where this could be a problem, a simple extension would be to associate a set of parameters

capturing the local affine transformations with the seed, as in Čech *et al.* (2010); Kannala & Brandt (2007) in the context of wide-baseline stereo matching.

3.2.2 Growing stereo (GCS)

A seed growing algorithm Čech & Šára (2007) for stereo matching between images $\mathbf{I}_l^0$ and $\mathbf{I}_r^0$ is used. The growing procedure is similar in spirit to Alg. 2, however the neighborhoods $\mathcal{N}_i$ are different. This algorithm is reported being not very sensitive to wrong seeds, which is achieved by a robust matching which selects the final solution among competing correspondence hypotheses from the growing process. In the experiments, we compare this algorithm when run frame-by-frame with the same algorithm integrated in the proposed pipeline shown in Figure 3.2.

3.2.3 Prematcher

The task of the prematcher is to deliver sparse correspondences of interest points. This is achieved in our implementation by matching Harris points and tracking them using multi-level version of LK tracker Lucas & Kanade (1981). The stereo seeds $\mathcal{S}_s$ are simply those Harris points which satisfy the epipolar constraint, and whose 5×5 MNCC correlation exceeds threshold τ. The scene flow seeds $\mathcal{S}$ are obtained by tracking the stereo seeds from $\mathbf{I}_l^0$ to $\mathbf{I}_l^1$ and from $\mathbf{I}_r^0$ to $\mathbf{I}_r^1$. The point matches which violates the epipolar constraint between $\mathbf{I}_l^1$ and $\mathbf{I}_r^1$ are discarded from the set.

The algorithm is not limited to Harris seeds. Any other seeds, *e.g.* from wide-baseline matching of distinguished regions, or other more sophisticated tracking techniques, could be used.

3.2.4 Predictor

The predictor estimates seeds for processing of the next frame based on the current solution and other assumptions on the motion of points. In our implementation, we use a simple assumption, that the point moves constantly in the

image plane, *i.e.* its optical flow remains the same in a subsequent frame. For each matched pixel (x_l^1, y^1) in $\mathbf{D}^1$, the predicted seed $\hat{\mathbf{s}} = (\hat{x}_l^0, \hat{x}_r^0, \hat{y}^0, \hat{x}_l^1, \hat{x}_r^1, \hat{y}^1)$ is

$$\begin{aligned} \hat{x}_l^0 &= x_l^1, & \hat{x}_l^1 &= x_l^1 + \mathbf{F}_h(x_l^1, y^1), \\ \hat{x}_r^0 &= x_l^1 - \mathbf{D}^1(x_l^1, y^1), & \hat{x}_r^1 &= \hat{x}_r^0 + (\hat{x}_r^0 - x_r^0), \\ \hat{y}^0 &= y^1, & \hat{y}^1 &= y^1 + \mathbf{F}_v(x_l^1, y^1), \end{aligned} \tag{3.5}$$

where $x_r^0 = x_l^0 - \mathbf{D}^0(x_l^0, y^0)$ and $x_l^0 = x_l^1 - \mathbf{F}_h(x_l^1, y^1)$, $y^0 = y^1 - \mathbf{F}_h(x_l^1, y^1)$. It follows from the output maps in (3.2). Notice that for stereo seed $\hat{\mathbf{s}}_s = (\hat{x}_l^0, \hat{x}_r^0, \hat{y}^0)$, the disparity map $\mathbf{D}^1$ is only 'translated' into the seed representation and subsequently grown again by stereo Čech & Šára (2007) to provide new disparity map $\mathbf{D}^0$. This is important since certain pixels may not be matched in $\mathbf{D}^1$ due to motion occlusions, and they are hereby recovered.

The constant motion assumption is rather naïve. More correct would be to use more sophisticated dynamic motion models and Kalman filtering. Nevertheless, despite the simplicity, the predictor usually helps producing enough correct seeds. When the assumption of the constant motion is violated, the affected seeds become wrong with low correlation and they are placed in an unfavorable position in the priority queue. Such regions are grown from other correct seeds (sparse Harris seeds from prematcher, or other seeds where the assumption holds).

3.2.5 Complexity of the algorithm

The algorithm has low complexity. Assuming $n \times n$ images, any algorithm searching the correspondences exhaustively has the complexity at least $\mathcal{O}(n^5)$ per frame Gong (2009), which is the size of the search space without limiting the ranges for disparity and horizontal and vertical flow. However, the proposed algorithm has the complexity $\mathcal{O}(n^2)$ per frame, since it searches the correspondences in a neighborhood of the seeds tracing discrete manifolds of a high correlation defined above the pixels of the reference image.

3.3 Experiments

The experiments demonstrate that the proposed algorithm produces accurate semi-dense results and that it benefits from a joint disparity – optical flow formulation in a sequence of stereo images. The proposed method is compared with a recent spatiotemporal stereo algorithm by Sizintsev & Wildes (2009), with a variational scene flow algorithm by Huguet & Devernay (2007), and with a recent optical flow by Brox & Malik (2010). The experiments show that our algorithm is more precise in disparity than Sizintsev & Wildes (2009) and Huguet & Devernay (2007), and in optical flow comparable to Huguet & Devernay (2007), and slightly inferior to Brox & Malik (2010).

3.3.1 Synthetic Data

To quantitatively evaluate and compare the methods, we carried out an experiment with simulated data. The synthetic scene consists of three moving objects: a sphere performing a complicated rotation while moving slowly to the right and away from the cameras, a small vertical bar moving very fast to the left (30 pixels/frame), and a slanted background plane moving towards the cameras. The scene was textured randomly with a white noise, see Figure 3.4. The scene was synthesized using Blender. The resulting sequence has 25 frames of stereo-pair images and each frame has associated ground-truth disparity, optical flow maps, and maps of stereo and motion occlusions. More details can be found in Appendix B.

The algorithms were tested under noise perturbation of data. An independent Gaussian noise was added into each image of the stereo sequence. The experiment was performed with several noise levels, starting from $\sigma = 0$ (no noise) up to $\sigma = 1$ where the variation of the noise is the same as of the image signal.

For all the experiments, we measured an average ratio of correctly matched pixels in non-occluded regions, *i.e.* number of all pixels without mismatches (error ≥ 1 pixel) and non-matches divided by total number of pixels, over all frames in the sequence. Notice, this evaluation is very strict for algorithms which do not

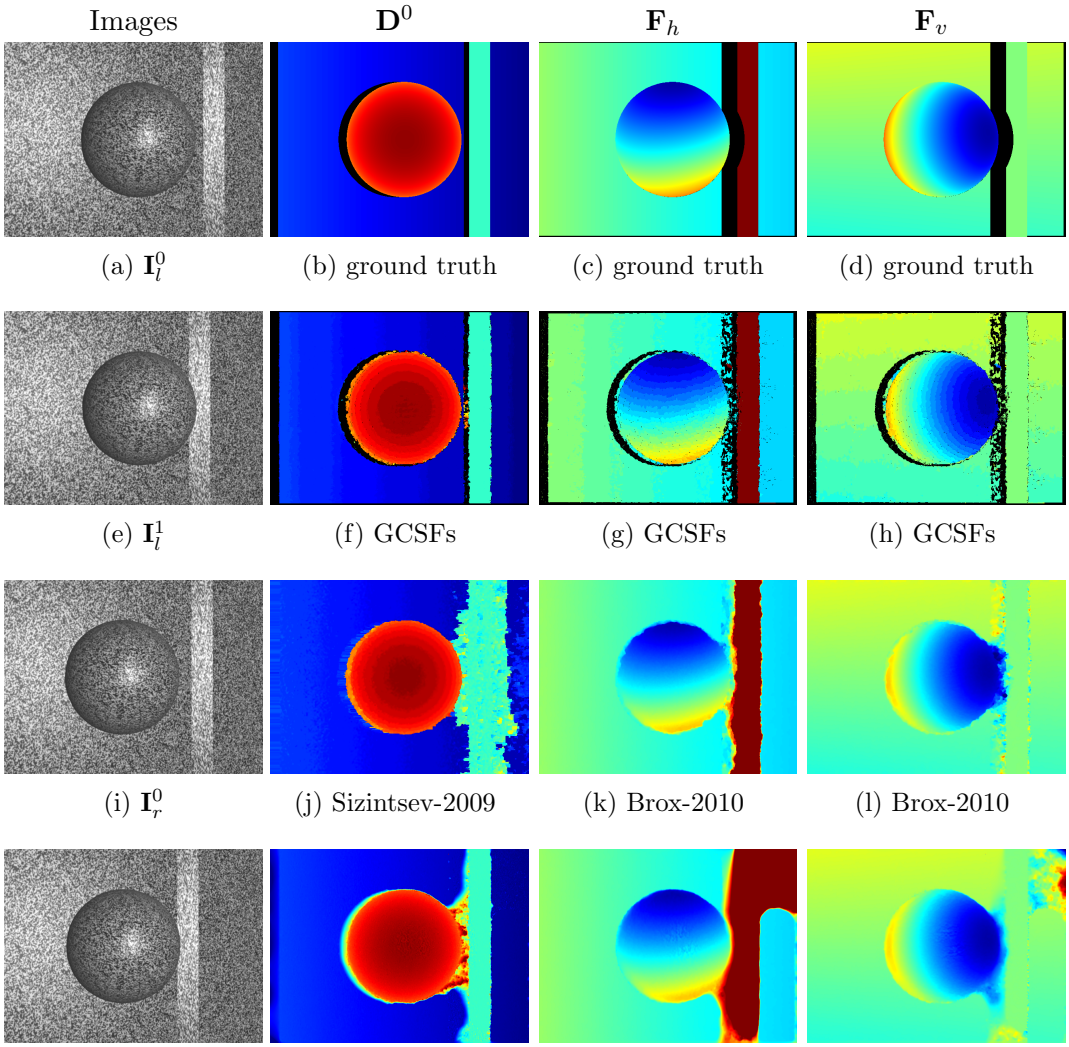


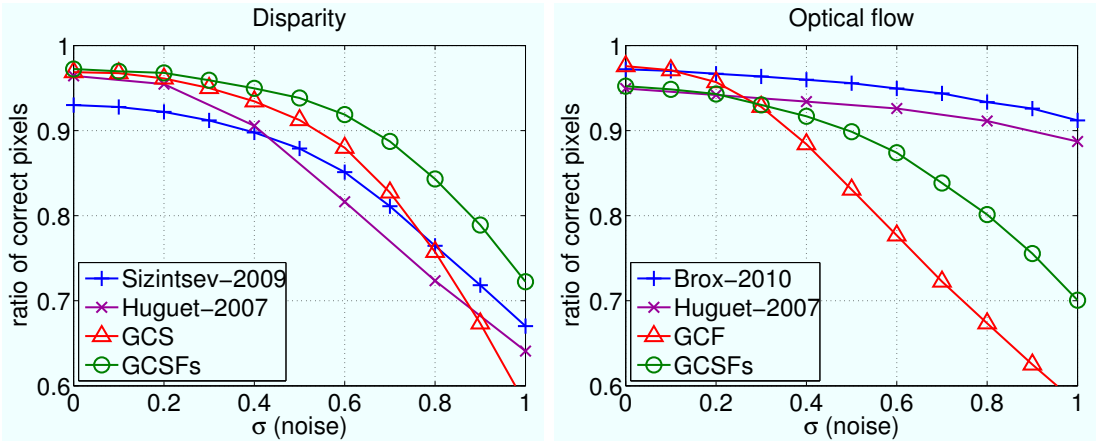
Figure 3.4: Synthetic experiment. Disparity and optical flow maps of the 6th frame of the sequence: Ground-truth maps with marked occlusions, results of tested algorithms.

give fully dense results, like ours. However this is an easy way to simply compare semi-dense and fully-dense results. On the other hand, since the mismatches are counted the same as unmatched pixels, we relax the correlation threshold $\tau = 0$ for all synthetic experiments, other parameters remained of the default values ($\alpha = \beta = 0.05$). This is the only exception in all the experiments in this paper.

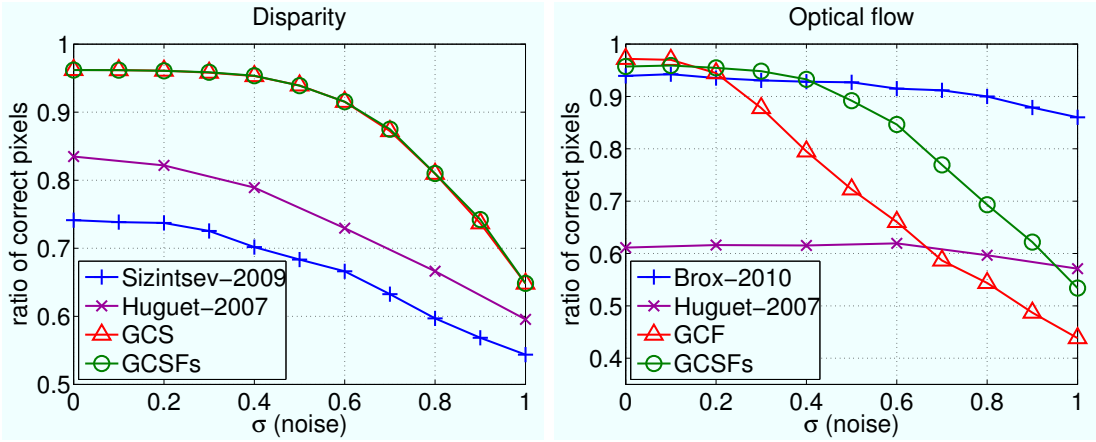
This statistic was measured for both disparity and optical flow errors. Optical flow is usually evaluated by average angular error, however the proposed algorithm is of the pixel level accuracy and therefore this usual evaluation would not be suitable. We understand the optical flow as pixel matching problem, similar to stereo without epipolar constraint. It is important to capture gross errors of the optical flow estimates, *i.e.* mismatches by more than 1 pixel error. This evaluation is again fair for classical sub-pixel optical flow methods, since the ground-truth is provided with a sub-pixel precision.

Results of the experiment are shown in Figure 3.5a. In case of stereo, we compared the proposed algorithm (GCSFs) which jointly estimates disparity and optical flow with: a seed growing algorithm which computes disparity maps frame-by-frame independently Čech & Šára (2007) (GCS), scene flow algorithm Huguet & Devernay (2007) (Huguet-2007), and the spatiotemporal stereo Sizintsev & Wildes (2009) (Sizintsev-2009). We can see, there is not much difference for GCSFs and GCS for low level of noise, however the GCSFs is more stable for higher level of noise. Algorithm Sizintsev & Wildes (2009), while performing well in slow moving regions, has severe difficulties with the quickly moving bar even without noise, see Figure 3.4j, which causes its inferior performance compared to the proposed method. Algorithm Huguet & Devernay (2007) has also severe difficulties with this scene. Corresponding disparity map of GCSFs is shown in Figure 3.4f. We can see no significant mismatches in either part of the scene, object boundaries are well preserved except for small phenomena due to fluctuations of the window similarity statistic. There are also small mismatches in occluded regions, since the threshold τ is relaxed, but they are not included in the evaluation.

In case of optical flow, we compared the flow provided by proposed GCSFs algorithm with another seed growing algorithm which frame-by-frame indepen-



(a) The error statistics evaluated over the entire scene



(b) The error statistics evaluated only in the area of the thin vertical bar.

Figure 3.5: Algorithm accuracy under contamination with a Gaussian Noise. The signal has equal variance as the noise for $\sigma = 1$.

dently searches the stereo-correspondences without epipolar constraint (GCF). This growing mechanism was used in Čech *et al.* (2010). Additionally we compare this with a recent variational method which can handle large displacement Brox & Malik (2010) (Brox-2010) and with the scene flow Huguet & Devernay (2007) (Huguet-2007). We can see, the results are even slightly better without noise for GCF then for GCSFs. This is because GCF allows non-bijective matching, while GCSFs insists on uniqueness which may cause small 1-pixel gaps of unmatched pixels between different motion layers. However, with increasing level of noise GCSFs outperforms its frame-by-frame seed growing counterpart. Results of Brox & Malik (2010) an Huguet & Devernay (2007) are comparable with GCSFs for low level of noise. For stronger noise these methods are significantly better than GCSFs. This is natural, since these global methods have reported excellent properties under perturbation by this kind of noise. Optical flow maps of GCSFs are shown in Figure 3.4g–3.4h. Object and motion boundaries are well preserved, there are no clear mismatches, there are a few 1-pixel gaps as mentioned above. Notice that, the motion occlusion on the bar, which is due to its motion behind the sphere in the next frame, has a 'correct' motion estimate, despite there is no evidence in data. This is a side effect of the prediction. Optical flow maps of Brox & Malik (2010) are shown in Figure 3.4k–3.4l. They are very precise inside the objects, however visually, there are some imperfections in motion boundaries of the objects.

Although the plot of Huguet & Devernay (2007) suggests its good overall performance, there are strong artifacts around the quickly moving bar, see Figure 3.4o–3.4p. Since the bar is relatively small with respect to the rest of the image, where the algorithm performs excellently, the error statistics do not reflect visually disturbing artifacts. Therefore, we evaluated the error statistics additionally in the area of the vertical bar only, see Figure 3.5b. Then, we can see the low performance of Huguet & Devernay (2007) compared to other algorithms.

The favorable results of the proposed GCSFs algorithm compared to the frame-by-frame independent seed growing methods are a consequence of: (1) joint

disparity and optical flow estimates which constrain each other, and (2) good temporal consistency and coherence. The mechanism is the following. When data is weak due to noise, there is a lack of correctly matched seeds and the growing process is either stopped early (by the condition in Step 7 of Alg. 2) for conservative choice of threshold τ, or produces mismatches if τ is relaxed. However, if we feed partially grown disparity and optical flow maps as the seeds to GCSF algorithm (using the predictor), it grows them further if they were correct. This effect is repeated, and after certain number of frames, high quality seeds are accumulated.

3.3.2 Real data

The proposed algorithm was tested on real data as well. For all these experiments, we used default values of parameters of the proposed GCSFs algorithm, $\alpha = \beta = 0.05$, $\tau = 0.6$. We show results on CAVA dataset of INRIA[1], where the stereo camera is static, and on the dataset of ETH Zürich[2] acquired by a mobile stereo platform.The results of tested algorithms are shown in Figure 3.6 and 3.7 as disparity $\mathbf{D}^1$ and optical flow $\mathbf{F}_h, \mathbf{F}_v$ maps.

For INRIA dataset, the results of the proposed GCSFs algorithm, Figure 3.6b–3.6d, are sufficiently dense even for weakly textured office environment. Important scene structures are matched. Notice sharply preserved boundaries between objects in both disparity and optical flow. We can see a left-down motion of the man coming through the door, which are closing afterward performing a slower left motion. One of the women is walking to the right to reach the chair, while moving her arm down. We can also recognize a hand gesticulation of the sitting man.

ETH dataset represents a complex scene with both camera forward motion and motion of pedestrians. There are up to 30 pixel displacements between consecutive frames. In our results, Figure 3.7b–3.7d, we can see a motion of the planar sidewalk close to cameras and well captured depth and motion boundaries of the people walking. There are only few small mismatches which are visible in

[1] http://perception.inrialpes.fr/CAVA_Dataset/
[2] http://www.vision.ee.ethz.ch/~aess/dataset/

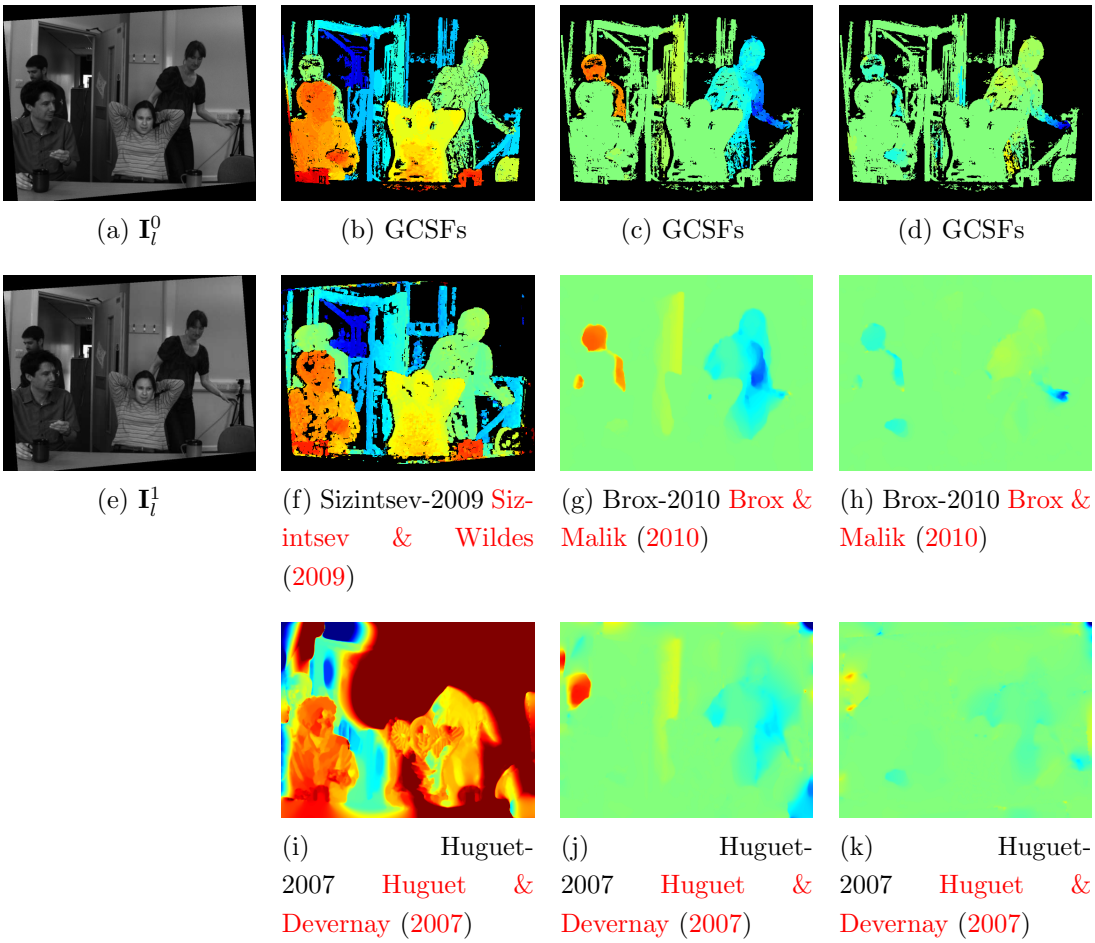
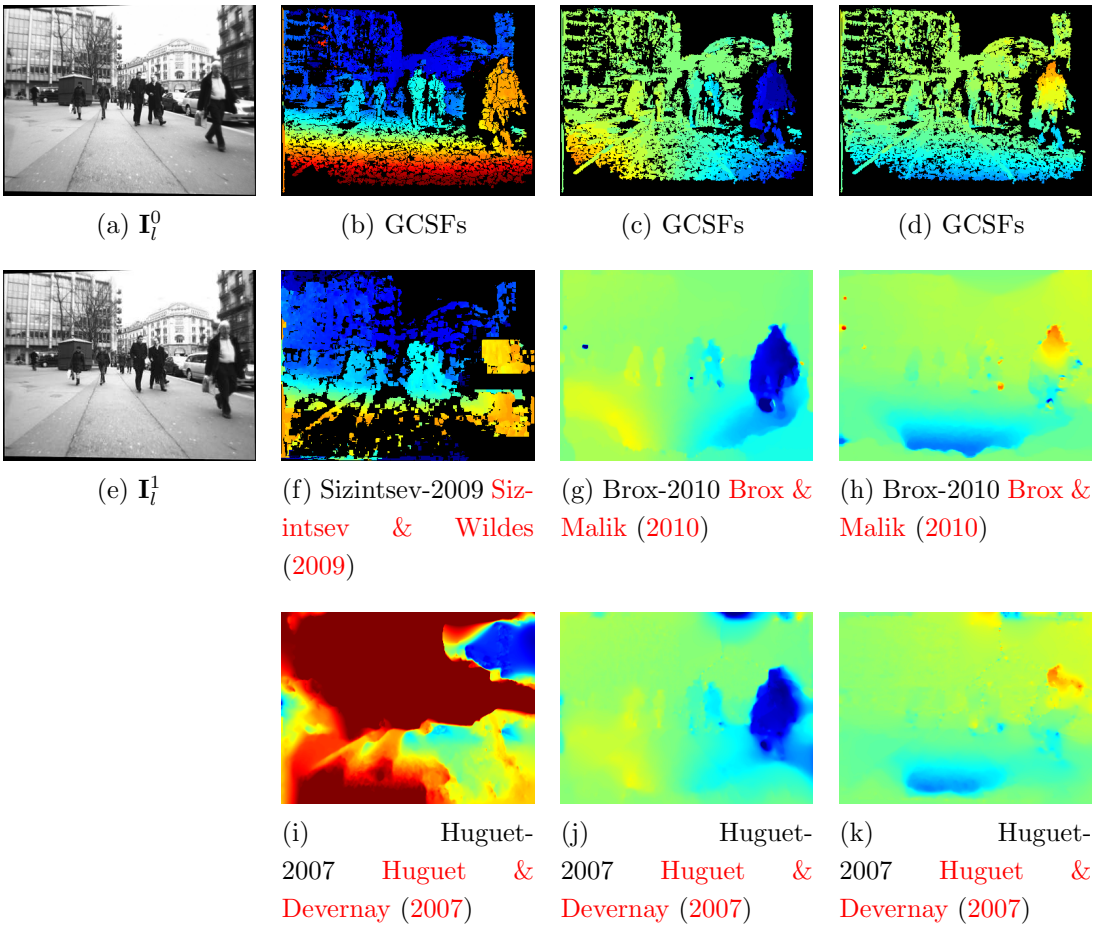


Figure 3.6: Real experiments: Results on INRIA dataset. This figure is better seen in the electronic version of the paper.



(a) $\mathbf{I}_l^0$ (b) GCSFs (c) GCSFs (d) GCSFs

(e) $\mathbf{I}_l^1$ (f) Sizintsev-2009 Sizintsev & Wildes (2009) (g) Brox-2010 Brox & Malik (2010) (h) Brox-2010 Brox & Malik (2010)

(i) Huguet-2007 Huguet & Devernay (2007) (j) Huguet-2007 Huguet & Devernay (2007) (k) Huguet-2007 Huguet & Devernay (2007)

Figure 3.7: Real experiments: Results on ETH dataset. This figure is better seen in the electronic version of the paper.

disparity map. This is in the region of the leftmost building which effects complicated non-Lambertian mirror like reflections. Some small mismatches can be found in optical flow in edge-like structures, which are consequence of improperly handled aperture problem.

Results of the spatiotemporal stereo Sizintsev & Wildes (2009) can be seen in Figure 3.6f and Figure 3.7f. Disparity maps were thresholded according to a stequel significance map to remove spurious matches. The threshold was set to 0.4 according to author's recommendation. After the thresholding, the disparity map on INRIA has roughly the same density as our result. However, the results are not so precise. It seems that all objects are fattened and especially those which moves in front of the weakly textured regions, see the walking woman and the man coming through the door in Figure 3.6f. These artifacts are probably caused by the large spatiotemporal extent of the matching elements (stequels). The method has severe difficulties with the ETH sequence. The part of the scene which is close to cameras and hereby undergoes a fast motion is not captured by this algorithm, Figure 3.7f. Matching of stequels probably does not work well for large displacement between frames.

Results of the large displacement optical flow Brox & Malik (2010) are shown in Figure 3.6g–3.6h and Figure 3.7g–3.7h. They are more or less consistent with our results, but they are fully dense. The motion boundaries seem to be a little bit fuzzy, but this could be only in the motion occluded regions, where there is no evidence in data. There are a few small patchy mismatches in ETH.

Results of the variational scene flow algorithm are shown Huguet & Devernay (2007) in Figure 3.6i–3.6k and Figure 3.7i–3.7k. The disparity maps are erratic, the algorithm fails dramatically in stereo for these scenes. This failure is probably due to a complexity of the scene (many occlusions, complicated motions, and varying strength of the texture), and perhaps also due to improper initialization and consequent problems with convergence. The optical flow given by this method is surprisingly much better than the stereo disparity. Nevertheless, we can see typical artifacts of smoothed motion boundaries, which is a consequence of the prior term winning over the data.

GCSFs	1.5 seconds
Sizintsev-2009 Sizintsev & Wildes (2009)	35 seconds
Brox-2010 Brox & Malik (2010)	3 minutes
Huguet-2007 Huguet & Devernay (2007)	3 hours

Table 3.1: Average running time per frame of VGA images.

For both sequences, our results are temporally coherent without flickering artifacts, which is not the case of results using Sizintsev & Wildes (2009) and Huguet & Devernay (2007). Results of Brox & Malik (2010) are fairly stable temporally, despite computed frame-by-frame.

3.3.3 Running time of tested algorithms

An average running time per frame of the tested algorithms is shown in Tab. 3.1. These times were measured on our synthetic sequence of 640×480 images, using a standard PC (Intel Core 2 2.6 GHz, 6 GB memory, Linux). Our GCSFs algorithm is faster by order of magnitudes than the other tested methods. Our implementation is not optimized and partially in Matlab. For the other algorithms we had binaries.

3.4 Discussion

We presented an algorithm which jointly estimates semi-dense disparity and optical flow of a stereo sequence by growing correspondence seeds. We experimentally proved that results are more accurate and temporally coherent than frame-by-frame independent algorithms.We tested with two different publicly available datasets and performed a quantitative ground-truth experiment. We made a fair comparison with state-of-the art methods spanning over spatiotemporal stereo, and variational methods for optical and scene flow.

The proposed algorithm is a practically well working trade-off between simple local methods and theoretically sound global MRF algorithms, since local

relations between adjacent pixels are considered. It can be also viewed as a 'semi-supervised' matching algorithm, where a few initial seeds are propagated.

Chapter 4

Descriptor for Action Recognition

4.1 Introduction

An extensive research has been done in action recognition throughout recent years, which is well documented in survey papers Poppe (2010); Weinland *et al.* (2011). Most of the methods work with monocular videos only. Very successful methods use image retrieval techniques, where each video sequence is represented as a histogram of visual words Laptev (2005), and large margin classifier is then used for recognition.

In particular, spatiotemporal interest points Laptev (2005) are detected in the image sequence. These points are described by a descriptor HoG (Histogram of Gradients)/HoF (Histogram of Optical Flow) Dalal & Triggs (2005) which captures the surrounding of an interest point. The descriptors are quantized by K-means clustering and each video clip is represented as a histogram with K bins. Support Vector Machine is then used for classification.

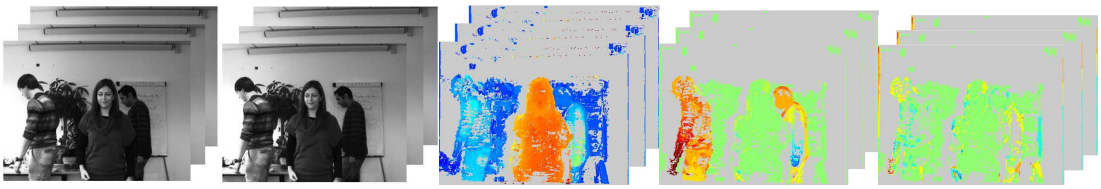
Further research to improve the recognition accuracy went in the direction of densifying the interest points and enhancing the local descriptors. The interest points employed in Laptev (2005) are spatiotemporal extensions of a Harris corner detector, i.e. locations in a video stream having large local variance in

both spatial and temporal dimensions, representing abrupt events in the stream. This is in order to achieve high repeatability of the detection. However, such points are quite rare and important relevant information can be missed. Therefore there were alternatives to these interest points, e.g. based on Gabor filters Bregonzio *et al.* (2009); Dollár *et al.* (2005), or even simply using a regular dense sampling Wang *et al.* (2009) to reach higher coverage, or a hybrid scheme by Tuytelaars (2010), which start by dense sampling and optimize the position and scale within a bounded area in order to increase the coverage and preserve the repeatability of the interest points. An extension of the original HoG/HoF descriptor was proposed e.g. by spatiotemporal gradients Klaser *et al.* (2008), or motion boundary histograms Wang *et al.* (2011).

However these methods can be quite sensitive to background clutter present in populated scenes, since interest points are detected not only in the actor but on the background as well. This causes the global histogram representation to be corrupted and the accuracy is significantly decreased.

Stereo vision or multiple view vision have not been much used in action recognition. Using stereo, the existing methods typically try to make the algorithm insensitive to a camera viewpoint Roh *et al.* (2010). Similarly Weinland *et al.* (2007) uses a special room and a multi-camera setup to construct viewpoint invariant action representation, and Yan *et al.* (2008) incorporate temporal information to the multi-view setup. Work Uddin *et al.* (2011) uses the depth map obtained by stereo matching to fit an articulated body model and use joint trajectories for action recognition.

An alternative to stereo vision is using RGB-D sensor, which provides a depth image besides the color/intensity image. It is based on time-of-flight or structured light technology. This research is vivid nowadays due to the recent irruption of Kinect device. For instance Holte *et al.* (2010) constructs 3D motion primitives from a cloud of 3D points. Work Li *et al.* (2010) extends 2D silhouette by projection of the point cloud into three orthogonal planes. In Zhang & Parker (2011) the authors uses local interest point descriptors which are computed from spatiotemporal image and depth gradients for each pixel of a spatiotemporal neighbourhood of interest points. Since the neighbourhood is large, they use



(a) (b) (c) (d) (e)

Figure 4.1: Example of data for one sequence. The input data consists of sequences of (a) Left and (b) Right images. The maps of (c) disparity, (d) horizontal, (e) vertical component of the optical flow computed by algorithm Čech *et al.* (2011). The maps are color-coded: gray color means unassigned value, for disparity warmer colors corresponds to points closer to the camera, for optical flow warmer colors corresponds to motion to the left and up respectively.

PCA for dimensionality reduction prior to quantization. In Ni *et al.* (2011), spatiotemporal interest points are divided into different layers based on depth and a multichannel histogram is created. Another direction is to estimate the body skeleton from the depth data. Commercially successful real-time game controller uses skeleton model from body part labelling of depth data of Kinect Shotton *et al.* (2011). Joint trajectories are used for action or gesture recognition in e.g. Sung *et al.* (2012); Xia *et al.* (2012). However, for some applications such active sensors are not suitable. For example, in outdoor setup or in a scenario with multiple autonomous robots whose active sensors would interfere to each other.

Therefore we propose a simple stereo vision based method, which can focus the algorithm to an active actor while disregarding the background activity based on completely passive system, see Figure 4.1. Our contribution is extending the original successful action recognition framework Laptev (2005) with descriptors based on stereo vision and scene flow. We observed a significant improvement of the proposed method in the robustness to the perturbations due to the uncontrolled motion of other people behind the actor.

4.2 Method Description

Before we give details on the proposed descriptor, we briefly revise the bag-of-words (BoW) paradigm for action recognition. Following Laptev (2005) it requires to:

1. Collect a set of local descriptors associated to the interest points for image frames of all training action video clips.
2. Apply clustering algorithm to these descriptors, for instance, K-means.
3. Quantize the descriptor to get the 'visual words'. For each descriptor, assign label according to its nearest cluster centroid.
4. Represent a video clip as a K-bins histogram of the quantized descriptor ('bag of words').
5. Train a classifier with these histograms, for instance, SVM.

In Steps 1–3, the the visual word vocabulary (or the codebook) is constructed. The dimensionality of the local descriptor is typically high and the space is consequently sparse, that is why it is represented by K clusters of observed data. In Steps 4–5, a compact (K-length vector) representation of training video clips with annotated labels is used to train a classifier. The 'bag of words' representation encodes a relative frequency of occurrences of the quantized descriptors and it turns out to be discriminative among action classes. Later for recognition, an unknown video clip is first represented as the K-length histogram and then it is fed to the classifier which assigns the class label.

We follow exactly this framework, except for the Step 1. Unlike the monocular HoG/HoF descriptor Laptev (2005), we introduce a new descriptor based on the scene flow described in Chapter 3.

(a) (b) (c) (d) (e) (f)

Figure 4.2: Construction of the proposed descriptor. The actor's face is detected from the left input image (a). The raw disparity map (b) is segmented, such that all pixels having the lower disparity than the actor's face are discarded (c). The descriptor is then computed for all remaining pixels undergoing non-zero motion, such that it consists of the pixel's position relative to the face, it's disparity (d), and horizontal (e) and vertical (f) components of optical flow.

4.2.1 Local Descriptor based on the Scene Flow

The Scene flow is a 3D extension of the optical flow. We represent a scene flow as depth and optical flow, which together with a camera calibration is equivalent to a vector field of 3D position and associated 3D velocities of reconstructed surface points. This intrinsic representation is potentially less sensitive to the changes of texture and illumination in the action dataset than the representation which relies solely on the intensity images. Moreover, with the notion of depth, it is straightforward to focus the actor performing the action to be recognized while discarding any activity from the background clutter.

We assume the action performing actor is the person which is the closest to the camera. We believe this is a reasonable assumption, which is typically the case of human-robot interaction or movies.

The proposed descriptor is constructed as follows, see Figure 7.4:

1. Get the synchronized sequences of the left $\mathbf{I}_l$ and right images $\mathbf{I}_r$. For each frame compute the disparity map $\mathbf{D}$ and optical flow maps $\mathbf{F}_h, \mathbf{F}_v$ by the algorithm Čech *et al.* (2011).

2. Find the actor's face with a face detector Šochman & Matas (2005): $(x_0, y_0) = \mathrm{FD}(\mathbf{I}_l)$. In case of multiple faces detected, the one with the highest dispar-

ity $d_0 = \mathbf{D}(x_0, y_0)$ is selected[1]. In case no face is detected, if the actor turns or the detector miss the face, we simply assume a previous face position.

3. Segment the scene using disparity and optical flow: (1) Only pixels with magnitude of optical flow greater than zero are considered, (2) Only pixels with disparity greater or equal to the disparity of the actor's face are considered. So the set of valid pixels

$$S = \{(x, y) : \mathbf{F}_h(x, y)^2 + \mathbf{F}_v(x, y)^2 > 0 \text{ and } \mathbf{D}(x, y) > d_0 - \mu\},$$

where $\mu = 5$ is a small margin to ensure the entire actor's body is included.

4. At each reconstructed pixel passing the above test $(x, y) \in S$, the local descriptor is 5-dimensional only:

$$L(x, y) = \Big(x - x_0, y - y_0, \mathbf{D}(x, y) - d_0, \mathbf{F}_h(x, y), \mathbf{F}_v(x, y)\Big).$$

Notice the face normalized position of the pixels, brings a kind of global information into the local descriptor.

Following the BoW procedure described above, after building the codebook and subsequent quantization of pixel descriptors, the resulting histograms of their occurrences in the action video sequence intuitively encodes the activity of actor's body parts in the sense of 3D motion. See Figure 4.3 for an illustration.

4.3 Experiments

To evaluate the performance of the proposed binocular method and compare it with a state-of-the-art monocular method Laptev (2005), we use the Ravel dataset, see Appendix A. The Ravel dataset consists of 7 actions (talk phone, drink, scratch head, turn around, check watch, clap, cross arms) performed by 12 actors in 6 trials each. First 3 trials are with stable static background without other people in the scene (we denote as 'Controlled'), while next 3 trials

[1]The disparity of the face is estimated as an average disparity inside the bounding box obtained from the face detection. The center of the bounding box is the pixel (x_0, y_0).

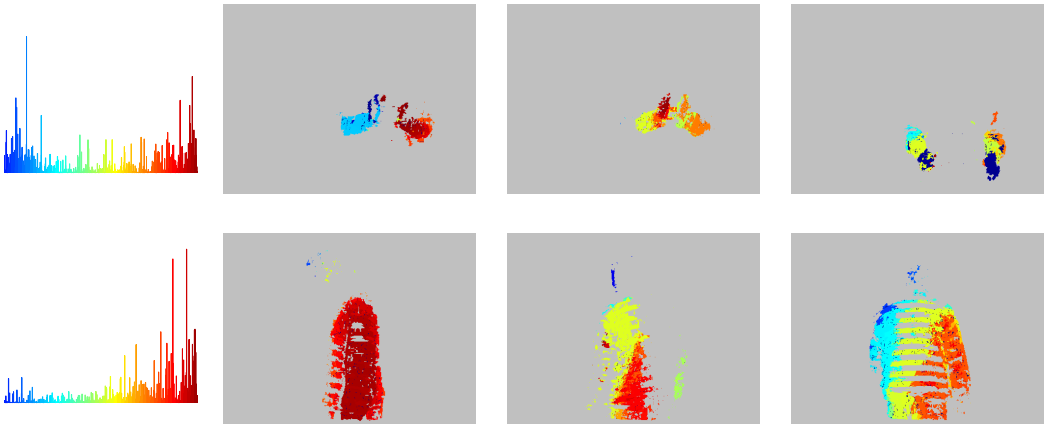


Figure 4.3: Histograms of visual words and corresponding assignment to pixels for frames of two actions: clap (top) and turn-around (bottom). The color encodes the indices of visual words $1, \ldots, K$. The coloring is such that similar visual words have similar color. We can see typical visual words occurring during the actions.

are performed with motion background clutter due to arbitrary activity of the people behind the actor (we denote as 'Clutter'). See Figure 4.4 and Figure 4.5 for respective examples. The dataset is challenging due to the strong intra-class variance, strong dynamic background in the 'Clutter', and unstable lighting conditions.

We will show results of two baseline algorithms. The first one is the algorithm described in Laptev (2005) works with monocular (left camera) stream only and uses the sparse spatiotemporal interests points and HoG/HoF descriptors, we denote as 'STIPs'. The other baseline is the same algorithm, however we ran it in both left and right camera sequences, matched the detected points along the epipolar lines, and removed the interest points which have smaller disparity than the disparity of the actor's face. The motivation behind is to remove the irrelevant interest points detected on the background clutter. The rest of the algorithm Laptev (2005) remains the same. We call this algorithm 'STIPs-stereo'. The proposed method described in Sec. 4.2, is denoted as '5DF'.

The codebook was built in a sequence of a single actor, namely 'character-

Figure 4.4: Ravel dataset examples - controlled setup. Note that different actors perform the same action quite differently as for example in "cross arms". Actions: "cross arms", "check watch", "scratch head", "cross arms", "talk phone", "cross arms", "scratch head", "clap".

09'. This actor was not later used either for learning a classifier or for testing. We believe a single actor performing the same set of actions as all other actors sweeps the space of local descriptors is enough and also K-means algorithm is run only once and not in the leave-one-out loop (see later), which would be too time consuming. The size of the codebook K was optimized for all the methods in the logarithmic range from $K = 10$ to $K = 10000$ and the optimum was found for $K = 1000$, the same for all the methods.

Learning a classifier and testing was performed in a standard leave-one-actor-out scenario. One actor was removed from the set, the linear SVM classifier was trained in the sequences of remaining actors and then tested on the sequence of the left actor and this was repeated for all actors. The recognition rate reported is the average error over all actors.

Results are shown in Table 4.1. We can see the proposed method (5DF) performs comparably in the setup when there is a single actor in the scene only. This proves the proposed descriptor computed in the meaningful semi-dense locations is informative. Furthermore, we can see the recognition accuracy of the proposed method does not drop much in cases of the background clutter of other people

Figure 4.5: Ravel dataset examples - cluttered setup. Actions: "turn around", "clap", "talk phone", "talk phone", "turn around", "drink", "check watch", "drink". Note different illumination conditions.

freely moving behind the actor. This demonstrates that the algorithm can properly focus the active actor while disregarding the background activity using the depth information from stereo. The monocular baseline method Laptev (2005) (STIPs) is naturally very sensitive to this type of the background clutter. The algorithm cannot distinguish the informative interest points of the clutter from corresponding descriptors on other people in the scene, which contaminates the histograms and the recognition accuracy drops significantly. The second baseline (STIPs-stereo), which attempts to remove the interest points detected on the background by stereo matching, is less sensitive to the background clutter, however its recognition accuracy is slightly lower for 'controlled' setup. The reason is that the sparse spatiotemporal interest points become even sparser, since the stereo matching may discard also points on the foreground due to matching ambiguity. Notice that in STIPs method, we have about 10 interest points per frame, but in our method we have about 10000 locations per frame where descriptors are computed.

For more insight, we show confusion matrices of both methods for both 'controlled' and 'clutter' setups, see Figure 4.6–4.8. For instance, we can see that scratch head is confusing with talk phone. This is not so surprising since these

Algorithm	Controlled	Clutter
STIPs Laptev (2005)	0.6883	0.4675
STIPs-stereo	0.6537	0.5238
5DF (the proposed method)	0.6840	0.6494

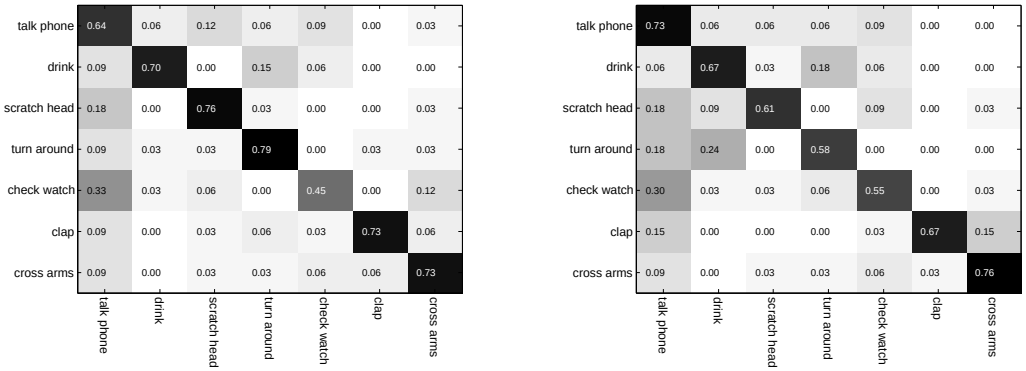
Table 4.1: Recognition accuracy of the tested methods. The proposed (5DF) method has comparable results with state-of-the-art method (STIPs) in the controlled setup with only one actor in the scene, while it much less sensitive to the strong dynamic background clutter. The other baseline (STIPs-stereo) is less sensitive to the background by using the stereo information, however due to insufficient coverage of interest points the recognition accuracy is lower.

actions starts with the hand at the level of the pocket and is directed to the head, where the difference is whereas it remains static (talk phone) or moving (scratch head). Again, there is significantly much less confusion in case of the background clutter in the proposed binocular method compared to the state-of-the-art method which only uses a monocular video. This corroborates that stereo vision brings an important extra information.

4.4 Discussion

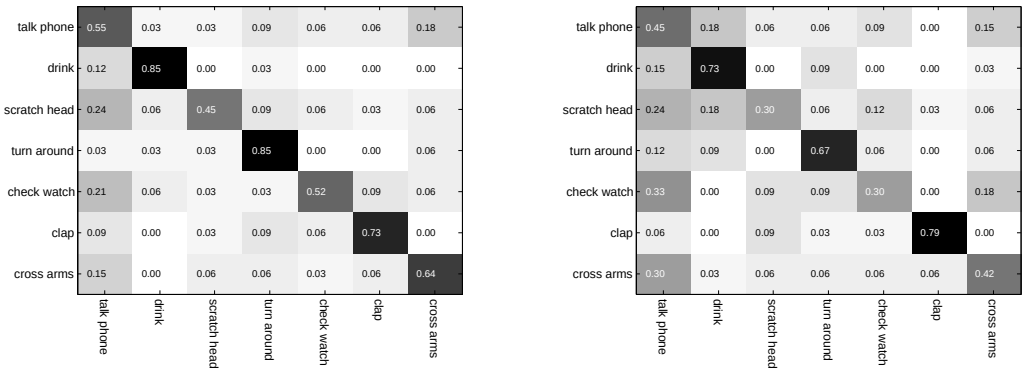
We presented an action recognition method which uses the scene flow computed from binocular video sequences. Experimentally we proved that the extra information from stereo significantly improves the recognition accuracy in the presence of strong background clutter.

The proposed method requires the actor's face is detected in majority of the frames. We expect that a tracker with a motion model would help to localize the face if it is turned away. Future work includes an elaboration on the design of the local descriptor. Combination of the local descriptor with the proposed one could further improve the recognition accuracy.



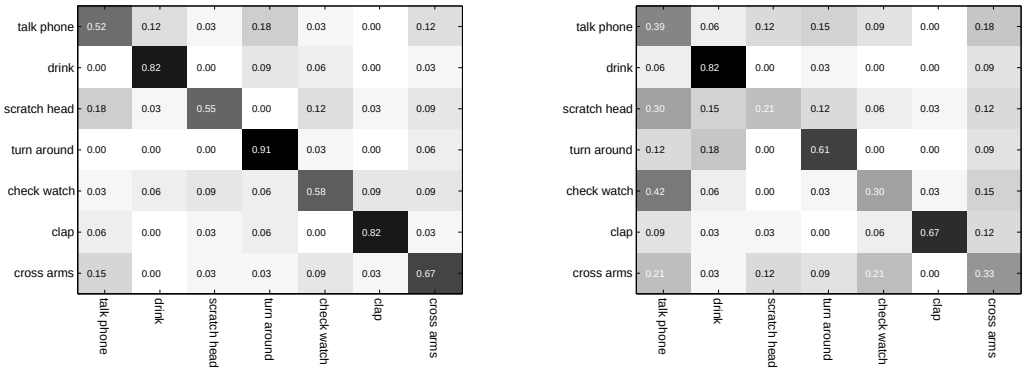
(a) Controlled (b) Clutter

Figure 4.6: Confusion Matrix for the proposed method (5DF) for a) Controlled and b) Cluttered setup.



(a) Controlled (b) Clutter

Figure 4.7: Confusion Matrix for the STIPs-stereo for a) Controlled and b) Cluttered setup.



(a) Controlled (b) Clutter

Figure 4.8: Confusion Matrix for the state-of-the-art method Laptev (2005) (STIPs) for a) Controlled and b) Cluttered setup.

Chapter 5

Adding Audio to Action Recognition

5.1 Introduction

For more natural communication with a humanoid robot, especially in populated environments, it would be desirable, that the robot can recognize several voice commands accompanied by gestures, and then it immediately behaves adequately without a noticeable latency. In this chapter we address the problem of audio-visual action recognition for social robots.

This task comprises a field of action/activity recognition, which has been widely studied in the literature recently. There are approaches where the action recognition is based on *wearable sensors* of the subject. For instance Koenemann & Bennewitz (2012) designs a system where humanoid robot NAO imitates complex motions of the human subject. However, there is no action recognition involved, unlike in Roggen *et al.* (2011) or in Zhu & Sheng (2009). The wearable sensors collects data as acceleration, rotation, etc. and this information is then used for training HMM models of the particular human activity.

Another deeply studied approach is using a *video camera* instead of expensive and uncomfortable wearable sensors. To name a few, authors Zhou *et al.* (2009) are interested in surveillance related actions such as fall detection. They use a

(a) (b) (c)

Figure 5.1: Audio-visual action recognition at a glance. (a) The user performs a action for NAO ('hello'). (b) The robot extracts features coming from the two cameras and one of the microphones and recognizes the action. (c) NAO performs the answer to the recognized action.

single camera, extract Hu moments, aspect ratio bounding box of the silhouette, etc. and they train an HMM on a publicly available dataset.

A humanoid robot could also serve inside a household. As in Volkhardt *et al.* (2010) two kind of actions are detected. Long term actions such as wash dishes and short term actions such as drinking. To this end, the features proposed are as eclectic as the height of the person and the position of the body that is tracked with a Kalman filter using leg and face detector. This is afterwards trained on a Bayesian Network. Another scenario could be in the kitchen. In Gehrig *et al.* (2011) several problems are addressed: Activity, Motion and Intention Recognition. For each one of the problems different features and learning methods are proposed. SVMs are used for activity recognition, HMMs for motion recognition, and Hybrid Dynamic Bayesian Models for intention recognition.

A more general approach is found in Jenkins *et al.* (2007a,b), where an offline trained motion model is used. Human motion is tracked, and silhouettes extracted to learn a vocabulary of motion primitives.

Recently, very popular approach uses *RGB-D (color+depth)* sensor, typically widely available Kinect. The actions are recognized usually via human skeleton model estimated from the depth image Shotton *et al.* (2011). These are for instance papers Ramey *et al.* (2011); Raptis *et al.* (2011); Sung *et al.* (2012);

Xia *et al.* (2012). These algorithms have impressive results, however due to the necessity of the IR structured light, they are not applicable outdoor for instance.

Another way to overcome weaknesses of individual sensors is the *sensor fusion*. Most prominently the video and audio modality are used because of their availability and complimentary nature. The intuition is that camera images can be corrupted by various kind of noise (low light, saturation, blur, occlusions, etc.) as well as audio data (microphone noise, noise from the robot's fan, another sound source in the scene, reverberations, etc.), however statistical fusion of these modalities can significantly improve the robustness and the recognition rate.

An example of such fusion method, but in this case devoted to object recognition, is Lacheze *et al.* (2009), in which the authors experiment different combination strategies for object detection. Visual features are based on texture description and entropy-based variable-size patches. Auditory features correspond to the energy of the signal's gammatone filter bank decomposition. Monocular video and monaural audio are used and there is a strong need of uniform visual background.

Paper Lopes & Singh (2006) targets general activity recognition. They use an early fusion, where high dimensional features (around 3000) are constructed for video and audio. This dimensionality is then reduced using the sequential forward floating selection (SFFS) algorithm to select most relevant features to low dimension (about 40). Finally a kNN algorithm is used as classifier.

We propose an algorithm that performs action recognition fusing data from two different modalities - visual and auditory. Visual features are based on disparity maps computed by stereo matching of two synchronized images streams and MFCC features are used as auditory. The proposed method uses a descriptor explained in Chapter 4 as visual features, and a late fusion strategy to combine the data from both modalities. Notice we propose a completely passive system (no active structured light sensors are involved) and fully robo-centric (all the sensors are on board of the humanoid robot NAO). We show an implementation, in Chapter 7, which runs on-line with a small latency.

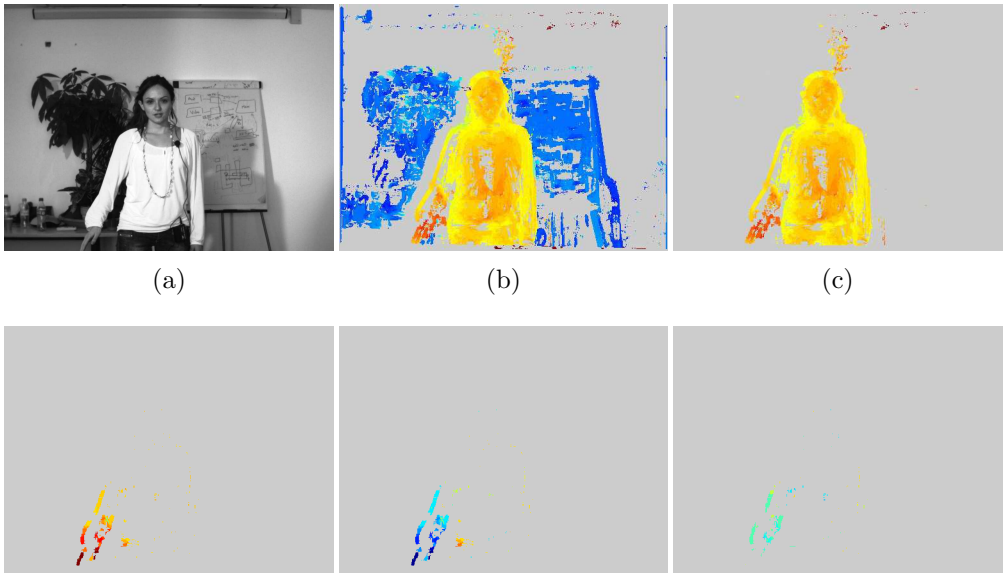
Most of the methods named so far use a single camera, some of them use Kinect sensor, but none of them to our best knowledge is using multi-modal information such as auditory and visual implemented on a humanoid robot. Moreover, the interaction with the robot is not clearly defined, despite most of the papers claim to use robotic platforms. We define a simple communication protocol that allows to interact with NAO robot in order to have a response accordingly to the input received. Also the method proposed has the potential to work in an environment with multiple persons Sanchez-Riera *et al.* (2012) where the algorithms reviewed so far a single person action recognition is addressed.

5.2 Audio-Visual Categorization

This section is devoted to the proposed audio-visual action learning approach, which performs classification-level fusion. By means of the scene flow, we are able to describe the visual information, see section 5.2.1. The auditory information is characterized by standard features used in speech recognition (section 5.2.2). The learning is performed through a traditional SVM framework and finally, the procedure to combine the output of the uni-modal classifiers is described in section 5.2.3.

5.2.1 The Visual Descriptor

We used a slightly modified visual descriptor than proposed in Chapter 4, which is based on the scene flow. The scene flow is represented by the optical flow plus the depth at each image position. Together with the camera calibration, this is equivalent to a vector field of 3D position and associated 3D velocities. This intrinsic representation is potentially less sensitive to changes of texture and illumination than the intensity images. Moreover, the notion of depth allows to focus on the actor, while discarding any activity from the background. We assume that the actor of interest is the person closest to the camera. This is a reasonable assumption, since it holds in most of the human-robot-interaction applications and on movies. The original descriptor consists on the position and



(a) (b) (c)

(d) (e) (f)

Figure 5.2: Construction of the scene flow descriptor. The actor's face is detected from the left input image (a). The raw disparity map (b) is segmented, such that all pixels having the lower disparity than the actor's face are discarded (c). The descriptor is then computed for all remaining pixels undergoing non-zero motion, such that it consists of the pixel's position relative to the face, it's disparity (d), and horizontal (e) and vertical (f) components of optical flow.

disparity relatives to the actor's face plus the optical flow (see Figure 5.2 for a detailed example).

However, currently there is no real time implementation of scene flow algorithm for NAO platform. This fact, forces us to segment only by disparity and to reduce the original descriptor from five dimensions to three dimensions. Although, one could think that this change will influence greatly in the final average recognition rate, we demonstrate that the impact of optical flow data when combined with auditory information is minimal at its optimal combination value as shown in Figure 5.3.

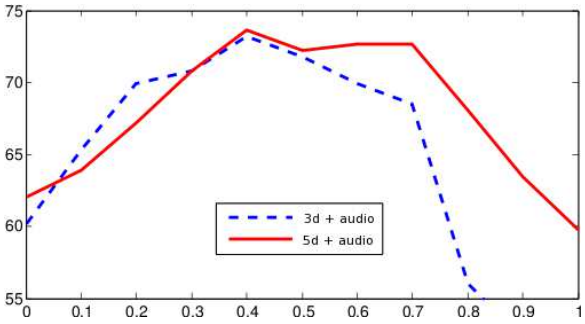


Figure 5.3: Average recognition rate for fused data in case of descriptor of Chapter 4 (5d) combined with audio, and modified descriptor without optical flow information (3d) combined with audio, for different weighting between visual and auditory information.

5.2.2 The Speech Descriptor

The auditory stream is represented by the Mel Frequency Cepstral Coefficients (MFCC). Widely used for speech and sound recognition (see Rabiner & Schafer (2011); Ramasubramanian *et al.* (2011)), the MFCC are computed following the three steps: (i) perform the short-time Fourier transform (STFT), (ii) map the power spectrum onto the Mel scale and (iii) take the discrete cosine transform of these mapped powers. The are three main parameters associated with MFCC features. First, the frame size defines the length of the STFT (denoted by W). Second, the frame shift (F) determines the time between two consecutive STFT windows. Third, the amount of cepstral coefficients (D), that sets the dimension of the output MFCC representation.

5.2.3 Fusing audio-visual data

The BoW representation, reviewed in Chapter 4.2, encodes the relative frequency of occurrences of the quantized descriptors, which discriminates among action classes. We use the BoW paradigm to build auditory and visual models for each of the actions. Hence we have both a visual and an auditory classifier. When an instance of an unknown action class has to be recognized, the auditory and visual representations are computed and sent to their respective classifiers.

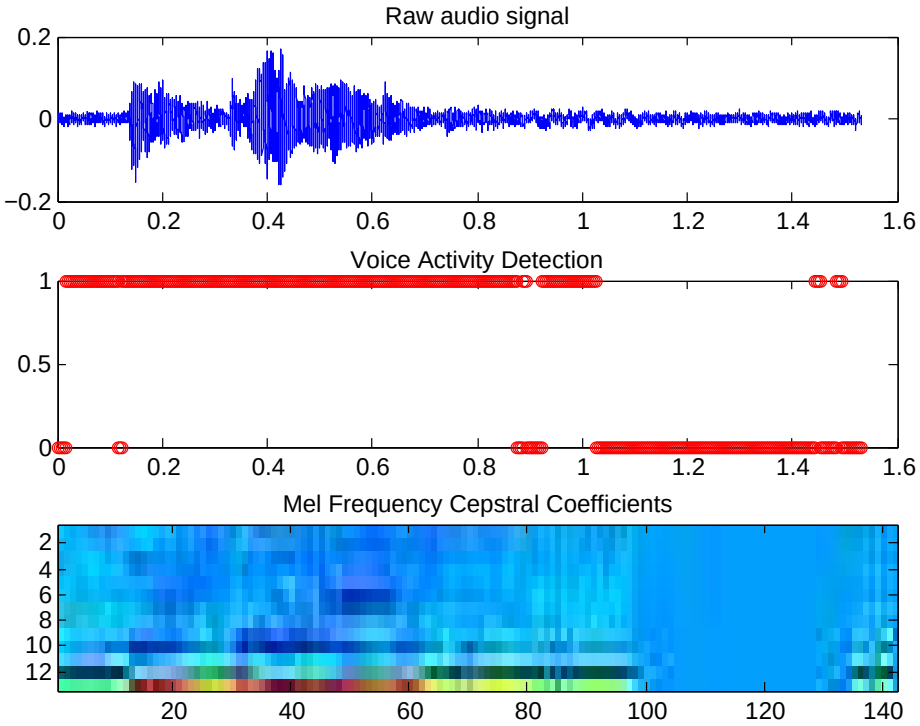


Figure 5.4: Mel Frequency Cepstral Coefficients for one voice-command instance. From the raw signal (top) the voice activity is detected (middle) and used to mask the extracted MFCC (bottom).

The outputs of the two linear SVMs classifiers are fused to perform audio-visual gesture recognition.

Let $a_c(g)$ and $v_c(g)$ denote the score of the action instance g to belong to class c given by the auditory and visual classifiers respectively. In order to combine the information from both classifiers we train a combined classifier consisting on (i) whitening the training data (uni-modal classifier scores) and (ii) apply a weighting function.

The whitening procedure consists on computing the mean (μ_a) and the standard deviation (σ_a) of the auditory classification scores $\{a_c(g_n)\}_{n=1,c=1}^{N,C}$, being N the number of command instances. A new auditory score is computed as $\tilde{a}_c(g) = \frac{a_c(g)-\mu_a}{\sigma_a}$. The same procedure is applied to the visual classification scores.

Finally, the combined score is the result of a convex combination of the two whitened scores:

$$m_c^l(g) = l\tilde{v}_c(g) + (1-l)\tilde{a}_c(g).$$

The value of l determines the trust we put on each modality. Actually, some cases deserve a special mention:

$l = 0$ is equivalent to audio-based classification.

$l = 0.5$ the auditory and visual scores stand on equal foot,

$l = 1$ is equivalent to vision-based classification, and

In general, $l > 0.5$ means that we put more trust on the visual classification score, whereas $l < 0.5$ means that we do it with the auditory score. This way of combining the two classifiers allows us to evaluate the relative trust we put on the modalities. The final classification is:

$$c^* = \arg\max_c m_c^l(g).$$

5.2.4 Boundary Action Detection

Since the Bag-of-Words framework is designed to perform isolated gesture recognition, we need to define the start and the end of each gesture instance to run it on NAO. This is important to determine the boundaries of the action to recognize. Hence, it triggers the computation of the visual and auditory descriptors as well as the categorization. In our case, the bounds are determined following a simple rule. When the detected motion in the left image exceeds a threshold, the systems gathers auditory and visual features, building both descriptors, during a fixed-length time interval. Hence the user has to point to the robot that a action has to be categorized, to further on, perform the action within a certain amount of time. This reduces the interaction with the robot, but allows to test the audio-visual gesture recognition framework on-line.

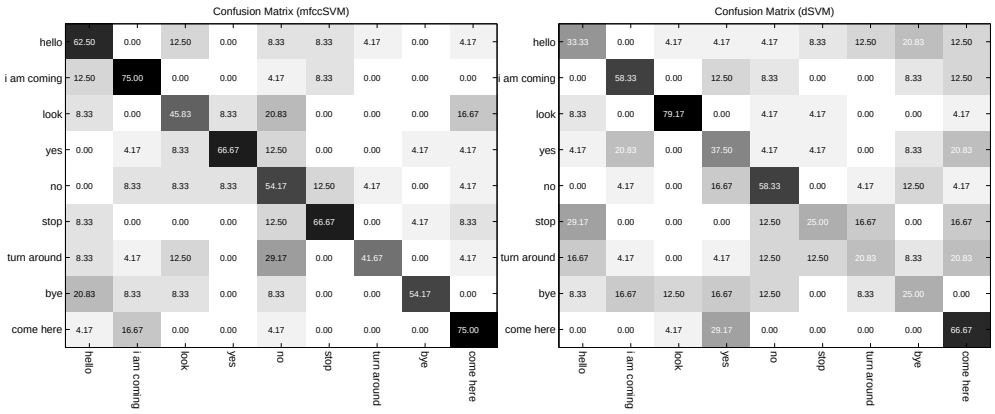
5.3 Experimental Validation

In order to validate the proposed approach, we need to perform two experiments. A first experiment to analyze the performance of the proposed visual and auditory features. A second experiment to test the proposed audio-visual recognition system implemented on NAO. Evaluating multi-category classifiers means providing the confusion matrix. The ij-th entry of such matrix contains how many instances of the i-th class have been classified as class j. By averaging the elements of the diagonal, one obtains the average recognition rate (ARR) of the classifier. Furthermore, to have a statistically significant quality measure, a leave-one-out strategy is used to cross-validate the method within actors.

The first experiment is done with the publicly available dataset Ravel, see A. We use the Robot Gestures part of the data set which consists on eight actors performing a set of nine actions: 'yes', 'no', 'come here', 'turn around', 'hello', 'I'm coming', 'look', 'stop' and 'bye'. The actor always accompanies the gesture with some word/action. Each gesture is performed three times under background clutter and three times in a more controlled level of clutter.

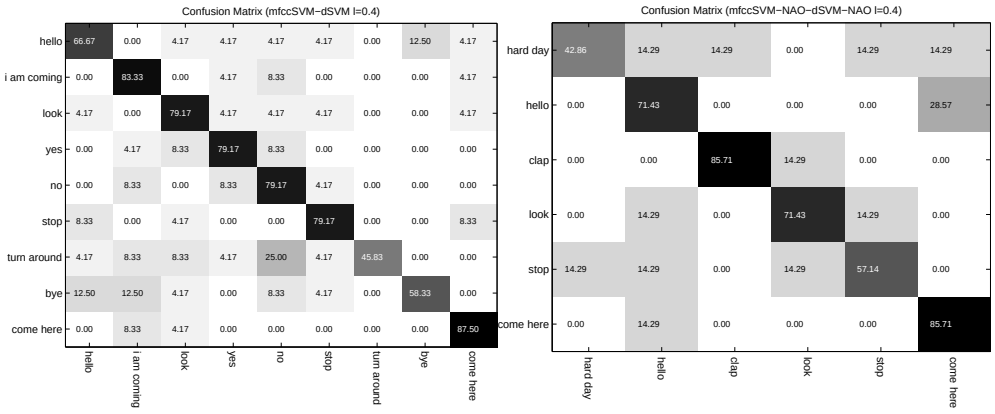
To support this idea, we also plot the confusion matrices of the audio-only classifier (Figure 5.5(a)), the video-only classifier (Figure 5.5(b)) and the multi-modal classifier for l^* (Figure 5.6(a)). Notice that there are three main confusion in the MFCC-based classifier: 'look' as 'no', 'bye' as 'hello' and 'turn around' as 'no'. While the first two are well discriminated by the video-only classifier, the third one is also confused, together with a few others (e.g.: 'hello' as 'bye', 'stop' as 'hello', 'come here' as 'yes', etc). We observe that this confusion in the mono-modal classifiers are remarkably reduced in the combined classifier when one of the modalities has high discriminative power. However, in the case of 'turn around' as 'no', where both auditory and visual classifiers are confused, the multi-modal classifier is, of course, also confused.

The second experiment validates the on-line implementation within the RSB ecosystem and proves the validity of the proposed system. The exact same evaluation strategy is used. However, the data set used is slightly different, since we needed data from NAO for this experiment. In this case, we recorded six actions



(a) Audio-based (b) Video-based

Figure 5.5: (a) Confusion matrix of the audio-based classifier. Three main mistakes: 'look' as 'no', 'bye' as 'hello' and 'turn around' as 'no'. The ARR obtained is around 60%. (b) Confusion matrix of the video-based classifier. Several big mistakes, e.g.: 'hello' as 'bye', 'stop' as 'hello', 'come here' as 'yes', 'turn around' as 'no', etc. The ARR is around 40%.



(a) Audio-Video Ravel (b) Audio-Video NAO

Figure 5.6: (a) Confusion matrix of the multi-modal classifier for $l = l^* = 0.4$. Just one big mistake (the only one shared by the two mono-modal classifiers), 'turn around' as 'no'. The ARR is around 73%. (b) Confusion matrix of the multi-modal classifier trained with the data acquired with NAO. One big mistake is made confusion 'hello' by 'come here'. The resulting average recognition rate is 68%.

('hard day', 'hello', 'clap', 'stop', 'look', 'stop' and 'come here') performed by seven actors. Each actor performed each action once. Figure 5.6(b) shows the confusion matrix for the optimal value of l (in that case $l^* = 0.4$). The average recognition rate of the audio-visual gesture recognition system proposed is 68%.

5.4 Discussion

We presented a system for audio-visual gesture recognition working on the humanoid robot NAO. Based on a modality weighting technique, the result of two mono-modal classifiers is mixed, building a multi-modal classifier. A bag-of-words approach using simple auditory and visual features is the main learning paradigm. The method has state-of-the-art performance. Implementation on NAO, explained in Chapter 7, gives us an average recognition rate of 68%.

Chapter 6

Audio Visual Fusion for Speaker Detection

6.1 Introduction

Humanoid robots acting in populated spaces require a large variety of communication skills. Perceptive, proprioceptive as well as motor abilities are mandatory to make the information flow natural between people and robots participating in interaction tasks. On the perceptive side of the communication process, the tasks are mainly detection, localization and recognition. Depending on the available sensory modalities, the robot should be able to perform tasks such as: sound/speech detection/recognition, action/gesture recognition, identity/voice recognition, face detection/recognition, etc. Moreover, if several modalities are combined, multi-modal tasks such as audio-visual event recognition or audio-visual speech processing are known to be more robust than uni-modal processing and hence multi-sensory perception can drastically improve the performance of a large variety of human-robot interaction activities.

The problem of data fusion and multi sensory integration has been recognized for a long time as being a key ingredient of an intelligent system, *e.g.,* Luo & Kay (1989). More recently, multi-modal integration has been used in action recognition applications Lili (2009); Wu *et al.* (2010). In these papers, the authors

Figure 6.1: A typical scenario in which a companion humanoid robot (NAO) performs audio-visual fusion in an attempt to detect the auditory status of each one of the speakers in the room. The system described in this chapter processes the raw data gathered with the robot's camera and microphone pairs. The system output is a speaking probability of each one of the actors together with the 3D location of the actors' faces.

exploit the fact that for some actions such as "talk phone" the auditory information is relevant for describing the action. Another multi-modal approach was followed in Lacheze *et al.* (2009), in which the auditory information was used to recognize objects that can be partially occluded or difficult to detect. Notice that, the visual information is also very helpful when the auditory data is strongly corrupted by noise or by multiple sound sources. Another example can be found in Itohara *et al.* (2011) where the authors combine information coming from the two modalities to perform beat tracking of a person playing the guitar. Auditory and visual information is combined together to better address the problems of beat-tracking, tempo changes, and varying note lengths. Also this different-modality combination is used for improvement of simultaneous speech signals in Nakadai *et al.* (2004). Using a pair of cameras faces are first detected and then in 3D. Using two microphones sound-source separation by ADPF (Active Direction Pass Filter) is applied. Finally these data is integrated at two levels, at the signal level and at the word level.

Among all possible applications using audio-visual data, we are interested in

detecting multiple speakers in informal scenarios. A typical example of such a scenario is shown in Figure 6.1, in which two people are sitting and chatting in front of the robot. The robot's primary task (prior to speech recognition, language understanding, and dialog) consists in retrieving the auditory status of several speakers along time. This allows the robot to concentrate its attention onto one of the speakers, *i.e.,* turn its head towards in the speaker's direction to optimize the emitter-to-receiver pathway, and attempt to extract the relevant auditory and visual data from the incoming signals. We note that this problem cannot be solved within the traditional human-computer interaction paradigm which is based on *tethered* interaction (the user must wear a close-range microphone) and which primarily works in the single-person-to-robot communication case. This considerably limits the range of potential interactions between robots and people engaged in a cooperative task or simply in a multi-party dialog. We investigate *untethered* interaction thus allowing a robot with its *on-board sensors* to perceive the status of several people at once and to communicate with them in the most natural way.

The original contribution of our approach is a complete real-time audio-visual speaker detection and localization system that is based binocular and binaural robot perception as well as on a generative probabilistic model able to fuse data gathered with camera and microphone pairs.

6.2 Related Work and Contributions

Audio-visual processing has been studied by many researchers. In Beal *et al.* (2002) the authors describe a speaker detection probabilistic graphical model fusing the information coming from one camera and two microphones. An EM algorithm estimates the model's parameters, i.e., the audio-visual appearance and the position of the speaker. In Fisher & Darrel (2004) the author proposes to use maximally informative projections to retrieve the main speaker. One camera per potential speaker and one microphone are used to gather the raw data, which

is projected in order to subsequently select the speaker, based on information-theoretic criteria. This approach is well suited for applications such as video-conferencing.

A second group of methods deal with interaction in smart-room environments. These methods assume the existence of several sensors distributed in the scene. For instance, Shivappa *et al.* (2010a) uses data acquired in a room equipped with a multi-camera system and an array of microphones. A tracking system is developed to complement information for a room with a smart interaction environment. The authors of Zhang *et al.* (2008) present an application aiming at making meetings more dynamic for people who are remotely connected. Based on one camera and one microphone array, this methodology is able to detect the speaking persons in real-time.

Because of on-line and on-board processing constraints associated with humanoid platforms, the computational load and complexity are constraints that need to be taken into account. Furthermore, the robot does not have a distributed sensor network, but merely a few sensors, which are all located in its head – *an agent-centered sensor architecture*. Hence, one should achieve a trade-off between performance and complexity.

In this chapter we present both a novel method and an original system approach to tackle the problem of on-line audio-visual detection of multiple speakers using the companion humanoid robot NAO. The proposed method uses data coming from a stereo pair of cameras and two microphones. Implemented on a hardware- and sensor-independent middleware, the software runs on-line with good performance. The 3D positions of the speakers' heads are obtained from the stereo image pair, and inter-aural time difference (ITD) values are extracted from the binaural signals. These features are then fused in a probabilistic manner in order to compute, over time, the probability of each person's speaking activity.

The approach exhibits a number of novelties with respect to previous work addressing audio-visual fusion for speaker detection: (i) visual features are obtained from a stereoscopic setup and thus represented in 3D, (ii) auditory features are obtained from only two microphones, while most of previous work uses an array

of microphones, (iii) the software is reusable with other robot sensor architectures, due to the flexibility of the underlying middleware layer, and (iv) good on-line performance in a complex environment, *e.g.,* echoic rooms, simultaneous auditory sources, background noise, uncontrolled lighting, cluttered scenes, etc.

6.3 An Audio-Visual Fusion Model

The overall goal is to retrieve the audio-visual (AV) state of the speakers in front of the robot. That is, the number of speakers as well as their positions and their speaking state. In order to reach this goal, we adopted the framework proposed in Alameda-Pineda *et al.* (2011). Based on a multi-modal Gaussian mixture model (mGMM), this method is able to detect and localize audio-visual events from auditory and visual observations. We chose this framework because it is able to account for several issues: (i) the observation-to-speaker assignment problem, (ii) observation noise and outliers, (iii) the possibility to weight the relevance of the two modalities, (iv) a generative formulation linking the audio and visual observation spaces, and (v) the possibility to deal with a varying number of speakers through a principled model selection method.

In a first stage, the low-level auditory and visual features are extracted. While the former correspond to the inter-aural time differences (ITDs), the latter correspond to interest points in image regions related to motion which are further reconstructed in the 3D space using a stereo algorithm. These 3D points will be referred to as the visual features.

The following direct sound propagation model:

$$\mathrm{ITD}(\boldsymbol{S}) = \frac{\|\boldsymbol{S} - \boldsymbol{M}_1\| - \|\boldsymbol{S} - \boldsymbol{M}_2\|}{\nu}, \tag{6.1}$$

is assumed. In this equation $\boldsymbol{S}$ corresponds to the sound source positions in the 3D space, *e.g.,* a speaker, $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ are the 3D coordinates of the microphones in some robot-centered frame, and ν denotes the sound speed. Equation 6.1 maps 3D points onto the 1D space of ITD observations. The key aspect of our generative audio-visual model Alameda-Pineda *et al.* (2011); Khalidov *et al.* (2011) is that

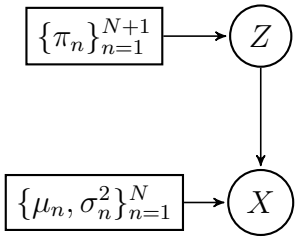


Figure 6.2: Graphical model generating the audio-visual observations. The hidden variable Z follows a multinomial distribution with parameters $\pi_1, \ldots, \pi_{N+1}$. The audio-visual observations X follow the law described by the probability density function in Equation 6.2.

Equation 6.1 can be used to map 3D points (visual features) onto the ITD space associated with two microphones, on the premise that the cameras are aligned with the microphones Khalidov *et al.* (2012). Hence the fusion between binaural observations and binocular observations is achieved in 1-D.

The underlying multi-modal GMM (mGMM) is a one-dimensional mixture of Gaussians. Each mixture component is associated with an *audio-visual object* centered at μ_n and with variance σ_n^2. This mixture has the following probability density function:

$$\text{prob}(x\,;\,\Theta) = \sum_{n=1}^{N} \pi_n\,\mathcal{N}(x\,;\,\mu_n, \sigma_n^2) + \pi_{N+1}\,\mathcal{U}(x), \tag{6.2}$$

where N is the number of components, *i.e.,* audio-visual objects, π_n is the weight of the n^{th} component, $\mathcal{N}(x\,;\,\mu_n, \sigma_n^2)$ is the value of the Gaussian distribution at x, $\mathcal{U}$ is the value of the uniform distribution accounting for outliers, and $\Theta = \{\pi_n, \mu_n, \sigma_n^2\}$. In this equation, x stands for a realization of the random variable X, shown in the corresponding graphical model on Figure 6.2, that could be either an auditory observation, *i.e.,* an ITD value or an observed 3D point, *i.e.,* a visual feature, mapped with Equation 6.1. Notice that both Θ and the hidden variable Z (modeling the observation-to-object assignments) need to be estimated. This is done using an Expectation-Maximization (EM) algorithm, derived from the probabilistic graphical model. Notice that with this formulation the number of AV objects N can be estimated from the observed data by maximizing a Bayesian

information criterion (BIC) score Alameda-Pineda *et al.* (2011). However, this implies to run the EM algorithm several times with different values of N, which is prohibitive in the case of an on-line implementation. From a practical point of view the problem of estimating N can be overcome by replacing the 3D visual points with *3D faces* as described below.

6.3.1 Visual Processing

The initial implementations of the nGMM EM algorithm was using 3D points Alameda-Pineda *et al.* (2011); Khalidov *et al.* (2011) as just described. Alternatively, one can replace 3D points with 3D faces, more precisely with 3D face centers which are fair approximations of 3D mouth positions, *i.e.,* the 3D acoustic emitters. In practice we start by detecting faces in images using Šochman & Matas (2005). Face centers are then detected in the left image of the stereo camera pair. For each left-image face center, the correspondent right coordinates for the same face center are obtained from the disparity map. This allows to reconstruct a 3D point, $\boldsymbol{S}_n$, that can be viewed as 3D face center. See Hansard & Horaud (2008) for more details. The use of faces drastically simplifies the complexity of the approach because a single semantically-meaningful face center replaces a cloud of points associated with a, possibly moving, 3D object. Initial means can be easily obtained from Equation 6.1, *i.e.,* $\mu_n = \text{ITD}(\boldsymbol{S}_n)$ while N, the number of AV objects can be easily estimated using the face detector Šochman & Matas (2005).

6.3.2 Auditory Processing

As already mentioned we use ITDs, *i.e.,* the time delay between the signals received at the left and right microphones. Notice that, due the symmetric nature of the ITD function, there is a front/back ambiguity, which is however slightly attenuated by the transfer function of the robot head. There are several methods to estimate ITDs (see Chan *et al.* (2006) for a review); We chose the cross-correlation method, since it optimizes a trade-off between performance and complexity. ITD

values are obtained in real-time by computing the cross-correlation function between the left and right perceived signals during an integration time window of length W, expressed in number of time samples, or frames. The time delay τ corresponding to the maximum of the cross-correlation function in the current integration window is computed as follows:

$$\tau = \frac{1}{F_s} \underset{d \in [-d_M, d_M]}{\operatorname{argmax}} \sum_{t=1}^{W} l(t) r(t+d) \tag{6.3}$$

where l and r are the left and right audio signals, F_s is the sampling frequency and d_M denotes the maximum possible delay between microphones, *i.e.,* $d_M = \|\boldsymbol{M}_1 - \boldsymbol{M}_2\|/\nu$. The time window W is a trade-off between reliability and significance. On one hand, a high W value implies more reliable ITD values, since the effect on the local maxima of the cross-correlation function is reduced. On the other hand, a small W value speeds up the computation. The parameter f denotes the shift of the sliding window used to compute the ITD. In order to extract one ITD value, two conditions need to be satisfied. First, there should be enough samples available within the integration window W. Second, the mean energy of the signals in the integration window should be higher than a given threshold E_A. In this way, we avoid to compute ITD values when the audio stream contains nothing but noise. Notice the method does not assume that the perceived sound signals are associated with some semantic *i.e.,* speech, pulse-resonance sounds, etc.

6.4 System Calibration

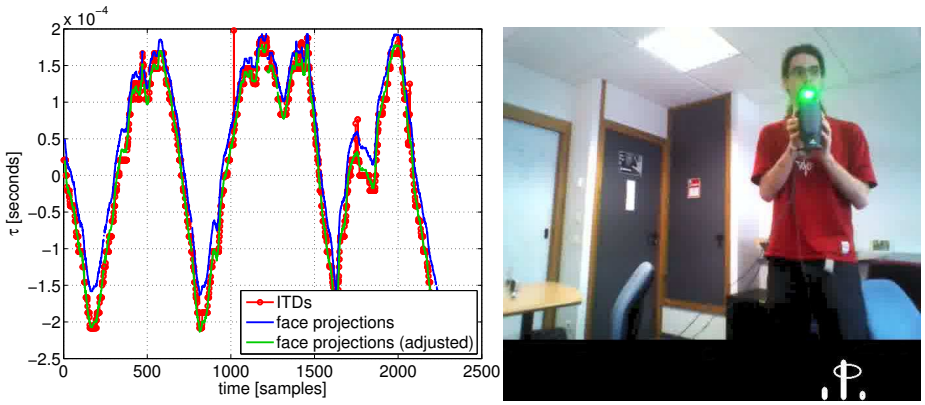
The audio-visual fusion model outlined above, and Equation 6.1 in particular, implies that the visual and auditory observations are computed in a common reference frame. This allows visual data to be *aligned* with auditory data. In practice it means that the cameras' extrinsic calibration parameters (position and orientation) and the microphones' positions are expressed in a common reference frame. Extrinsic camera calibration is performed using the state-of-the-art algorithm of Yves Bouguet (2010).

Audio-visual calibration can be achieved using Equation 6.1. A sound-source is placed in a known position $\boldsymbol{S}$ while $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ are unknown and hence must be estimated. The method of Khalidov *et al.* (2012) (i) uses an audio-visual target (a loud-speaker emitting white noise coupled with a small red-light bulb) to precisely position the sound source in the camera-pair reference frame, and (ii) estimates the unknown parameters $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ by considering several target positions and by solving a non-linear system of equations of the form of Equation 6.1.

This calibration procedure does not take into account the fact that the microphones are plugged into the robot head, as already mentioned above. To account for head effects we introduce two corrective parameters, α and β, to form of an affine transformation.

$$\mathrm{ITD}_{\mathrm{AD}}(\boldsymbol{S}) = \alpha \frac{\|\boldsymbol{S} - \boldsymbol{M}_1\| - \|\boldsymbol{S} - \boldsymbol{M}_2\|}{\nu} + \beta, \tag{6.4}$$

These parameters are estimated using the same audio-visual target mentioned above. The audio-visual target is freely moved in front of the robot thus following a zigzag-like trajectory. The use of white noise greatly facilitate the task of cross-correlation, *i.e.,* there is single sharp peak, and hence, makes the ITD computation extremely reliable. The reverberant components are suppressed by the direct component of the long lasting white noise signal. However, it is possible to set up the experimental conditions such as to reduce the effects of reverberation, *e.g.,* the room size is much larger than the target-to-robot distance. If the microphone positions are estimated in advance, the estimation of α and β can be carried out via a linear least-square estimator derived from Equation 6.4. Figure 6.3 shows the extracted ITDs (red-circle), mapped 3D face centers before the adjustment (blue), *i.e.,* using Equation 6.1 and after the adjustment (green), *i.e.,* using Equation 6.4. Clearly, the affine correction model allows a better alignment between the visual and auditory data.



(a) (b)

Figure 6.3: (a) The effect of the corrective parameters α and β onto audio-visual calibration. ITD values estimated as peaks of the auto-correlation function are shown with red circles. 3D face centers are mapped onto the ITD space without the corrective parameters (shown in blue) and once these parameters have been estimated (shown in green). (b) The speaker is emitting white noise, see ITDs in bottom part of the figure. The bright point is detected and mapped to ITD space (round circle in bottom). To calibrate the person moves the speaker in up and down from right to left.

6.5 Experimental Validation

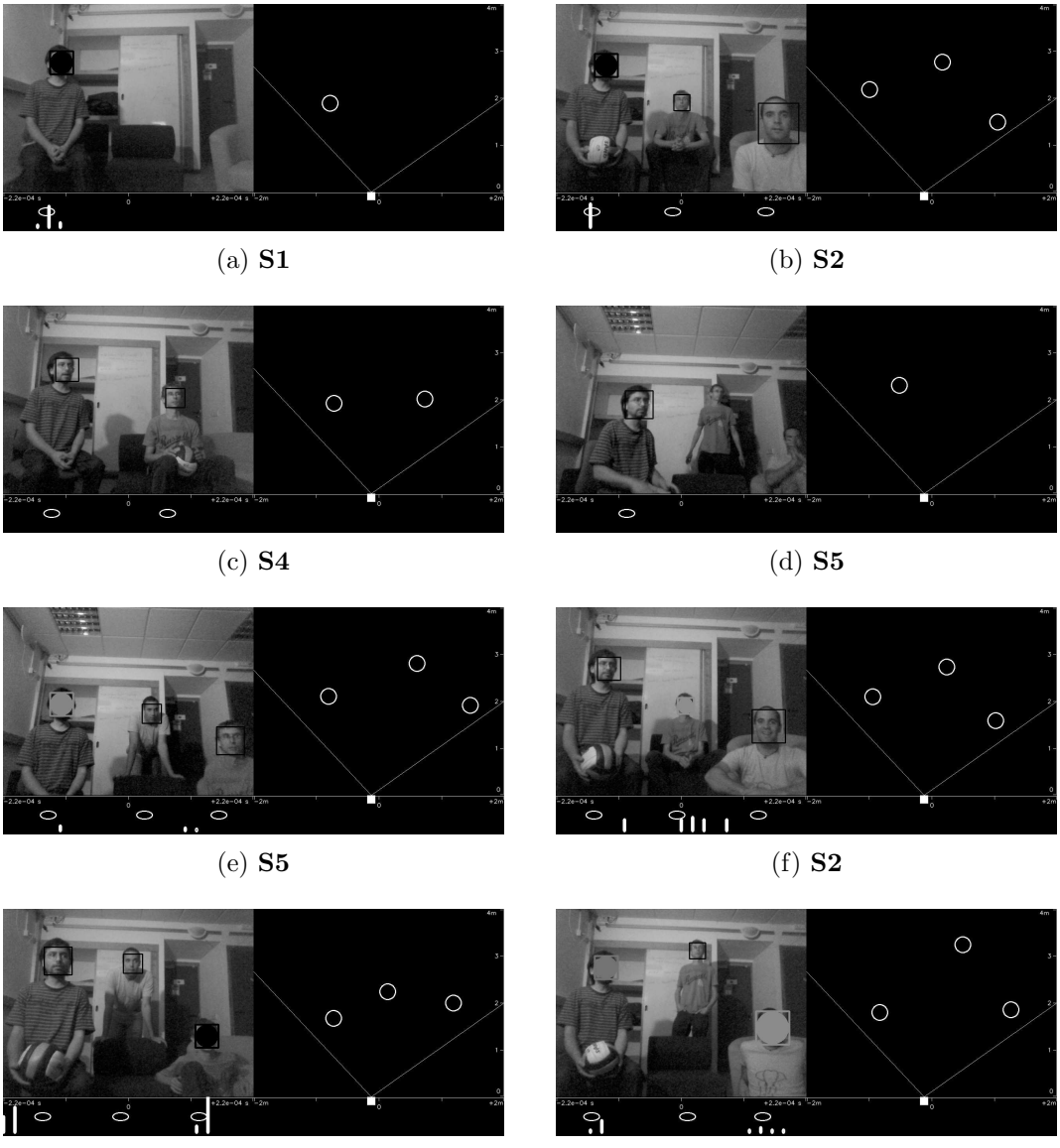
To validate our algorithm we performed a set of experiments with five different scenarios. The scenarios were recorded in a room around 5×5 meters, and designed to test the algorithm in different conditions in order to identify the limitations of the proposed approach. Each scenario is composed by several sequences in which people count from one up to sixteen. Except for the first scenario, composed by one sequence due to its simplicity, the rest of scenarios were recorded several times. Moreover, a video is recorded to show the different scenarios and have a visual validation of the results.

In scenario **S1**, only one person is in the room sitting in front of the robot and counting. In the rest of the scenarios (**S2**-**S5**) three persons are in the room. People are not always in the field of view (FoV) of the cameras and sometimes they move. In scenario **S2** three persons are sitting and counting alternatively

one after the other. The configuration of scenario **S3** is similar to the one of **S2**, but one person is standing instead of sitting. These two scenarios are useful to determine the precision of the ITDs and experimentally see if the difference of height (elevation) affects the quality of the extracted ITDs. The scenario **S4** is different from **S2** and **S3** because one of the actors is outside the FoV. This scenario is used to test if people speaking outside the FoV affect the performance of the algorithm. In the last scenario (**S5**) the three people are in the FoV, but they count and speak independently of the other actors. Furthermore, one of them is moving while speaking. With **S5**, we aim to test the robustness of the method to dynamic scenes.

In Fig. 6.4 we show several snapshots of our visualization tool. These frames are selected from the different scenarios aiming to show both the successes and the failures of the proposed system. Fig. 6.4a shows an example of perfect alignment between the ITDs and the mapped face, leading to a high speaking probability. A similar situation is presented in Fig. 6.4b, in which among the three people, only one speaks. A failure of the ITD extractor is shown in Fig. 6.4c, where the actor in the left is speaking, but no ITDs are extracted. In Fig. 6.4d we can see how the face detector does not work correctly: one face is missing because the actor is too far away and the other's face is partially occluded. Fig. 6.4e shows a snapshot of an AV-fusion failure, in which the extracted ITDs are not significant enough to set a high speaking probability. The Fig. 6.4f, Fig. 6.4g and Fig. 6.4h show the effect of reverberations. While in Fig. 6.4h we see that the reverberations lead to the wrong conclusion that the actor on the right is speaking, we also see that the statistical framework is able to handle reverberations (Fig. 6.4f and Fig. 6.4g), hence demonstrating the robustness of the proposed approach.

The scenarios are manually annotated such that we get the ground truth. In order to systematically evaluate the proposed system we adopted an overlap-based strategy. The ground truth of actor n is split in speaking intervals I_n^k indexed by k and silent intervals J_n^l indexed by l. For clarity purposes let us denote by $p_n(t)$ the detected speaking state of actor n at time t. For each of the speaking intervals I_n^k we compute $c_n^k = \sum_{t \in I_n^k} p_n(t)/|I_n^k|$. If $c_n^k \geq 0.5$ we count one correct detection, otherwise we count one false negative. We also compute



(a) **S1** (b) **S2**

(c) **S4** (d) **S5**

(e) **S5** (f) **S2**

(g) **S3** (h) **S3**

Figure 6.4: Snapshots of the visualization tool. Frames are selected among the five scenarios such as to show both the method's strengths and weaknesses. (a) Good results on **S1**. (b) Good results on **S2**, three people. (c) The ITD extractor does not work correctly, thus missing the speaker. (d) Misses of the face detection module. (e) The audio-visual fusion fails to set a high probability to the current speaker. (f,g) The audio-visual fusion model is able to handle reverberations. (h) The reverberations are too close to the mapped head, leading to a wrong decision.

	CD	FP	FN	Total
S1	14	0	0	14
S2	76	12	3	79
S3	75	19	0	75
S4	60	13	2	62
S5	26	20	0	26

Table 6.1: Quantitative evaluation of the proposed approach for the five scenarios. The columns represent, in order: the amount of correct detections (CD), the amount of false positives (FP), the amount of false negatives (FN) and the total number of counts (Total).

$\tilde{c}_n^k = \sum_{t \in J_n^l} p_n(t)/|J_n^l|$. In case $\tilde{c}_n^l \geq 0.5$ we count one false positive. In summary, if the speaker is detected during more than half of the speaking time, we count on correct detection (CD), otherwise a false negative (FN). And if it is detected more than half of the speaking time, we count a false positive (FP).

Table 6.1 shows the results obtained with this evaluation strategy on the presented scenarios. First of all we notice the small amount of false negatives: the system misses very few speakers. A part from the first scenario (easy conditions), we observe some false positives. These false positives are due to reverberations. Indeed, we notice how the percentage of FP is severe in **S5**. This is due to the fact that high reverberant sounds (like hand claps) are also present in the audio stream of this scenario. We believe that an ITD extraction method more robust to reverberations will lead to more reliable ITD values, which in turn will lead to a better speaker detector. It is also worth to notice that actors in different elevations and non-visible actors do not affect the performance of the proposed system, since the results obtained in scenarios **S2** to **S3** are comparable.

6.6 Discussion

We presented a system targeting speaker detection working on the humanoid robot NAO in regular indoor environments. Implemented on top of a platform-

independent middleware, the system processes the audio-visual data flow from two microphones and two cameras at a rate of 17 Hz. We proposed a statistical model which captures outliers from the perception processes. The method runs in normal echoic rooms with just two microphones mounted inside the head of a companion robot with noisy fans. We demonstrated good performance on different indoor scenarios involving several actors, moving actors and non-visible actors. This works contributes to a better understanding of the audio-visual scene using a robo-centric set of sensors mounted in an autonomous platform, such as NAO, under the constraints of an on-line application.

It is worth noticing that the module limiting the performance of the system is the ITD extraction, due to the room reverberations. We will work on making this module more robust to this kind of interferences. Moreover, audio-visual tracking capabilities are also a desirable property for any robot, since they provide for temporal coherence of the scene. In a more developed stage, it would be desirable that NAO is able to choose regions of interest, so that it could perform active learning, and enhance its audio-visual skills.

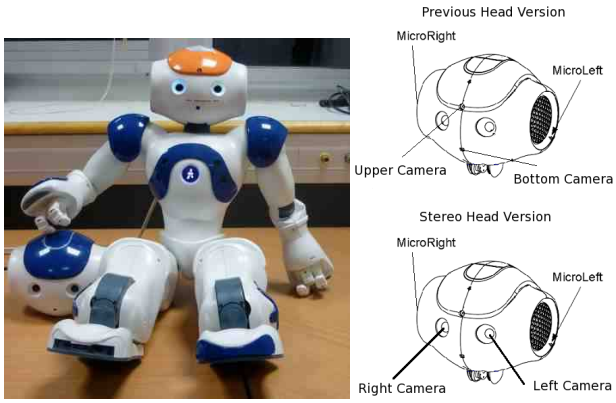
Chapter 7

Implementing the Algorithms to NAO

7.1 New NAO stereo head

As mentioned in the Introduction, the selected robotic platform is the humanoid robot NAO. It is important to notice that the version of NAO used has different specifications. In concrete, it has a stereo head. As shown in Figure 7.1, the new head has different distribution of the two VGA cameras, instead of top and bottom position where there was no overlap, the new positions are in left and right eyes, with enough overlap to allow to run stereo algorithms. The new head is also equipped with four microphones distributed in front, rear, left and right positions of the head. While front, left and right microphones have acceptable SNR, the rear microphone, due is located very close to the microprocessor fan, is very noisy. The rest of characteristics remains the same, even though are not used for this thesis.

To interface with these hardware a middleware, explained in Section 7.2, is used. This is important to have code that is platform independent and it has the potential to run in other humanoid robots.



(a) (b)

Figure 7.1: NAO robot with the new and old heads. Notice that position of cameras have changed from top-bottom in the old head, to left-right in the new head allowing to have overlap between images which it was not the case. New head has 4 microphones, even here we only name 2.

7.2 RSB middleware

The distributed components of our system are integrated using the *Robotics Service Bus* (RSB) middleware Wienke & Wrede (2011). RSB is a platform-independent event-driven middleware specifically designed for the needs of distributed robotic applications. It is based on a logically unified bus which can span over several transport mechanisms like network or in-process communication. The bus is hierarchically structured using scopes on which events can be published with a common root scope. Through the unified bus, full introspection of the event flow between all components is easily possible. Consequently, several tools exist which can record the event flow and replay it later, so that application development can largely be done without a running robot. RSB events are automatically equipped with several timestamps, which provide for introspection and synchronization abilities. Because of these reasons RSB was chosen instead of NAO's native framework NAOqi and we could implement and test our algorithms remotely without performance and deployment restrictions imposed by the robot platform. Moreover, the resulting implementation can be reused for other robots.

From a client program's perspective, communication over RSB is based on asynchronous event notifications. Clients need to install handlers which are invoked immediately once a new event is received from the bus. Based on the asynchronous notifications, synchronous remote procedure calls (RPC) are implemented. Language implementations of RSB exist for C++, Java, Python and Common Lisp. The usage of RSB results in a lose coupling between different modules of the architecture and the introspection support facilitates the development process, and export to other robotic platforms.

One tool available in the RSB ecosystem is an event synchronizer, which synchronizes events based on the attached timestamps with the aim to free application developers from such a generic task. However, several possibilities of how to synchronize events exist and need to be chosen based on the intended application scenario. For this reason, the synchronizer implements several strategies, each of them synchronizing events from several scopes into a resulting compound event containing a set of events from the original scopes. We used two strategies for the implementation. The *ApproximateTime* strategy is based on the algorithm available in ROS and outputs sets of events containing exactly one event from each scope. The algorithm tries to minimize the time between the earliest and the latest event in each set and hence well-suited to synchronize events which originate from the same source (in the world) but suffered from perception or processing delays in a way that they have non-equal timestamps. The second algorithm, *TimeFrame*, declares one scope as the primary event source and for each event received here, all events received on other scopes are attached that lie in a specific time frame around the time stamp of the source event.

Also, a record-replay solution is available, which was used to record the event stream of the running robot, particularly containing the audio buffers and vision frames. The recorded events could be replayed transparently for the remaining software modules. Hence, development of processing modules could be performed without the robot, which speeds up the testing cycle.

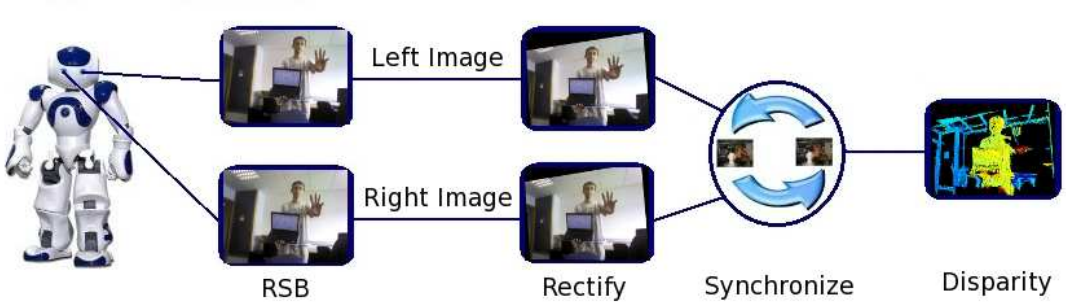


Figure 7.2: Modular decomposition of the stereo algorithm. Left and right images are sent to respective rectification modules. The rectified images are synchronized and stereo module computes the disparity map corresponding to the synchronized and rectified images.

7.3 Stereo GCS implementation

Even though, the algorithms presented in Chapter 2 and Chapter 3 have the potential to run in real time (20fps) in a standard computer of 2010, the algorithm ported to NAO is just the seed growing algorithm to compute disparity maps. The version used is optimized and implemented in C++ by Dobiaš & Šára (2011).

The algorithm is interfaced with the RSB middleware, and a scheme of the modules involved in the implementation is depicted in Figure 7.2. What the algorithm expects is two rectified and synchronized images. The rectification is done by a separate module *Image rectification* which produces an affine transform to the image using the homography matrix provided as a parameter. To ensure that the images are synchronized a *ApproximateTime* strategy is used, and finally this is send to the module *Stereo GCS* that computes the disparity map.

The homography matrices for rectification are derived from the fundamental matrix. At the same time, the fundamental matrix is derived from the calibration of the left and right cameras. The algorithm has also two parameters to tune it. One is the disparity range, which in our particular case is set to -100 and +100, but can vary for each robot. The other parameter is a threshold, set to 0.8 which

indicates that all correlations that are below this value are not considered and consequently pixel is left unmatched.

7.4 Audio-Visual Action Recognition implementation

The algorithm presented in Chapter 5 is implemented in a modular fashion through the RSB middleware. A schema of the modules involved in the implementation is shown in Figure 7.3. The application has several modules, first the visual and auditory modules, *Visual Descriptor* and *Auditory Descriptor*, computes visual and auditory features respectively. The visual descriptor needs a disparity map, which its structure has just been commented in previous subsection. The face detection is a module that is already given to us and compute the position of the faces in a image, which is necessary to define the reference point of the descriptor explained in Chapter 4. The auditory descriptor computes MFCC coefficients of a signal corresponding to the left microphone. Both, visual and auditory, descriptor modules are controlled by the *Bound Command Detection* module. This module is in charge of defining the beginning and ending of an action, which is needed since the action recognition performed is not continuous. To this end, the left image is used. When is detected enough (decided by a threshold) amount of motion in the image, it triggers the descriptors that start to accumulate the histograms. After a few seconds, the action is considered finished, and the histograms sent to the synchronization module. The synchronization strategy used is *ApproximateTime*. Finally, with the visual and auditory histograms the *Categorization* module can decide which action it was performed. The *Categorization* module implements a linear SVM and has to be trained offline.

Visual descriptor uses parameters commented in previous section for stereo. Auditory descriptor uses default parameters of libmfcc library and the threshold for bound command detection is set to 0.4. This means that at least exist movement of at least 40 per cent of the image.

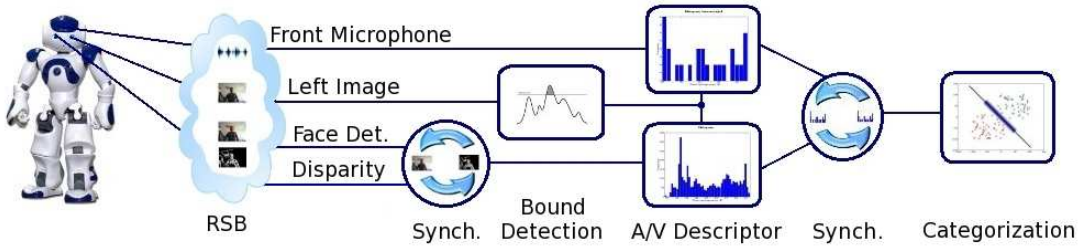


Figure 7.3: Modular decomposition of the audio-visual command recognition algorithm. The left image is used to detect the action boundaries. During a command, visual and auditory modules extract and accumulate the descriptors (histograms). When the command is over, both histograms are synchronized and sent to the categorization module which will decide which command was performed.

7.5 Audio-Visual Speaker Detection implementation

The algorithm presented in Chapter 6 is divided into four components which are described in the pipeline shown in Figure 7.4.

In detail, the visual part has five different modules. *Left video* and *Disparity image* stream the images received from left and disparity images. The *Left face detection* module extracts the faces from the left image. These are then synchronized with the right image in *Face-image Synchronization*, using the *ApproximateTime* strategy. The *3D Faces* module computes the 3D head (or face) centers.

The auditory component consists of three modules. Interleaved audio samples coming from the two microphones are streamed by the *Audio* module. These are de-interleaved by *Sound formatting* and stored into two circular buffers; for the left and right microphone's signals. Finally, the module called *ITD extraction* is in charge of compute the ITD values.

Both visual and auditory features flow until the *Audio-visual synchronization* module; the *TimeFrame* strategy is used here to find the ITD values coming

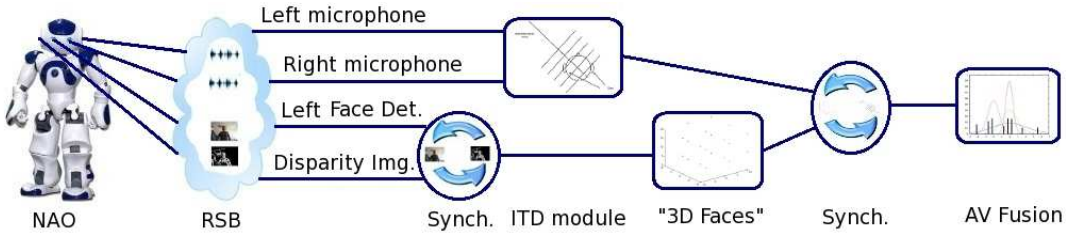


Figure 7.4: Modular structure of the audio-visual fusion algorithm. Data coming from left and right microphones is processed and used to compute the ITDs. Left face detection and the disparity map are synchronized and used to extract 3D position of each detected face by the *3D Faces* module. Finally, to proceed to the fusion, these faces are synchronized with previously computed ITDs. A last module for visualization is explained later, see Fig. 7.5.

from the audio pipeline associated to the 3D head positions coming from the visual processing. These synchronized events feed the *Audio-visual fusion* module, which is in charge of estimating the emitting sound probabilities p_n.

Several considerations need to be done regarding the details of the algorithm implementation in order to guarantee the repeatability of the experiments. When computing the ITD values, a few parameters need to be set. There is a trade off when setting the integration window W and the frame shift f. A good compromise between low computational load, high rate, and reliability of ITD values was found for $W = 150$ ms and $f = 20$ ms. Finally, we set the activity threshold to $E_A = 0.001$. Notice that this parameter could be controlled by a higher level module which would learn the characteristics of the scene and infer the level of background noise. When computing the probabilities p_n, the variances of the mGMM are set to $\sigma_n^2 = 10^{-9}$, we found this value big enough to take into account the noise in the ITD values and small enough to discriminate speakers that are close to each other.

ApproximateTime is used in our case to synchronize the results from the left and right camera as frames in general form matching entities but due to independent grabbing of both cameras have slightly different time stamps. Results from the stereo matching process are synchronized with ITD values using the

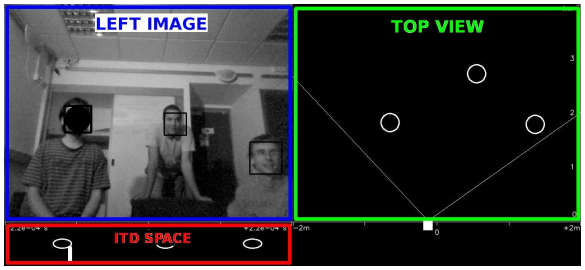


Figure 7.5: Snapshot of the visualization tool. Top-left (blue-framed): The original left image overlaid with one bounding box per detected face. In addition, an intensity-coded circle appears when the face emits a sound. The darker the circle, the higher the speaking probability. Top-right (green-framed): A bird-view of the 3D scene, in which each circle corresponds to a detected head. Bottom-left (red-framed): The ITD space. The projected faces are displayed with an ellipse while extracted ITD values are shown as bars in a histogram.

TimeFrame strategy because the integration time for generating ITD values is much smaller than for a vision frame and hence multiple ITD values belong to a single vision result.

Finally, we developed the module *Visualization*, in order to get a better insight of the proposed algorithm. A snapshot of this visualization tool can be seen in Figure 7.5. The visualization plot consists of three parts. The top-left part displays a bounding box around each detected face overlaid onto the original left image. In addition to the bounding boxes, a solid circle is plot on a face whenever it emits a sound. The intensity encodes the emitting sound probability, the higher it is, the darker the circle. The top-right part, framed in green, is a bird-view of the 3D reconstructed scene, in which the detected 3D faces are shown with circles. The bottom-left part, with a red frame, represents the ITD space in which both the mapped face centers (ellipses) and the histogram of ITD values are plot.

Chapter 8

Conclusions

The work presented in this manuscript has been done in the framework of the Humavips European project, which its goal is to provide humanoid robot NAO with social skills. This means that the robot can interact up to some extend with humans. Each of the partners of the project played a role towards this objective. Here has only been described a part focused in the perception capabilities which contributions and future work will be presented below.

8.1 Main Contributions

The contributions in this thesis are two fold. From one side there are the scientific contributions which are in form of new algorithms and on the other side there are the software contributions which are some implementations for humanoid robot NAO of some of these algorithms.

First contribution is in the visual domain developing two different stereo algorithms based on seed growing. One using temporal information and the other growing jointly disparity and optical flow. An existing implementation C++ version of only disparity estimation, which runs at 20fps in VGA images, is interfaced with RSB middle-ware to be used in NAO.

Several approaches had attempted to use temporal information in stereo due temporal information can be useful in texture-less regions and moreover can help

to disambiguate matching pixels. However, this assumption valid for static scenes is not so clear when scene is dynamic and disparity is not constant in time. This factor adds a problem since spatio-temporal cubes used for matching will look different in both cameras, thus making it difficult to match. Most of the methods proposed in the past are only able to cope with really small disparity changes in time. Our contribution is to propose a method that is able to deal with big disparity changes in time by detecting when a change in disparity is produced and hence disconnecting the spatio-temporal matching for that particular pixel. In the worst case we have the same performance as no using temporal information, but we clearly get improvements in the other cases.

In case of sceneflow, existing algorithms solve the problem looking for disparity and optical flow separately. Methods based on global optimizations are slow and highly dependent of a good initialization, otherwise unable to obtain a reasonable map. In contraposition, the method used has no need for initialization, is not optimizing any complex function and is fast due the search space is highly reduced. The main contribution is to propose a joint disparity and optical flow search since both are constrained by the epipolar geometry and can be nicely integrated in a seed growing algorithm.

A second contribution towards human robot interaction is an action recognition algorithm. First by using only visual information and then adding auditory information using a convex function to fuse data from both modalities is developed. A version of this audio-visual action recognition is also implemented into NAO.

Scene flow information is very valuable since provides depth information and 3D velocity vectors for each pixel. The contribution is a descriptor inspired in the widely used HOG/HOF based on this information. It is quite important to detect "meaningful" pixels in scene, and these pixels generally coincides with moving objects of the scene. The advantage of having scene flow information is that segmentation to discriminate the "meaningful" pixels from the others is not only based on moving information, which is the case of 2D, but also can be done by depth information. Often 2D methods have difficulties to distinguish

moving objects corresponding to the actor (foreground) from the rest of the scene (background) and it is here when scene flow information plays an important role.

Only visual information is not always the optimal situation, due some occlusions can happen and affect to the final recognition. Adding auditory information can complement and cover for such problems. Conversely, auditory information can be noisy and for this cases visual information can be of a great help. Our contribution here is to use a fusion method to combine both modalities and provide an online implementation for NAO. Audio-visual information has been used in several applications however, is never been implemented in a humanoid robot. Moreover, generally the implementations on robotic platforms consist on a robotic system from one side and the audio-visual system on the other side as separate systems but not integrated. To overcome this problem we establish a simple protocol to detect the boundaries of an action in order that the algorithm can be used online in the robot.

The last contribution is also in the domain of audio-visual fusion, but in this case for speaker detection. A version for this algorithm is implemented into NAO.

Different speaker detection applications exist. Some use an array of microphones, some use only one camera, etc. Our contribution is an audio-visual fusion method that detect visible speaking persons using information from both cameras of NAO and only two microphones. With the cameras a 3D position of where the person is localized is used together with auditory information, which by means of ITDs providing also a localization, for a Gaussian mixture model that give us the probability of a visible person is speaking.

8.2 Future Work

Despite the contributions towards a better perception capabilities in humanoid robot NAO, there is still some improvement to do in the presented work, as well as, new possible directions of work that are presented below.

As seen in Chapter 2, temporal stereo is very complex problem, since movement is seen differently in each camera. Algorithms based on minimization meth-

ods that tries to reshape the spatio-temporal cubes, only work for a small movements. The solution proposed is based on an observations of only few frames correlation to determine whether is used the temporal matching or not. This decision could be improved using learning techniques based on the history of observations of the correlation to take a better decision. A learning technique presented in Zhang *et al.* (2011b), could be applied in this case.

The algorithm proposed in Chapter 3, has similar properties as the one presented in the previous chapter, and it could be improved in several ways. Learning is also an option. When used the predictor to decide the seeds for the next frame, more complex learning function could be learnt, in this case a suitable method such as Hadfield & Bowden (2011), could be applied to track and predict future seeds. The way that seeds are obtained it can also be improved. While now seeds for the optical flow are searched independently in left and right image, it can be interesting to enforce the epipolar constrain for this search and hence start from better seeds.

A part from the improvements of the methodology, both algorithms can be ameliorated by introducing sub-pixel accuracy in order to have more continuous disparity and optical flow values and densifying the output given. However, these changes are not crucial. Usually one pixel disparity is enough for most of possible applications, when the semi-dense output corresponds to unmatched pixels that at same time are occlusions or texture-less parts of the image. Finally, using the same growing methodology new applications can be derived, such as Zhang *et al.* (2011a), which obtains a segmentation map at the same time as stereo map.

The descriptor presented in Chapter 4 has the problem of the dependency of the face detector. A reference point is needed to express everything in relative coordinates, to make the descriptor scale and position invariant. However, the presence of a reference point can be problematic when the track of this reference is lost due occlusions or others. This is more evident in case of frontal face detection (our case), when track is lost just turning the head. To overcome this problem, the first it could be done is to use a head detector or upper-body detector in order to be less sensible to changes of face orientations and rotations. Another

option is to use a different descriptor not dependent of a reference point as the ones described in Fehr *et al.* (2012) or Bo *et al.* (2010).

Audio-visual fusion, presented in Chapter 5, is very useful to increase the recognition rate on any action recognition framework. The complementary nature of both modalities is important to for example when in vision there is an occlusion the auditory information can help. Conversely when a sound is corrupted, vision can help. The fusion algorithm proposed is based on a convex function where the optimal value is found sweeping the whole space from zero to one to find the optimal combination of the data from both modalities. However, the ways to fuse data from different modalities can be done at different levels. For example, the fusion could be done at feature level or decision level, etc. This could be some future work to explore, a survey on this is Shivappa *et al.* (2010b). Moreover, the classifier used is a linear SVM, but other improvement could be to use multiple kernel learning (MKL) to learn both kernel and optimal parameter for fusion data from different modalities at the same time.

The action recognition presented works as an isolated framework. This means that is necessary to define in advance the beginning and the end of the action, in order to perform the classification. Other future direction could be to further investigate in continuous recognition as in Huang *et al.* (2012), which will allow to get rid of the necessity of boundaries for an action.

In Chapter 6, an audiovisual method to determine visible speaking persons is presented. This method uses "3D faces" as visual features and ITDs as auditory features that feeds a fusion algorithm using GMMs. The feature work is several fold. From one side, as same as in Chapter 4, "3D faces" depends on a frontal face detector. This could be changed with a more robust detector. However most of the problems in this application are given by the audio sensors. Auditory features could be improved using more cues as inter-aural level differences (ILDs) which works well for high frequencies and use ITDs for low frequencies and mitigate the reverberation problems, which causes false positive detections. In Li *et al.* (2012) is addressed exactly the same problem and also some ideas could be used to our particular method. Another interesting option would be to add tracking to the application as in Shivappa *et al.* (2010a).

Appendix A

RAVEL Dataset

A.1 Data Set Description

The RAVEL data set has three different categories of scenarios. The first one is devoted to study the recognition of actions performed by a human being. With the second category we aim to study the audio-visual recognition of gestures addressed to the robot. Finally, the third category consists of several scenarios; they are examples of human-human interaction and human-robot interaction. Table A.1 summarizes the amount of trials and actors per scenario as well as the size of the visual and auditory data. Figure A.1 (a)-(h) shows a snapshot of the different scenarios in the RAVEL data set. The categories of scenarios are described in detail in the following subsections.

A.1.1 Action Recognition [AR]

The task of recognizing human-solo actions is the motivation behind this category; it consists of only one scenario. Twelve actors perform a set of nine actions alone and in front of the robot. There are eight male actors and four female actors. Each actor repeats the set of actions six times in different – random – order, which was prompted in two screens to guide the actor. This provides for various co-articulation effects between subsequent actions. The following is a detailed list

Table A.1: Summary of the recorded data size per scenario.

Scenario	Trials	Actors	Video in MB	Audio in MB
AR	12	12	4,899	2,317
RG	11	11	4,825	1,898
AD	6	6	222	173
C	5	4	118	152
CPP	1	1	440	200
MS	7	6	319	361
IP	5	7	327	204
Total	–	–	11,141	5,305

(a) *Talk on the phone* (b) *Stop!* (c) *Where is the kitchen?* (d) *Cheers!*

(e) *Cocktail party* (f) *Hand-shaking* (g) *Let me introduce you!* (h) *Someone arrives*

Figure A.1: Scenario examples from the RAVEL data set. (a) Human activity – *talk on the phone*–, (b) Robot command – *stop!*–, (c) Asking the robot for instructions, (d) Human-human interaction, (e) Cocktail party, (f) Human introducing a new person (g) Robot introducing a new person, and (h) New person.

of the set of actions: (i) *stand still*, (ii) *walk*, (iii) *turn around*, (iv) *clap*, (v) *talk on the phone*, (vi) *drink*, (vii) *check watch* (analogy in Weinland *et al.* (2006)), (viii) *scratch head* (analogy in Weinland *et al.* (2006)) and (ix) *cross arms* (analogy in Weinland *et al.* (2006)).

A.1.2 Robot Gestures [RG]

Learning to identify different gestures addressed to the robot is another challenge in HRI. Examples of such gestures are: waving, pointing, approaching the robot, etc. This category consists of one scenario in which the actor performs six times the following set of nine gestures: (i) *wave*, (ii) *walk towards the robot*, (iii) *walk away from the robot*, (iv) *gesture for 'stop'*, (v) *gesture to 'turn around'*, (vi) *gesture for 'come here'*, (vii) *point action*, (viii) *head motion for 'yes'* and (ix) *head motion for 'no'*. In all cases, the action is accompanied by some speech corresponding to the gesture. In total, eleven actors (nine male and two female) participated in the recordings. Different English accents are present in the audio tracks which makes the speech processing challenging.

A.1.3 Interaction

This category contains the most interactive part of the data set, i.e. human-human as well as human-robot interaction. Each scenario consists of a natural scene in which several human beings interact with each other and with the robot. In some cases one of the actors and/or the robot act as a passive observer. This category contains six different scenarios detailed in the following. In all cases, a person emulated the robot's behavior.

Asking for Directions [AD]

In this scenario an actor asks the robot for directions to the toilets. The robot recognizes the question, performs gender identification and gives the actor the right directions to the appropriate toilets. Six different trials (four male and two female) were performed. The transcript of this scenario is in Script 1.

Actor	(enters the scene)
Actor	Excuse me, where are the toilets?
Robot	Gentleman/Ladies are to the left/right and straight on 10 meters.
Actor	(leaves the scene)

Script 1: The script encloses the text spoken by the actor as well as by the robot in the "*Asking for directions*" scenario.

Chatting [C]

We designed this scenario to study the robot as a passive observer in a dialog. The scenario consists of two people coming into the scene and chatting for some undetermined time, before leaving. There is no fixed script – occasionally two actors speak simultaneously – and the sequences contain several actions, e.g. hand shaking, cheering, etc. Five different trials were recorded.

Cocktail Party Problem [CPP]

Reviewed in Haykin & Chen (2005), the Cocktail Party Problem has been matter of study for more than fifty years (see Cherry (1953)). In this scenario we simulated the cocktail party effect: five actors freely interact with each other, move around, appear/disappear from the camera field of view, occlude each other and speak. There is also background music and outdoor noise. In summary, this is one of the most challenging scenarios in terms of audio-visual scene analysis, action recognition, speech recognition, dialog engaging and annotation. In the second half of the sequence the robot performs some movements. Figure A.2 is a frame of the (left camera of the) CPP scenario. Notice the complexity of the scene in terms of number of people involved, dialog engagement, etc.

Where Is Mr. Smith? [MS]

The scenario was designed to test skills such as face recognition, speech recognition and continuous dialog. An actor comes into the scene and asks for Mr.

Figure A.2: A frame of the CPP sequence representative of the complexity of this scenario.

Actor	(enters and positions him in front of the robot)
Actor	I am looking for Mr. Smith?
Robot	Yes Sir, Mr. Smith is in Room No. 22
Actor	(leaves the scene)
Mr. Smith	(enters the scene)
Mr. Smith	Hello Robot.
Robot	Hello Mr. Smith.
Robot	How can I help you?
Mr. Smith	Haven't you seen somebody looking for me?
Robot	Yes, there was a gentleman looking for you 10 minutes ago.
Mr. Smith	Thank you Bye.
Robot	You are welcome.
Mr. Smith	(leaves the scene)

Script 2: Detail of the text spoken by both actors (Actor and Mr. Smith) as well as the Robot in the "*Where is Mr. Smith?*" scenario.

Smith. The robot forwards the actor to Mr. Smith's office. However, he is not there and when he arrives, he asks the robot if someone was looking for him. The robot replies according to what happened. The transcript for the scenario is in Script 2. Seven trials (five male and two female) were recorded to provide for gender variability.

Introducing People [IP]

This scenario involves a robot interacting with three people in the scene. There are two versions of this scenario: passive and active. In the passive version the camera is static, while in the active version the camera is moving to look directly at speakers' face. Together with the *Cocktail Party Problem* scenario, they are the only exception where the robot is not static in this data set.

In the passive version of the scenario, Actor 1 and Actor 2 interact together with the Robot and each other; Actor 3: only interacts with Actor 1 and Actor 2.

The transcript of the passive version is in Script 3. In the active version, Actor 1 and Actor 2 interact with the Robot and each other; Actor 3 enters and leaves room, walking somewhere behind Actor 1 and Actor 2, not looking at the Robot. The transcript of the active version is detailed in Script 4

A.1.4 Background Clutter

Since the RAVEL data set aims to be useful for benchmarking methods working in populated spaces, the first two categories of the data set, action recognition and robot gestures, were collected with two levels of background clutter. The first level corresponds to a controlled scenario in which there are no other actors in the scene and the outdoor and indoor acoustic noise is very limited. During the recording of the scenarios under the second level of background clutter, other actors were allowed to walk around, always behind the main actor. In addition, the extra actors occasionally talked to each other; the amount of outdoor noise was not limited in this case.

A.1.5 Data Download

The RAVEL data set is publicly available at http://ravel.humavips.eu/ where a general description of the acquisition setup, of the data, and of the scenarios can be found. In addition to the links to the data files, we provide previews for all the recorded sequences for easy browsing previous to data downloading.

A.2 Acquisition Setup

Since the purpose of the RAVEL data set is to provide data for benchmarking methods and techniques for solving HRI challenges, two requirements have to be addressed by the setup: a robocentric collection of accurate data and a realistic recording environment. In this section the details of this setup are given, showing that the two requisites are satisfied to a large extent. In a first stage the recording device is described. Afterward, the acquisition environment is delineated. Finally,

Figure A.3: Two views of the recording environment. The POPEYE robot is in one side of the room. As shown, the sequences were shot with and without daylight providing for lighting variations. Whilst two diffuse lights were included in the setup to provide for good illumination, no devices were used to modify neither the illumination changes nor the sound characteristics of the room. Hence, the recordings are affected by all kind of audio and visual interferences and artifacts present in natural indoor scenes.

the properties of the acquired data in terms of quality, synchrony and calibration are detailed and discussed.

The POPEYE robot was designed in the framework of the POP project[1]. This robot is equipped with four microphones and two cameras providing for auditory and visual sensory faculties. The four microphones are mounted on a dummy-head, as shown in Figure A.4, designed to imitate the filtering properties associated with a real human head. Both cameras and the dummy head are mounted on a four-motor structure that provides for accurate moving capabilities: pan motion, tilt motion and camera vergence.

The POPEYE robot has several remarkable properties. First of all, since the device is alike the human being, it is possible to carry out psycho-physical studies using the data acquired with this device. Secondly, the use of the dummy head and the four microphones, allows for the comparison between using two microphones and the Head Related Transfer Function (HRTF) against using four microphones without HRTF. Also, the stability and accuracy of the motors ensure

[1] http://perception.inrialpes.fr/POP/

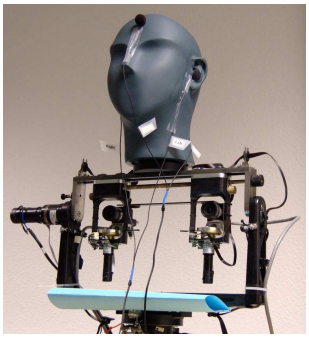


Figure A.4: The POPEYE robot head was used to collect the RAVEL data set. The color-camera pair as well as two (front and left) out of four microphones are shown in the image. Four motors provide the rotational degrees of freedom and ensure the stability of the device and the repeatability of the recordings.

the repeatability of the experiments. Finally, the use of cameras and microphones gives to the POPEYE robot head audio-visual sensory capabilities in one device that geometrically links all six sensors.

All sequences from the data set were recorded in a regular meeting room, shown in Figure A.3. Whilst two diffuse lights were included in the setup to provide for good illumination, no devices were used to modify neither the effects of the sunlight nor the acoustics characteristics of the room. Hence, the recordings are affected by exterior illumination changes, acoustic reverberations, outside noise, and all kind of audio and visual interferences and artifacts present in unconstrained indoor scenes.

For each sequence, we acquired several streams of data distributed in two groups: the *primary* data and the *secondary* data. While the first group is the data acquired using the POPEYE robot's sensors, the second group was acquired by means of devices external to the robot. The *primary* data consists of the audio and video streams captured using POPEYE. Both, left and right, cameras have a resolution of 1024×768 and two operating modes: 8-bit gray-scale images at 30 frames per second (FPS) or 16-bit YUV-color images at 15 FPS. The four Soundman OKM II Classic Solo microphones mounted on the Sennheiser MKE 2002 dummy-head were linked to the computer via the Behringer ADA8000 Ultragain Pro-8 digital external sound card sampling at 48 kHz. The *secondary*

data are meant to ease the task of manual annotation for ground-truth. These data consist of one flock of birds (FoB) stream (by Ascension technology) to provide the absolute position of the actor in the scene and up to four wireless close-range microphones PYLE PRO PDWM4400 to capture the audio track of each individual actor.

Both cameras were synchronized by an external trigger controlled by software. The audio-visual synchronization was done by means of a clapping device. This device provides an event that is sharp – and hence, easy to detect – in both audio and video signals. The FoB was synchronized to the visual stream in a similar way: with a sharp event in both FoB and video signals. Regarding the visual calibration, the state-of-the-art method described in Bouguet (2008) uses several image-pairs to provide an accurate calibration. The audio-visual calibration is manually done by annotating the position of the microphones with respect to the cyclopean coordinate frame Hansard & Horaud (2008).

Following the arguments presented in the previous paragraphs it can be concluded that the setup suffices conceptual and technical validation. Hence, the sequences have an intrinsic value when used to benchmark algorithm targeting HRI applications. The next section is devoted to fully detail the recorded scenarios forming the Ravel data set.

A.3 Data Set Annotation

Providing the ground truth is an important task when delivering a new data set; this allows to quantitatively compare the algorithms and techniques using the data. In this section we present two types of annotation data provided together with the data set.

A.3.1 Action Performed

The first kind of annotation we provided is related to the action and robot gesture scenarios of the data set. This annotation is done using a classical convention,

that each frame is assigned a label of the particular action. Since the played action is known only one label is assigned to each frame. Because the annotation we need is not complex a simple annotation tool was designed for this purpose in which a user labels each start and end of each action/gesture in the recordings. The output of that tool is written in the standard ELAN Brugman *et al.* (2004) annotation format. A screen shot of the annotation tool is shown in Figure A.5.

A.3.2 Position and Speaking State

The second kind of annotations concern the interaction part of the data set and consists on the position of the actors (both in the images and in the 3D space) and on the speaking state of the actors. In both cases the annotator uses a semi-automatic tool that outputs an ELAN-readable output file. The semi-automatic procedures used are described in the following.

Regarding the annotation of the actors' position, the tracking algorithm described in Tracking-Learning-Detection (2012) is used to semi-automatize the process. The annotator is asked for the object's bounding box, which is then tracked along time. At any point, the annotator can reinitialize the tracker to correct its mistakes. Once the object is tracked along the entire left camera image sequence, the correspondent trajectory in the other image is automatically estimated. To do that, the classical approach of maximizing the normalized cross-correlation across the epipolar constraint is used Hartley & Zisserman (2004). From these correspondence pairs, the 3D location is computed at every frame using the DLT reconstruction procedure Hartley & Zisserman (2004). The location of the speaker in the images is given in pixels and the position in the 3D space are given in millimeters with respect to the cyclopean coordinate reference frame Hansard & Horaud (2008).

Concerning the speaking state, the state-of-the-art voice activity detector described in Brookes (2013) is used on the per-actor close range microphones. In a second step, the annotator is in charge of discarding all false positives generated by the VAD, leading to a clean speaking state annotation per each actor.

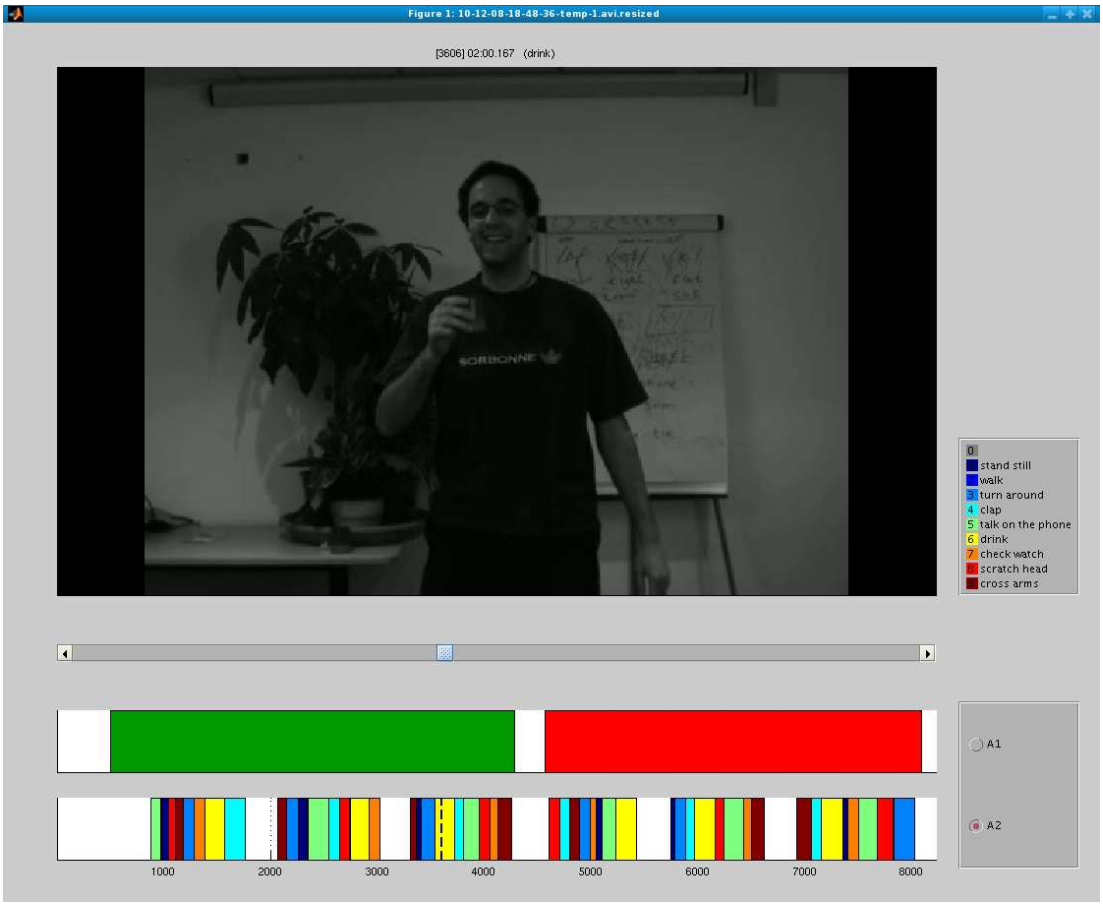


Figure A.5: The annotation tool screen shot. Two time lines are shown below the image. The first one (top) is used to annotate the level of background clutter. The second one (bottom) details which action is performed at each frame.

Actor 1	(enters room, positions himself in front of robot and looks at robot)
Actor 1	Hello, I'm Actor 1.
Robot	Hello, I'm Nao. Nice to meet you.
Actor 2	(enters room, positions himself next to Actor 1 and looks at robot)
Robot	Excuse me for a moment.
Robot	Hello, I'm currently talking to Actor 1. Do you know Actor 1?
Actor 2	No, I don't know him.
Robot	Then let me introduce you two. What is your name?
Actor 2	Actor 2
Robot	Actor 2, this is Actor 1. Actor 1 this is Actor 2.
Actor 3	(enters room, positions himself next to Actor 1, looks at Actor 1and Actor 2)
Actor 3	Actor 1 and Actor 2, have you seen Actor 4?
Actor 2	No I'm sorry, we haven't seen her.
Actor 3	Ok, thanks. I'll have to find her myself then. Bye.
Actor 3	(leaves)
Actor 2	Actor 1, (turn heads towards robot)
Actor 1	We have to go too. Bye
Robot	Ok. See you later.

Script 3: Detail of the script of the scenario "*Introducing people - Passive*". The three people interact with the robot. The robot is static in this scenario.

Actor 1	(enters room, positions himself in front of robot and looks at robot)
Actor 1	Hello, I'm Actor 1.
Robot	Hello, I'm Nao. Nice to meet you.
Actor 2	(enters room, positions himself next to Actor 1 and looks at robot)
Robot	Excuse me for a moment.
Robot	(turns head towards Actor 2)
Actor 1	(turns head towards Actor 2)
Robot	Hello, I'm currently talking to Actor 1. Do you know Actor 1?
Actor 2	No, I don't know him.
Robot	Then let me introduce you two. What is your name?
Actor 2	Actor 2
Robot	Actor 2 this is Actor 1. (turns head towards Actor 1) Actor 1 this is Actor 2.
Actor 3	(enters room, walks somewhere behind Actor 1 and Actor 2, leaves room)
Actor 1	We have to go now. Bye
Robot	(turns head towards Actor 1)
Robot	Ok. See you later.

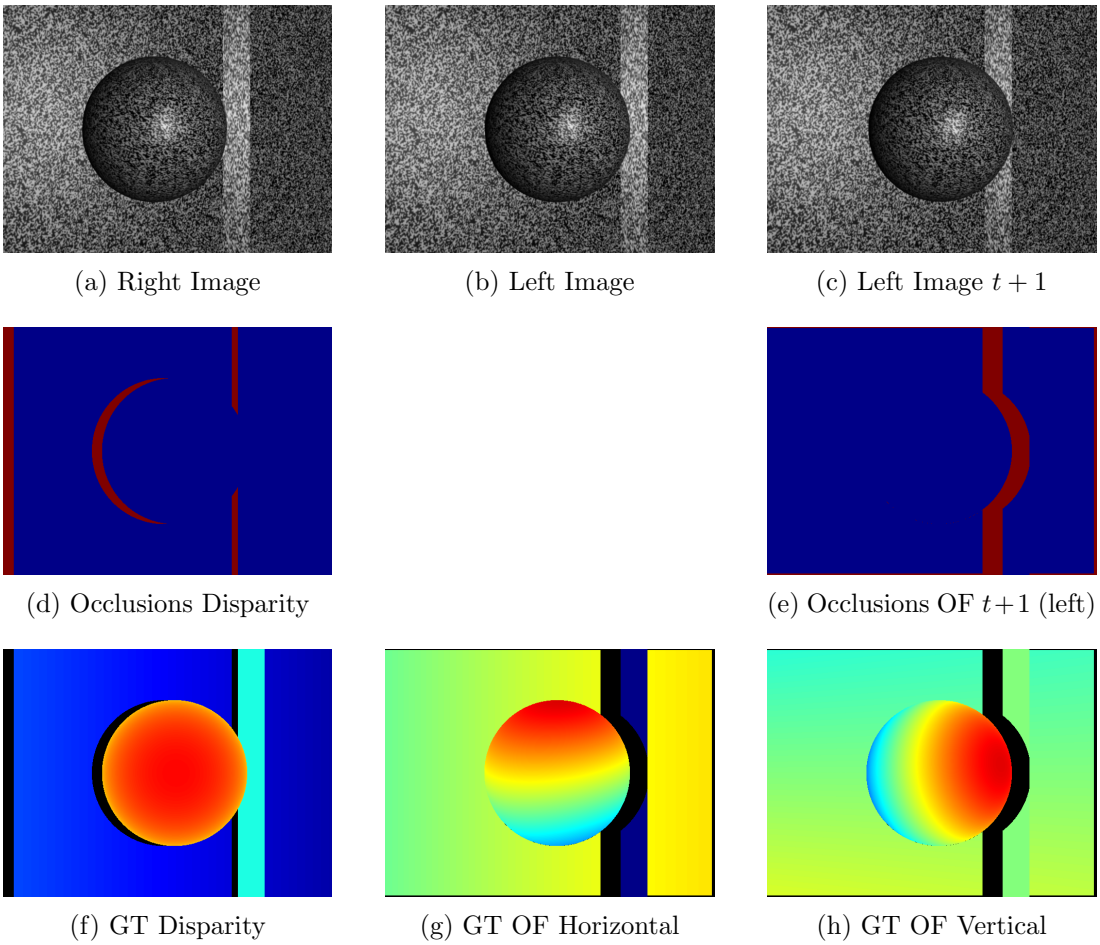
Script 4: Detail of the script of the scenario "*Introducing people - Active*". Two out of the three people interact with the robot. The latter is a moving robot.

Appendix B

Ground Truth for SceneFlow

In Chapter 2,3 a scene with ground truth is used to validate the algorithms and evaluate them quantitatively. Due the lack of suitable ground truth datasets we decide to create a new one from scratch. In the following lines is described the theory and the steps that are done to create this new dataset. This can be useful if in the future more complex scenes needs to be designed to evaluate other scenarios. To generate ground truth we assume cameras (intrinsic and extrinsic parameters) and 3D vertices of the scene to be known. In section B.2 is explained in more detail how the scene is constructed. The technique used to generate the ground truth disparity and optical flow is ray tracing, consisting in project a ray from the camera to the scene and detect which objects rendered on the scene intersects with the ray. If it intersects with several objects, the closest to the camera is the point taken. For disparity a ray is traced with the left camera, the 3D world point is computed and then projected back to the right camera, the difference between x-coordinates in the image plane is taken as the disparity. For the scene flow, a ray is traced in the time t, the the scene is moved to the time $t + 1$, and again a ray is traced and the point projected back to the left camera. The difference between the two points in the image plane is the optical flow. The objects are textured with white noise, which is the texture that best correlates when searching for correspondences.

In our particular scene there are only planes and spheres, so we will center in the intersection of a ray with a plane and a sphere, actually a plane can be



(a) Right Image (b) Left Image (c) Left Image $t+1$

(d) Occlusions Disparity (e) Occlusions OF $t+1$ (left)

(f) GT Disparity (g) GT OF Horizontal (h) GT OF Vertical

Figure B.1: Information available after ground truth generation. (a),(b) is the right/left image respectively at time t. (c) is left image at time $t+1$. (d) is the occlusion map for disparity. (e) is the occlusion map corresponding to the optical flow at time $t+1$ for the left image. (f) is the disparity (taking left image as reference). (g),(h) is the optical flow horizontal/vertical respectively.

considered as two triangles, so we will explain how to intersect a triangle with a ray and a sphere with a ray, in the next section.

B.1 Theory

Here we give a few mathematical notes on how ray tracing works.

B.1.1 Ray equation

A Ray it's a line and can be determined by two points or equivalently by a point and a direction. In case of cameras we know the camera center, and where the ray projects into the camera plane, following Hartley & Zisserman (2004) book pag. 162, we find that a Ray can be expressed as: $R(t) = (M^{-1}x \quad 0) + (C_0 \quad 1)$ where M is the 3x3 submatrix of camera matrix (without last column). Ensure that the determinant of M is positive in order to guarantee that the rays goes in front direction of the camera and not backwards, otherwise multiply M by -1. C_0 is the center position of the camera. x is the point in the camera plane in homogeneous coordinates. The equation can be rewriten as: $R(t) = (x(t), y(t), z(t)) = (c_0, c_0, c_0) + (x_1, y_1, z_1)t$

B.1.2 Intersecting a Ray with a triangle

Given three points $P_1 = (x_1, y_1, z_1)$, $P_2 = (x_2, y_2, z_2)$, and $P_3 = (x_3, y_3, z_3)$ that defines a triangle, the normal vector to the triangle is $n = (a, b, c) = (P_2 - P_1)x(P_3 - P_1)$

The equation of the plane that contains the triangle can be written as $ax + by + cz + d = 0$ where a, b and c are the components of the normal vector and the value of d is obtained substituting any of the vertices to the plane equation. Taking the first vertex this results to $d = -(ax_1 + by_1 + cz_1)$

To find the intersection point with the ray and the triangle we equals the two equations giving: $(ax_1 + by_1 + cz_1)t + (ax_0 + by_0 + cz_0 + d) = 0$ Getting the value

t we can substitue then into the ray equation to obtain the 3D point when ray and triangle intersects.

It could be possible that the ray doesn't intersect the triangle in any point, this can be determined computing the barycentric coordinates of the point.

B.1.2.1 Determine if a ray intersects a triangle

The barycentric coordinates are used to describe a point inside a triangle the equation follows like: $P = sP_s + tP_t + uP_u$ where P is the point inside a triangle and The barycentric coordinates can be computed as:

$$s = \frac{(P_t - P)x(P_u - P)}{(P_t - P_s)x(P_u - P_s)} \tag{B.1}$$

$$t = \frac{(P_u - P)x(P_s - P)}{(P_t - P_s)x(P_u - P_s)} \tag{B.2}$$

$$u = \frac{(P_s - P)x(P_t - P)}{(P_t - P_s)x(P_u - P_s)} \tag{B.3}$$

B.1.3 Intersect a Ray with an sphere

To find where a shpere intersect with a ray we do similarly as befor we equals the ray equation with the sphere equation $(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2 - r^2 = 0$ where (x_c, y_c, z_c) is the center of the sphere and r is the radius of the sphere. As before we equal the ray equation with sphere equation giving: $at^2 + bt + c = 0$ where $a = (x_1^2 + y_1^2 + z_1^2)$ $b = 2[(x_0 - x_c)x_1 + (y_0 - y_c)y_1 + (z_0 - z_c)z_1]$ and $c = [(x_0 - x_c)^2 + (y_0 - y_c)^2 + (z_0 - z_c)^2 - r^2]$ giving the following solutions: $t = \frac{-b+\sqrt{(b^2-4ac)}}{2a}$ and $t = \frac{-b-\sqrt{(b^2-4ac)}}{2a}$ Again substituting the value of t in the ray equation we can obtain the 3D point.

B.1.3.1 Determine if a ray intersects a sphere

If $b^2 - 4ac < 0$ then the ray doesn't hit the sphere, if equals to 0 the ray is tangent and otherwise intersects the sphere by two points.

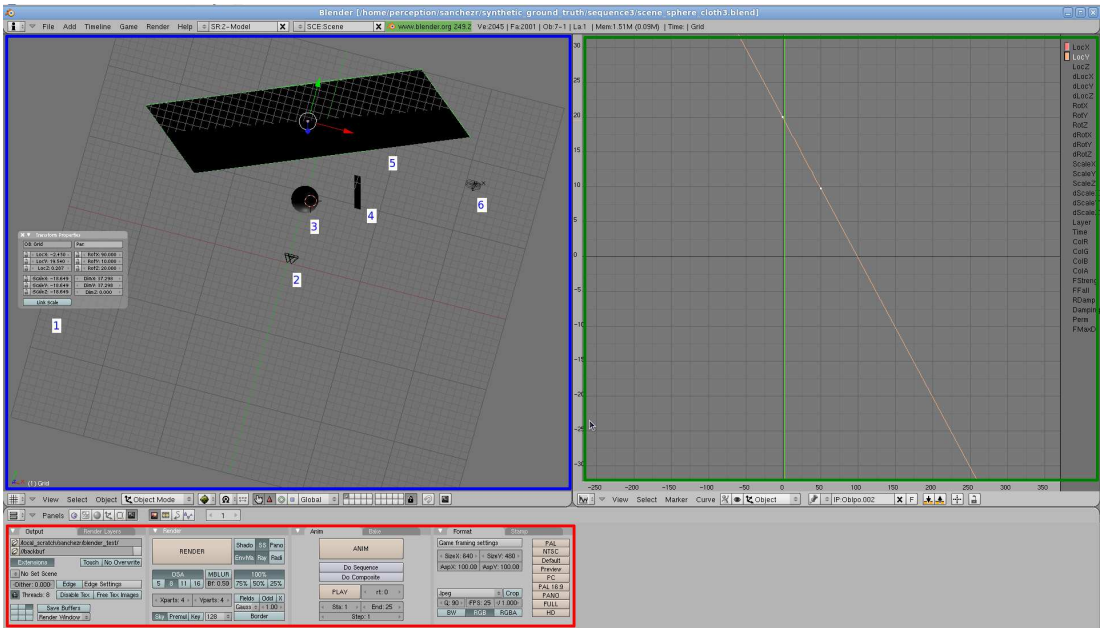


Figure B.2: Screenshot of Blender software.

B.2 Blender

In Figure B.2 an screenshot of a Blender session is depicted. Three different parts are marked with different colors: blue, green and red. The "blue" correspond to the world editing. With number '1' is marked the panel that controls the position of the objects. Number '2' is the stereo camera rig. Number '3' is an sphere, '4' a vertical bar, '5' a background plane, and '6' is wind simulation in case some deformable object want to be added into the scene. Since all the objects defined are rigid the wind has no effect on them. In "green" is marked the physics editor panel. With this panel we determine the velocities and acceleration of each object on each of all directions of the axes. Note that in the example is a rect (constant acceleration) but it can be more complicated and produce some curves. Finally in "red" is the rendering panel, which allows to render the scene, give some texture and produce the animations.

Appendix C

Publications

C.1 Journals

- RAVEL: An Annotated Corpus for Training Robots with Audiovisual Abilities. X. Alameda-Pineda, J. Sanchez-Riera, et al. In Journal on Multimodal User Interfaces (JMUI), 2012.
- Feature distribution modelling techniques for 3D face verification. C. McCool, J. Sanchez-Riera, S. Marcel. Pattern Recognition Letters. Feb. 2009.

C.2 Conferences

- Benchmarking methods for Audio-Visual Recognition using Tiny Training Sets. X. Alameda-Pineda, J. Sanchez-Riera, R. Horaud. In ICASSP, 2013.
- Action Recognition Robust to Background Clutter by Using Stereo Vision. J. Sanchez-Riera, J. Cech, R. Horaud. In WS. on Video Event Categorization, Tagging and Retrieval (VeCTaR), 2012.
- Audio-Visual Robot Command Recognition. J. Sanchez-Riera, X. Alameda-Pineda, R. Horaud. In International Conference on Multimodal Interaction (ICMI),2012.

- Online multimodal speaker detection for humanoid robots. J. Sanchez-Riera, X. Alameda-Pineda, et al. IEEE International Conference on Humanoid Robotics (HUMANOIDS), 2012.
- Robust Spatiotemporal Stereo for Dynamic Scenes. J. Sanchez-Riera, J. Cech, R. Horaud. In 21st International Conference on Pattern Recognition (ICPR), 2012.
- Scene Flow Estimation by Growing Correspondence Seeds. J. Cech, J. Sanchez-Riera, R. Horaud. In Computer Vision and Pattern Recognition (CVPR), 2011.
- Simultaneous Pose, Correspondence and Non-Rigid Shape. J. Sanchez-Riera, J. Ostlund, P. Fua, F. Moreno-Noguer. In Computer Vision and Pattern Recognition (CVPR), 2010.
- Indoor PTZ Camera Calibration with Concurrent PT Axes. J. Sanchez-Riera, J. Salvador, J. R. Casas. In Int. Conf. on Computer Vision Theory and Applications (VISAPP). 2009.

References

Alameda-Pineda, X., Khalidov, V., Horaud, R. & Forbes, F. (2011). Finding audio-visual events in informal social gatherings. In *Proceedings of the International Conference on Multimodal Interaction*. 74, 76

Basha, T., Moses, Y. & Kiryati, N. (2010). Multi-view scene flow estimation: A view centered variational approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 9, 27

Beal, M.J., Attias, H. & Jojic, N. (2002). Audio-visual sensor fusion with probabilistic graphical models. In *Proceedings of the European Conference on Computer Vision*. 72

Bleyer, M. & Gelautz, M. (2009). Temporally consistent disparity maps from uncalibrated stereo videos. In *Internation Symposium on Image and Signal Processing and Analysis*. 9

Bo, L., Ren, X. & Fox, D. (2010). Kernel descriptors for visual recognition. In *Advances in Neural Information Processing Systems*. 96

Bouguet, J.Y. (2008). Camera calibration toolbox for Matlab. `http://www.vision.caltech.edu/bouguetj/calib_doc/`. 106

Bregonzio, M., Gong, S. & Xiang, T. (2009). Recognising action as clouds of space-time interest points. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 48

Brookes, M. (2013). Voicebox: Speech processing toolbox for matlab. `http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html`. 107

Brox, T. & Malik, J. (2010). Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, in press. 36, 40, 42, 43, 44, 45

Brugman, H., Russel, A. & Nijmegen, X. (2004). Annotating multi-media / multimodal resources with elan. In *Proceedings of the International Conference on Language Resources and Evaluation*, 2065–2068. 107

Čech, J. & Šára, R. (2007). Efficient sampling of disparity space for fast and accurate matching. In *Proceedings of the BenCOS 2007: CVPR Workshop Towards Benchmarking Automated Calibration, Orientation and Surface Reconstruction from Images*. 27, 34, 35, 38

Čech, J., Matas, J. & Perďoch, M. (2010). Efficient sequential correspondence selection by cosegmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32**. 12, 14, 15, 16, 25, 27, 34, 40

Čech, J., Sanchez-Riera, J. & Horaud, R. (2011). Scene flow estimation by growing correspondence seeds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 9, 25, 49, 51

Chan, Y., Tsui, W., So, H. & Ching, P. (2006). Time-of-arrival based localization under nlos conditions. *IEEE Transactions on Vehicular Technology*, **55**, 17–24. 76

Cherry, E.C. (1953). Some experiments on the recognition of speech, with one and with two ears. *Journal of the Acoustical Society of America*, **25**, 975–979. 100

Dalal, N. & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 47

Davis, J., Nehab, D., Ramamoorthi, R. & Rusinkiewicz, S. (2005). Spacetime stereo: A unifying framework for depth from triangulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**, 296–302. 9

Dobiaš, M. & Šára, R. (2011). Real-time global prediction for temporally stable stereo. Research Report CTU–CMP–2011–13, Center for Machine Perception, K13133 FEE Czech Technical University. 25, 87

Dollár, P., Rabaud, V., Cottrell, G. & Belongie, S. (2005). Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, 65–72. 48

Ess, A., Leibe, B., Schindler, K. & van Gool, L. (2009). Moving obstacle detection in highly dynamic scenes. In *Proceedings of the International Conference on Robotics and Automation*. 8

Fehr, D., Cherian, A., Sivalingam, R., Nickolay, S., Morellas, V. & Papanikolopoulos, N. (2012). Compact covariance descriptors in 3d point clouds for object recognition. In *Proceedings of the International Conference on Robotics and Automation*, 1793–1798. 96

Fisher, J.W. & Darrel, T. (2004). Speaker association with signal-level audiovisual fusion. *IEEE Transactions on Multimedia*, **6**. 72

Gehrig, D., Krauthausen, P., Rybok, L., Kuehne, H., Hanebeck, U.D., Schultz, T. & Stiefelhagen, R. (2011). Combined intention, activity, and motion recognition for a humanoid household robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 60

Gong, M. (2009). Real-time joint disparity and disparity flow estimation on programmable graphics hardware. *Journal on Computer Vision and Image Understanding*, **113**. 27, 35

Hadfield, S. & Bowden, R. (2011). Kinecting the dots: Particle based scene flow from depth sensors. In *Proceedings of the International Conference on Computer Vision*, 2290 –2295. 95

Hansard, M. & Horaud, R. (2008). Cyclopean geometry of binocular vision. *Journal of the Optical Society of America*, **25**, 23572369. 76, 106, 107

Harris, C. & Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of The Fourth Alvey Vision Conference*, 147–151. 16

Hartley, R.I. & Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518. 107, 113

Haykin, S. & Chen, Z. (2005). The cocktail party problem. *Journal on Neural Compututation*, **17**, 1875–1902. 100

Holte, M.B., Moeslund, T.B. & Fihl, P. (2010). View-invariant gesture recognition using 3d optical flow and harmonic motion context. *Journal on Computer Vision and Image Understanding*, **114**, 1353–1361. 48

Huang, C.M. & Mutlu, B. (2012). Robot behavior toolkit: Generating effective social behaviors for robots. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*. 4

Huang, T., Lin, K.H., Xiao, J., Wang, Z. & Wang, J. (2012). Substructure and boundary modeling for continuous action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 96

Huguet, F. & Devernay, F. (2007). A variational method for scene flow estimation from stereo sequences. In *Proceedings of the International Conference on Computer Vision*. 9, 27, 36, 38, 40, 42, 43, 44, 45

Isard, M. & MacCormick, J. (2006). Dense motion and disparity estimation via loopy belief propagation. In *Proceedings of the Asian Conference on Computer Vision*. 9, 27

Itohara, T., Otsuka, T., Mizumoto, T., Ogata, T. & Okuno, H.G. (2011). Particle-filter based audio-visual beat-tracking for music robot ensemble with human guitarist. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 71

Jenkins, O., Gonzalez, G. & Loper, M. (2007a). Interactive human pose and action recognition using dynamical motion primitives. *International Journal of Humanoid Robotics*, **4**, 365–385. 60

Jenkins, O.C., González, G. & Loper, M.M. (2007b). Tracking human motion and actions for interactive robots. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*. 60

Kannala, J. & Brandt, S.S. (2007). Quasi-dense wide baseline matching using match propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 28, 34

Khalidov, V., Forbes, F. & Horaud, R. (2011). Conjugate mixture models for clustering multimodal data. *Neural Computation*, **23**, 517–557. 74, 76

Khalidov, V., Forbes, F. & Horaud, R. (2012). Calibration of a binocular-binaural sensor using a moving audio-visual target. Tech. Rep. 7865, INRIA Grenoble Rhone-Alpes. 75, 78

Klaser, A., Marszalek, M. & Schmid, C. (2008). A spatio-temporal descriptor based on 3D-gradients. In *Proceedings of the British Machine Vision Conference*. 48

Koenemann, J. & Bennewitz, M. (2012). Whole-body imitation of human motions with a nao humanoid. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*. 59

Lacheze, L., Guo, Y., Benosman, R., Gas, B. & Couverture, C. (2009). Audio/video fusion for objects recognition. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 61, 71

Laptev, I. (2005). On space-time interest points. *International Journal on Computer Vision*, **64**, 107–123. 47, 49, 50, 52, 53, 55, 56, 58

Larsen, E.S., Mordohai, P., Pollefeys, M. & Fuchs, H. (2007). Temporally consistent reconstruction from multiple video streams using enhanced belief propagation. *Proceedings of the International Conference on Computer Vision*. 9

Leeper, A., Hsiao, K., Chu, E. & Salisbury, K. (2010). Using near-field stereo vision for robotic grasping in cluttered environments. In *ISER*. 8

Lhuillier, M. & Quan, L. (2002). Match propagation for image-based modeling and rendering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**. 28

Li, W., Zhang, Z. & Liu, Z. (2010). Action recognition based on a bag of 3d points. In *Proc. CVPR workshop on Human Communicative Behaviour Analysis*. 48

Li, Z., Herfet, T., Grochulla, M. & Thormhlen, T. (2012). Audio-visual multiple active speaker localisation in reverberant environments. In *Conference on Digital Audio Effects*. 96

Lili, N.A. (2009). A framework for human action detection via extraction of multimodal features. *International Journal of Image Processing*, **3**. 70

Liu, F. & Philomin, V. (2009). Disparity estimation in stereo sequences using scene flow. In *Proceedings of the British Machine Vision Conference*. 9, 27

Lopes, J. & Singh, S. (2006). Audio and video feature fusion for activity recognition in unconstrained videos. In *Intelligent Data Engineering and Automated Learning*. 61

Lucas, B. & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 34

Luo, R.C. & Kay, M.G. (1989). Multisensor integration and fusion in intelligent systems. *IEEE Transactions on Systems, Man and Cybernetics*, **19**, 901–931. 70

Marks, T.K., Howard, A., Bajracharya, M., Cottrell, G.W. & Matthies, L. (2008). Gamma-slam: using stereo vision and variance grid maps for slam in unstructured environments. In *Proceedings of the International Conference on Robotics and Automation*. 8

Moravec, H.P. (1977). Towards automatic visual obstacle avoidance. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 584. 13, 31

Nakadai, K., Matsuura, D., Okuno, H.G. & Tsujino, H. (2004). Improvement of recognition of simultaneous speech signals using av integration and scattering theory for humanoid robots. *Speech Communication*, 97–112. 71

Ni, B., Wang, G. & Moulin, P. (2011). Rgbd-hudaact: A color-depth video database for human daily activity recognition. In *Proc. ICCV Workshop on Consumer Depth Cameras for Computer Vision*. 49

Pons, J.P., Kerive, R., Faugeras, O. & Hermosillo, G. (2003). Variational stereovision and 3D scene flow estimation with statistical similarity measures. In *Proceedings of the International Conference on Computer Vision*. 27

Poppe, R. (2010). A survey on vision-based human action recognition. *Journal on Image and Vision Computing*, **28**, 976–990. 47

Rabiner, L.R. & Schafer, R.W. (2011). *Theory and Applications of Digital Speech Processing*. Pearson. 64

Ramasubramanian, V., Karthik, R., Thiyagarajan, S. & Cherla, S. (2011). Continuous audio analytics by hmm and viterbi decoding. In *Proceedings of the IEEE International Conference on Audio, Speech and Signal Processing*, 2396–2399, IEEE. 64

Ramey, A., Gonzalez-Pacheco, V. & Salichs, M.A. (2011). Integration of a low-cost rgb-d sensor in a social robot for gesture recognition. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*. 60

Raptis, M., Kirovski, D. & Hoppes, H. (2011). Real-time classification of dance gestures from skeleton animation. In *SIGGRAPH/Eurographics Symposium on Computer Animation*. 60

Richardt, C., Orr, D., Davies, I., Criminisi, A. & Dodgson, N.A. (2010). Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid. In *Proceedings of the European Conference on Computer Vision*. 11, 27

Roggen, D., Magnenat, S., Waibel, M. & Tröster, G. (2011). Designing and sharing activity recognition systems across platforms: methods from wearable computing. *IEEE Robotics and Automation Magazine*, **12**, 83–95. 59

Roh, M.C., Shin, H.K. & Lee, S.W. (2010). View-independent human action recognition with volume motion template on single stereo camera. *Pattern Recognition Letters*, **31**, 639–647. 48

ROS (2013). message_filters/approximatetime. http://www.ros.org/wiki/message_filters/ApproximateTime, accessed: 06/21/02012. 86

Sanchez-Riera, J., Čech, J. & Horaud, R. (2012). Action recognition robust to background clutter by using stereo vision. In *International Workshop on Video Event Categorization, Tagging and Regrieval (VECTaR), in conjunction with IEEE ECCV*. 62

Shi, J. & Tomasi, C. (1994). Good features to track. In *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 13

Shivappa, S.T., Rao, B.D. & Trivedi, M.M. (2010a). Auvio-visual fusion and tracking with multilevel iterative decoding: Framework and experimental evaluation. *Journal of Selected Topics in Signal Processing*. 73, 96

Shivappa, S.T., Trivedi, M.M. & Rao, B.D. (2010b). Audiovisual information fusion in human-computer interfaces and intelligent environments: A survey. *Proceedings of the IEEE*, **98**, 1692–1715. 96

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T. & Finocchio, M. (2011). Real-time human pose recognition in parts from single depth images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 49, 60

Sizintsev, M. & Wildes, R.P. (2009). Spatiotemporal stereo via spatiotemporal quadratic element (stequel) matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 11, 17, 19, 22, 27, 36, 38, 42, 43, 44, 45

Šochman, J. & Matas, J. (2005). Waldboost – learning for time constrained sequential detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 51, 76

Stankiewicz, O. & Wegner, K. (2010). Generation of temporally consistent depth maps using nosie removal from video. In *Proceedings of the 2010 international conference on Computer vision and graphics: Part II*, 292–299. 9

Sung, J., Ponce, C., Selman, B. & Saxena, A. (2012). Unstructured human activity detection from rgbd images. In *Proceedings of the International Conference on Robotics and Automation*. 49, 60

Talukder, A. & Matthies, L. (2004). Real-time detection of moving objects from moving vehicle using dense stereo and optical flow. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 8

Tracking-Learning-Detection (2012). Zdenek kalal and krystian mikolajczyk and jiri matas. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7, 1409–1422. 107

Tuytelaars, T. (2010). Dense interest points. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 48

Uddin, M.Z., Thang, N.D., Kim, J.T. & Kim, T.S. (2011). Human activity recognition using body joint-angle features and hidden markov model. *ETRI Journal*, **33**, 569–579. 48

Vedula, S., Baker, S., Rander, P., Collins, R. & Kanade, T. (1999). Three-dimensional scene flow. In *Proceedings of the International Conference on Computer Vision*. 27

Volkhardt, M., Müller, S., Schröter, C. & Gross, H.M. (2010). Real-time activity recognition on a mobile companion robot. In *Proceedings of International Scientific Colloquium*. 60

Wang, H., Ullah, M.M., Klaser, A., Laptev, I. & Schmid, C. (2009). Evaluation of local spatio-temporal features for action recognition. In *Proceedings of the British Machine Vision Conference*. 48

Wang, H., Kläser, A., Schmid, C. & Liu, C.L. (2011). Action recognition by dense trajectories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 48

Weinland, D., Ronfard, R. & Boyer, E. (2006). Free viewpoint action recognition using motion history volumes. *Journal on Computer Vision and Image Understanding*, **104**, 249–257, http://4drepository.inrialpes.fr/public/viewgroup/6. 99

Weinland, D., Boyer, E. & Ronfard, R. (2007). Action recognition from arbitrary views using 3d exemplars. In *Proceedings of the International Conference on Computer Vision*. 48

Weinland, D., Ronfard, R. & Boyer, E. (2011). A survey of vision-based methods for action representation, segmentation and recognition. *Journal on Computer Vision and Image Understanding*, **115**, 224–241. 47

Wienke, J. & Wrede, S. (2011). A middleware for collaborative research in experimental robotics. In *2011 IEEE/SICE Int. Symposium on System Integration, SII2011*, IEEE, IEEE, Kyoto, Japan. 85

Wu, Q., Wang, Z., Deng, F. & Feng, D. (2010). Realistic human action recognition with audio context. In *DICTA*. 70

Xia, L., Chen, C.C. & Aggarwal, J.K. (2012). View invariant human action recognition using histograms of 3d joints. In *Proceedings CVPR workshop on Human Activity Understanding from 3D Data (HAU3D)*. 49, 61

Yamazaki, A., Yamazaki, K., Ohyama, T., Kobayashi, Y. & Kuno, Y. (2012). A techno-sociological solution for designing a museum guide robot: Regarding choosing and appropriate visitor. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*. 5

Yan, P., Khan, S.M. & Shah, M. (2008). Learning 4d action feautre models for arbitrary view action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 48

Yves Bouguet, J. (2010). Camera calibration toolbox for matlab. Http://www.vision.caltech.edu/bouguetj/calib_doc/. 77

Zhang, C., Yin, P., Rui, Y., Cutler, R., Viola, P., Sun, X., Pinto, N. & Zhang, Z. (2008). Boosting-based multimodal speaker detection for distributed meeting videos. *IEEE Transactions on Multimedia*, **10**. 73

Zhang, G., Jia, J. & Bao, H. (2011a). Simultaneous multi-body stereo and segmentation. In *Proceedings of the International Conference on Computer Vision*, 826–833. 95

Zhang, H. & Parker, L.E. (2011). 4-dimensional local spatio-temporal features for human activity recognition. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 48

Zhang, L., Curless, B. & Seitz, S.M. (2003). Spacetime stereo: Shape recovery for dynamic scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 11, 13, 17, 27, 33

Zhang, Z., Wang, Y., Jiang, T. & Gao, W. (2011b). Stereoscopic learning for disparity estimation. In *IEEE International Symposium on Circuits and Systems*, 365 –368. 95

Zhou, Q., Yu, S., Wu, X., Gao, Q., Li, C. & Xu, Y. (2009). Hmms-based human action recognition for an intelligent household surveillance robot. In *ROBIO*. 59

Zhu, C. & Sheng, W. (2009). Human daily activity recognition in robot-assisted living using multi-sensor fusion. In *Proceedings of the International Conference on Robotics and Automation*. 59

Zhu, J., Wang, L., Gao, J. & Yang, R. (2010). Spatial-temporal fusion for high accuracy depth maps using dynamic mrfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32**, 899–909. 9