Pedestrian head detection using automatic scale selection for feature detection and statistical edge curvature analysis

Raffaele Ceccarelli
Diploma project
Assistant: Gianluca Antonini
Supervisor: Dr. Jean-Philippe Thiran

Lausanne, September 2004

Siamo arrivati alla fine di questo progetto e vorrei ringraziare tutte le persone che mi hanno permesso di poter fare questa esperienza. Un ringraziamento va al prof. Jean-Philippe Thiran e al prof. Alessandro Mecocci che mi hanno dato la possibilita' di sviluppare questo progetto e soprattutto grazie a Gianluca per quanto mi ha seguito e per la ”pazienza” avuta nei miei confronti. Ringrazio fortemente i miei genitori, Benedetta e gli amici ”vecchi” e ”nuovi” che mi hanno accompagnato e che fanno parte della mia vita.

Abstract

In this report we focus on pedestrian head detection and tracking in video sequences. The task is not trivial in real and complex scenarios where the deformation induced by the perspective field requires a multi-scale analysis. Multi-scale shape models for the human head are considered to identify the correct size of the region of interest. Anisotropic diffusion is used as a pre-processing step and edge detection is performed using an automatic scale selection process. A non parametric statistical description is given for the edge curvature and detection is performed by means of goodness-of-fit tests. The head detector is used as a validation tool in a correlation-based tracker. The local maxima of the correlation matrix are analyzed. Tracking is performed associating the displacement vector of the target with that local maximum which maximizes the goodness-of-fit with the distribution of the edge curvature of the head.

Contents

List of Figures **5**

1 Introduction **6**

2 State of the art **8**
2.1 Image segmentation and object detection 8
2.2 Tracking and motion-based approaches 11

3 Exploiting the prior information **15**
3.1 Background/foreground modeling 15
3.2 Camera calibration and top-view representation 16
3.3 Patch resizing 19
3.4 Head shape models 22

4 Head detector **32**
4.1 System overview 32
4.2 Automatic scale selection 33
4.3 Edge detection 34
4.4 Template based matching 37
4.5 Statistical edge curvature analysis 40
4.6 Spatial Clustering 46

5 Head based tracker **47**
5.1 Correlation maxima computation
and tracking 47

6 Experiments and Results **49**
6.1 Data-sets 49
6.2 Head detection results 49
6.3 Tracking results 50

List of Figures

3.1 Top view plane and image view plane. 17

3.2 Angles θ_v and θ_h from pixel coordinates. 18

3.3 Camera position and angle. 19

3.4 Different distance maps for different camera parameter vectors. We note the curves representing the top-view points at the same distance from the origin. The color represents the distance value. 20

3.5 Parameters related to the real head size. 21

3.6 The size of the patch changes according to the perspective field. 23

3.7 The head model is created according to the real human head size. Height and width of the head are linked to the patch size. 24

3.8 Gaussian function. 24

3.9 The first model used. 25

3.10 The first model used. 25

3.11 Ellipse curve. 26

3.12 Examples of elliptic models for two patches with different size. 27

3.13 Examples of elliptic models diffuse with Gaussian filter with 3 and 7 coefficients. 28

3.14 Axes of symmetry. 29

3.15 Maxima intensity value horizontal and vertical considering the edge map. 29

3.16 The angles of the tangent directions to each point of a circular shape, is uniformly distributed over the range $[-\pi/2, \pi/2]$. . . 30

3.17 GDM and the six regions. 31

4.1 Head detector scheme. 33

4.2 Examples of filter patch with Beltrami Flow. 35

4.3 An ideal step edge and its derivative profile. 36

4.4 Example of matching. 38

4.5 (a) Edge map of the current patch, (b) Diffuse mask, (c) Correlation result. 39

4.6 GDM defined by the gradient direction and the ROI. 41

4.7 Comparison between rejected candidate and accepted candidate. 42
4.8 Comparison between rejected candidate and accepted candidate. 43
4.9 Empirical distribution function for 100 normal random numbers. 45

5.1 Algorithm tracking. 48

6.1 Video1: head detection with template correlation matching. . 51
6.2 Video1: head detection with template correlation matching. . 52
6.3 Video1 : head detection with template correlation matching. . 53
6.4 Video1: head detection with edge curvature analysis (without threshold) and Chi-Square test as statistic tool. 54
6.5 Video1 : head detection with edge curvature analysis (without threshold) and K-S test as statistic tool. 55
6.6 Video1 : head detection with edge curvature analysis (without threshold) and with K-S test as statistic tool. 56
6.7 Video1 : head detection with edge curvature analysis (without threshold) and with K-S test as statistic tool. 57
6.8 Video2 : head detection with edge curvature analysis (without threshold) and Chi-Square test as statistic tool. 58
6.9 Video2 : head detection with edge curvature analysis (without threshold) and with K-S test as statistic tool. 59
6.10 Video2 : head detection with edge curvature analysis (without threshold) and with K-S test as statistic tool. 60
6.11 Video3: (a) Patch of fixed side 25 pixels, (a) Patch of fixed side 50 pixels, (a) Patch of fixed side 200 pixels. 61
6.12 Video3: Head detector. 62
6.13 Video4: Head detector. 62
6.14 Tracking with the first frame manually initialized. 63
6.15 Segmentation and tracking 65

Chapter 1

Introduction

The strong need of smart surveillance systems stems from those security-sensitive areas such as banks, department stores, parking lots, and borders. Surveillance cameras are already prevalent in commercial establishments, while camera outputs are usually recorded in tapes or stored in video archives. Smart surveillance has been proposed to measure traffic flow, monitoring of pedestrian congestion in public spaces, and others. In this work we focus on automatic pedestrian detection and tracking. They are not trivial tasks which are strongly dependent on illumination conditions, image quality and resolution. Moreover, the presence of shadows and occlusions between targets contributes to increase the complexity of the scene and consequently that of the detection and tracking algorithms.

The most of this work deals with pedestrian detection. Recognizing the human shape is a quite complex task. Human body is composed of multiple parts, each of which can move independently with many degrees of freedom. As a consequence, the number of possible different poses is quite large. Moreover, if we add multiple scales and partial occlusions the number of states becomes quickly too high to suggest an approach based on the whole human shape. In order to simplify our approach, we detect pedestrians by means of head detection. This choice is motivated by the suitable geometrical proprieties of the head shape. It is in fact easily represented using symmetric and rotation invariant curves. Moreover, the color content of the head region shows most of the time a certain degree of homogeneity.

Recording devices for video surveillance systems are placed indoor and outdoor. Often, the perspective field of the scene is quite depth. In order to face to these operational conditions a multi-scale approach is necessary and (at least) a certain degree of illumination invariance would be suitable. In this spirit we make an extensive use of the *a-priori* information. We know what we are looking for (i.e. pedestrian head). Moreover, most of the time the

camera devices for video surveillance are fixed. So, it is reasonable to assume to know the camera parameters. This working condition (calibrated camera) allows us to define how our target appears in different image positions, based on its scale. So, automatic scale selection is used for edge extraction and shape models are used to improve the head detection step.
Finally, we apply the head detector to the pedestrian tracking problem. We aim to track the pedestrian heads using a template-based method. A simple correlation-based tracker has been implemented. Random sampling techniques are adopted to estimate the maxima of the correlation function. Each maximum point is tested using the head detector and the *best-of-fit* is used to track the target.

Chapter 2

State of the art

2.1 Image segmentation and object detection

The segmentation is one of the most important techniques for image analysis, understanding and interpretation. Besides, it is required as a low-level step to a large number of high level computer vision task. Feature-based image segmentation is performed using two basic image process techniques: *Region-based segmentation* relies on the homogeneity of spatially localized features and properties, while the *Boundary-based segmentation* (which is often referred as edge-based) relies on the generation of a strength image and the extraction of prominent edges.

Feature-based image segmentation

Region-based segmentation

The first approach is made considering the color. If a specific color is used as representation of an object, it is fairly easy to identify all pixels with same color as the object. The first works in this direction were based on color-based models to segment the image and detect background/foreground objects. In [1] the author developed an algorithm to detect and track vehicles or pedestrian in real-time using color histogram based technique and Gaussian Mixture Models to describe the color distribution. Others region-based approaches can be roughly classified in two categories: *The Region-growing techniques* [2, 3, 4, 5] and the *Markov Random Fields based approaches* [6, 7, 8]. The region growing methods are based on split-and-merge procedures using statistical homogeneity test, where the statistics are generated and updated automatically, while the manner with which initial regions are formed and the criteria for splitting and merging them are set *a priori.*

In the Markov Random Fields (MRF), the segmentation problem is viewed as a statistical estimation where each pixel is statistically dependent only on its neighbors. The segmentation is obtained by finding the maximum a posteriori map given the observed data. The main advantages of this approaches is that they are less affected from the present of noise.

Boundary-based segmentation

The object detection task is complex being strongly dependent on the image quality, illumination conditions, presence of shadows and perspective object deformation. So, more complex methods have been developed based on shape constraints. Shape-based object detection is one of the hardest problems due to the difficulty of segmenting objects of interest in the images. Shape-based algorithms attempt to detect the boundaries of the objects using noise removal techniques and transformations invariant to scale and rotation for more complex scenes. Usually we consider the edges. They often occur at points where there is a large variation in the luminance values in an image, and consequently they often indicate the edges, or occluding boundaries, of the objects in a scene. Points of tangent discontinuity in the luminance signal (rather than simple discontinuity) can also signal an object boundary in the scene. The usual approach is to simply define edges as step discontinuities in the image signal. The method of localizing these discontinuities often then becomes one of finding local maxima in the derivative of the signal, or zero-crossings in the second derivative of the signal. This idea was first suggested to the AI community, both biologically and computationally, in [9], and later developed in [10, 11, 12, 13, 14]. In computer vision, edge detection is traditionally implemented by convolving the signal with some form of linear filter, usually a filter that approximates a first or second derivative operator. An odd symmetric filter will approximate a first derivative, and peaks in the convolution output will correspond to edges (luminance discontinuities) in the image. An even symmetric filter will approximate a second derivative operator. Zero-crossings in the output of convolution with an even symmetric filter will correspond to edges; maxima in the output of this operator will correspond to tangent discontinuities, often referred to as bars, or lines. Shape constraints have been recently introduced also in active contour models [15]. The detection and shape characterization of the objects becomes more difficult for complex scenes where there are many objects with occlusions and shading, (see [16, 17, 18, 19]).

Template-based segmentation

An another approach is based on the template-based matching which consist in a matching features between the template and the image sequence under analysis. There are two types of object template matching, *Fixed* and *Deformable-template-matching*.
Fixed templates are useful when object shapes do not change with respect to the viewing angle of the camera. We can use fixed templates with two techniques: image subtraction and correlation. In image subtraction, the template position is determined from minimizing the distance function between the template and various positions in the image. This technique perform well in restricted environments where imaging conditions, such as image intensity and viewing angles between the template and images containing this template are the same [20, 21]. Matching by correlation utilizes the position of the normalized cross-correlation peak between a template and an image to locate the best match. This technique is generally immune to noise and illumination effects in the images, but suffers from high computational complexity caused by summations over the entire template. Point correlation can reduce the computational complexity to a small set of carefully chosen points for the summations, (see in [22, 23]).
Deformable template matching approaches are used in cases where objects vary due to rigid and non-rigid deformations. These variations can be caused by either the deformation of the object or just by different object pose relative to the camera. In [24] the author proposes a new formulation of the two-dimensional (2-D) deformable template matching problem. It uses a lower-dimensional search space than conventional methods by pre-computing extensions of the deformable template along orthogonal curves. The reduction in search space allows the use of dynamic programming to obtain globally optimal solutions and reduces the sensitivity of the algorithm to initial placement of the template. An improved method for deformable shape-based object detection and segmentation is described in [25] where a pre-computed index tree is used to improve the speed of deformable template fitting. Simple shape features are used as keys in a pre-generated index tree of model instances. A coarse to fine indexing scheme is used at different levels of the tree to further improve speed.

Multi-scale representation

An inherent property of objects in the world is that they only exist as meaningful entities over certain ranges of scale. If one aims at describing the structure of unknown real-world signals, then a multi-scale representation of data is of crucial importance. Whereas conventional scale-space theory provides a well-founded framework for dealing with image structures at different scales, this theory does not directly address the problem of how to select appropriate scales for further analysis. In [26] the author outlines a systematic methodology of how mechanisms for automatic scale selection can be formulated in the problem domains of feature detection and image matching (flow estimation), respectively. For feature detectors expressed in terms of Gaussian derivatives, hypotheses about interesting scale levels can be generated from scales at which normalized measures of feature strength assume local maxima with respect to scale. It is shown how the notion of γ-normalized derivatives arises by necessity given the requirement that the scale selection mechanism should commute with rescalings of the image pattern. Specifically, it is worked out in detail how feature detection algorithms with automatic scale selection can be formulated for the problems of edge detection, blob detection, junction detection, ridge detection and frequency estimation. A general property of this scheme is that the selected scale levels reflect the size of the image structures. When estimating image deformations, such as in image matching and optic flow computations, scale levels with associated deformation estimates can be selected from the scales at which normalized measures of uncertainty assume local minima with respect to scales. It is shown how an integrated scale selection and flow estimation algorithm has the qualitative properties of leading to the selection of coarser scales for larger size image structures and increasing noise level, whereas it leads to the selection of finer scales in the neighborhood of flow field discontinuities (see [27, 28, 29]). Recent applications of this theory attempt to map the *real world* information (for example in the case of *known size* objects) into the scale parameter for automatic feature detection [30]. In [31] the author present a method where the scale of the filtering and feature detection is varied locally according to the distance to the scene pixel, which they estimate through stereoscopy. The features that are detected are, thus, at the same scale in the world, rather than at the same scale in the image. That method has been implemented efficiently by filtering the image at a discrete set of scales and performing interpolation to estimate the response at the correct scale for each pixel.

2.2 Tracking and motion-based approaches

The simplest tracking scenario involves a static observer (camera) that observes a static scene. The goal of tracking in this case is to detect and follow the moving objects that appear in the scene. Object detection and tracking are closely related because tracking usually starts with detecting objects, while detecting an object repeatedly in subsequent image sequence is often necessary to help and verify tracking. During the last twenty years, a great variety of object detection and tracking algorithm have been proposed, some examples are [32, 33, 34, 35, 36, 37, 38, 39]. The difficulty level of this problem highly depends on how you define the object to be detected and tracked. Motion-based approaches rely on robust methods for grouping consistent visual motion over time. These methods focus on recovering either the complete 3-d motion from the optical flow field or 2-d motion models [40, 41, 42, 43, 44, 45]. Two main approaches have been considered: *Boundary-based methods*, namely edge-based rely on the information provided by the tracked object boundaries while *Region-based approaches* rely on information provided by the entire tracked region like intensity, texture and motion-based proprieties.

Boundary-based tracking Methods

Boundary-based tracking algorithms employ active contour models, like snakes-balloons models [46, 47, 48] which are a forerunner to a large variety of physic-based tracking approaches [40, 49, 50, 51, 52, 53, 54]. These models lie on a flexible curve with some internal stiffness properties that ensure regularity. Some of the most important snake-based tracking approaches are the following. The authors in [55, 56, 57] have proposed a parametric B-spline curve. This model makes use a set of control points and set of basis elements to represent the evolving curve and allows many degree of freedom for stable tracking. These B-splines snakes are used to localize and track occluding contours of 3-D objects. In [58] the authors have presented an application of active contours to the tracking of points in anatomical structures in time sequences of ultrasound medical images. They use active deformable models to track the global structure. In [49] the authors have introduced a more efficient method for feature search for B-spline contours, whereby image edges are sought along contour using a divide-and-conquer strategy. The obtained framework can run at video rates and is used for surveillance of people and vehicles. The authors in [52] have reformulated the snake dynamics within a probabilistic framework and have made a link with another standard tool used in tracking, the Kalman filtering. The obtained framework integrates information from multiple sources (sensor fusion) and introduces a new sequential motion estimator, the Kalman Snake that uses the snake dynamics to constrain and predict possible motions.

Region-based tracking Methods

The region-based methods use a motion estimation-segmentation technique. In this case, the estimation of the velocity of the target is based on a correspondence between the associated target regions at different time instants. This correspondence is usually established using either the full motion field or parametric motion models, using the Markov Random Field framework. In [59] the author has proposed which combines intensity and motion information to segment incrementally a sequence of images. This approach employs local constraints on intensity and motion and is implemented using an incremental stochastic minimization technique that results in a robust and dynamic segmentation paradigm. Instead, the authors in [60] have proposed a MRF-based multi-resolution approach which handles discontinuities and occlusions of the motion field. This method integrates several information sources like velocity vectors, motion boundaries, intensity edges, occluded surfaces, etc. Furthermore this approach makes use of multiple hypotheses resulting on a model that preserving discontinuities, can deal with the occlusion cases, etc. In [45] the authors have proposed a region-based tracking algorithm that exploits the output of a motion-based segmentation process. Within this framework, two interacting filters are supposed; the region shape and position is update using a geometric filter, while the corresponding region motion parameters are estimated recursively with the aid of a motion filter. In this approach first order linear motion models are assumed. The authors in [61] have proposed a method that combines motion and intensity-based information to detect and segment multiple moving objects. This approach consist in two steps: the first relies on the estimation of the apparent motion using robust regression techniques while the second on the integration between the motion estimates and the intensity information to perform motion detection-segmentation. However, these models have difficulty in tracking the target boundaries successfully but they are robust due to fact that they make use of information provided by the whole region and do not depend critically on the initialization step.

Combining Boundary and Region-based Tracking Methods

In [62] the authors have integrated the snake-based contour tracking with the region-based motion analysis. Initially, a snake tracks the region of interest and performs segmentation. Then, the region motion parameters are estimated using the spatio-temporal image gradients. Additionally, this motion is filtered to predict the region location over. In [40] the authors have proposed a tracking approach that combines deformable region models and deformable contours. This model is based on texture correlation and is constrained by the use of a motion model, such as the rigid, the affine or the homographic. The author in [63] have proposed an elliptical model that tracks human faces by combining boundary and region-based information. This model optimizes the ellipse parameters as well as its position by using two othonormal modules; one that is based on the intensity gradient around the head perimeter and another that is based on the color histogram of the head interior. The authors in [64] have proposed an active region model for frame-rate tracking in color images. They constrain the deformable snake model using one of the rigid, conformal, or affine motion models. According to this method, the initial curve is propagated towards the solution under the influence of boundary information, motion measurements and region statistics.

Chapter 3

Exploiting the prior information

In this chapter we describe which are the sources of the *a-priori* information and how we apply this information in our algorithm. The chapter is structured as follows:
in section 1 we motivate and describe the use of the background/foreground partition. We discuss in section 2 the motivations and the advantages related the use of a monocular calibrated camera and the computation of the top-view projection is described. In section 3 the resizing of the patch according to the scale of the pedestrian head is shown. Section 4 deals with the definition of the head shape models used in the system. A definition of the gaussian/elliptic models, gradient-maxima constraints and curvature-based models are given as well as the description of their main characteristic.

3.1 Background/foreground modeling

The technique selected for background/foreground modelling depends on the application. In that cases where robustness against illumination changes is required the background should be statistically modelled. These kinds of techniques model the image pixels separating brightness from chromaticity. This allows to classify pixels into background (similar chromaticity and brightness as the original background), shadowed background (similar chromaticity as the background but lower brightness), highlighted background (similar chromaticity as the background but higher brightness) and foreground (different chromaticity as the background). Selecting the appropriate thresholds, which can be obtained automatically using statistical learning procedures, background and foreground regions can be robustly distinguished [65]. For less demanding applications, as for example the ones inside buildings that have usually the same environmental conditions, the easiest way to achieve the partition background/foreground is subtracting a reference background image from the original one. This implies that we have a background image. This picture has to be taken under the same conditions in order to achieve good foreground approximation. In our case we adopt the second solution. This choice is dictated uniquely by the simplicity of implementation. We are aware of the approximation introduced by this method and we reserve the statical background modelling for future developments.

3.2 Camera calibration and top-view representation

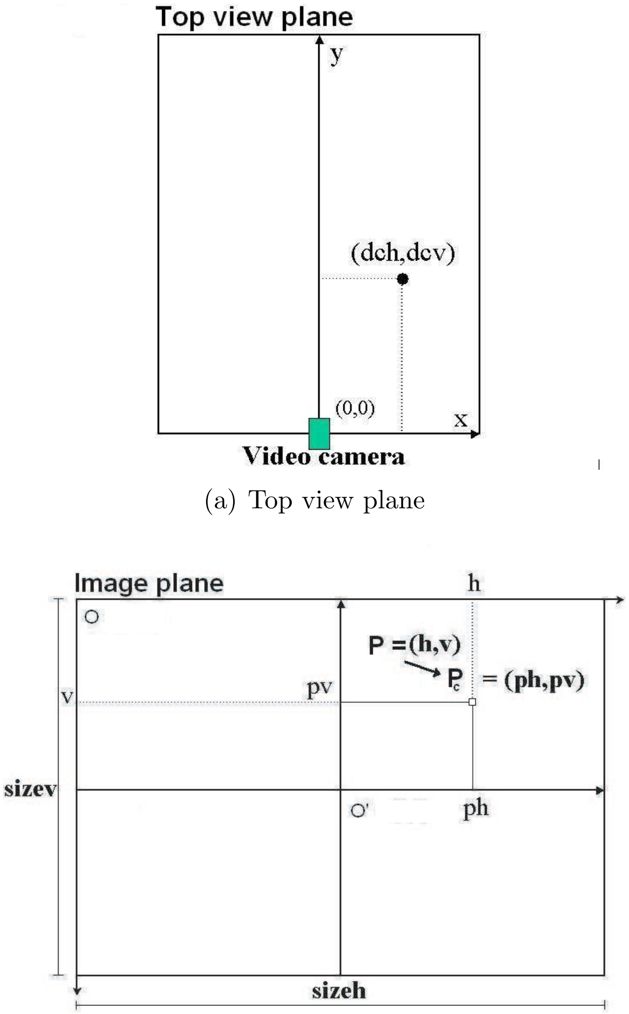
Camera calibration

Camera calibration is the process which allows to determine the value of the parameters vector $C_p = [fd, h_{\text{camera}}, \alpha, \varphi]$, where fd is the focal length of the lens, h_{camera} is the height of the camera's optical center, φ is the camera's tilt angle around the vertical axis and α is the camera's roll angle around the horizontal axis. This process consist in solving a non-linear system of the equations with respect to the camera parameters. The problem is not trivial and different solutions have been proposed in literature [66],[67],[68]. In our method, we use a quadrilateral of known dimensions in the floor plane. The four sides and the two diagonals of this quadrilateral give us six non linear equations in the four unknowns fd,h_{camera},α and φ, thus an over-determined system [1](11 equations).
To solve this system, we transform it into a leats-squares problem, which is a minimization problem that we solve using Newton's method.

Top-view representation

The top-view plane correspond to the view we obtain placing the camera on the top of the scene. The importance of working with the top view resides in the fact that occlusions are avoided on this plane. We do not have in fact the miss perception of the depth given by the perspective effect. Moreover, it corresponds to the real walking plane, where people generate trajectories. Knowing the camera parameters allows us to have a unique correspondence

[1] The calibration algorithm has been developed by Mats Weber, Operation Research, EFPL, Lausanne



(a) Top view plane

(b) Image plane

Figure 3.1: Top view plane and image view plane.

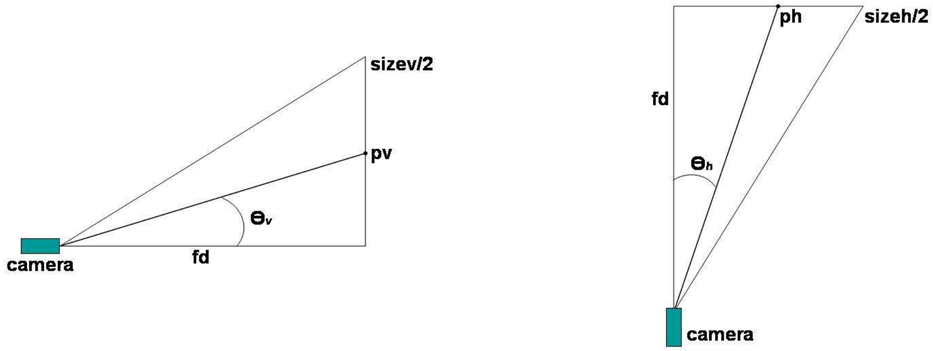


Figure 3.2: Angles θ_v and θ_h from pixel coordinates.

between the image plane (see figure 3.1,(a)) and the top view plane (see figure 3.1,(b)). It means that we can compute the real world coordinates of a given pixel $p = (h, v)$ (see figure 3.1,(b)).

The first step is the translation of the origin of the coordinates system, from o to o', as shown in figure 3.1(a). In this new system the $p = (h, v)$ becomes $p_c = (ph, pv)$, where:

$$pv = \frac{\text{sizev}}{2} - v \tag{3.1}$$

$$ph = h - \frac{\text{sizeh}}{2} \tag{3.2}$$

where sizev and sizeh are the dimensions of the original image (see figure 3.1,(b)).
Using the camera parameters we obtain the equivalent angles θ_v and θ_h (see figure 3.2).
According to the camera focus, for the point p_c, we have:

$$\theta_v = arctg\left(\frac{pv}{fd}\right) \tag{3.3}$$

$$\theta_h = arctg\left(\frac{ph}{fd}\right) \tag{3.4}$$

Now we can compute the top-view coordinates (dch, dcv) (see figure 3.3) as:

$$dcv = hc \cdot tan(\alpha + \theta_v) \tag{3.5}$$

$$dch = d \cdot tan(\theta_h) \tag{3.6}$$

where

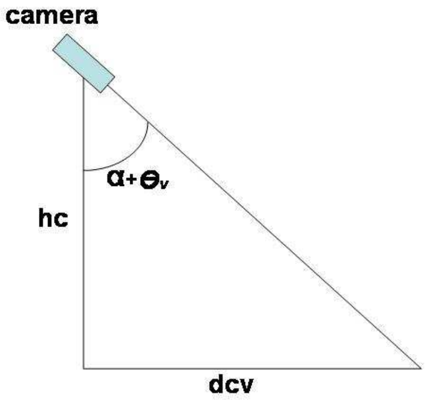


Figure 3.3: Camera position and angle.

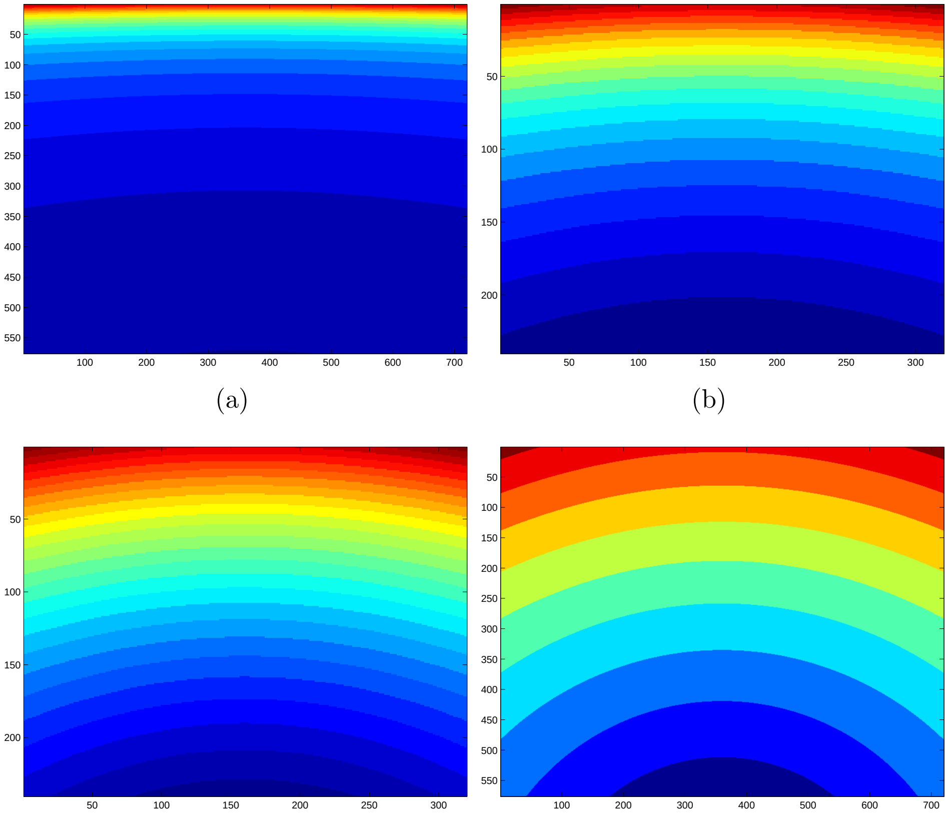
$$d = \sqrt{dcv^2 + hc^2)} \tag{3.7}$$

For each point (x, y) on the image we can then compute the corresponding distance on the top view, where the origin of the reference system is represented by the projection of the real 3D camera position on the top-view itself. This procedure lives rise to a distance map for the image at hand illustrated in figure 3.4. This map is similar to the range data obtained in stereo vision [31].

3.3 Patch resizing

To analyze the foreground pixels to check for pedestrian heads we crop a squared image patch centered around the current pixel position, and we want the patch being centered on the head. The size of the patch changes according to the scale of the head, which is a function of the real distance between the camera and the pedestrian. In order to perform patch resizing we define the following parameters (see figure 3.5):

$$\begin{aligned} h_{\text{down}} &= h_{\text{camera}} - (h_{\text{men}} - h_{\text{head}}/2 - h_{\text{window}}/2) && (3.8)\\ h_{\text{c}} &= h_{\text{down}} - h_{\text{window}}/2 && (3.9)\\ h_{\text{top}} &= h_{\text{c}} - h_{\text{window}}/2 && (3.10) \end{aligned}$$



(a)

(b)

(c)

(d)

Figure 3.4: Different distance maps for different camera parameter vectors. We note the curves representing the top-view points at the same distance from the origin. The color represents the distance value.

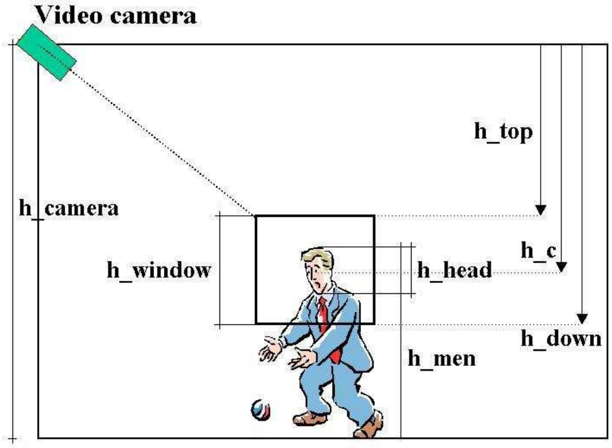


Figure 3.5: Parameters related to the real head size.

These parameters resume our prior knowledge. We assume $h_{\text{men}} = 1.70cm$, h_{head} refers to the real human head size and is assumed to be 20 cm, as described in section 3.4. Following similar arguments as those in section 3.2 and by means of the parameters defined in equation 3.3, we can re-project an hypothetic pedestrian position (dch, dcv) from the top-view back on the image plane.

We compute the angles θ_v^{top} and θ_v^{down} as follows:

$$\theta_v^{\text{top}} = arctg\Big(\frac{dcv}{h_{\text{top}}}\Big) - \alpha \tag{3.11}$$

$$\theta_v^{\text{down}} = arctg\Big(\frac{dcv}{h_{\text{down}}}\Big) - \alpha \tag{3.12}$$

Then we can obtain pv_{top} and pv_{down}:

$$pv_{\text{top}} = fd \cdot tan(\theta_v^{\text{top}}) \tag{3.13}$$

$$pv_{\text{down}} = fd \cdot tan(\theta_v^{\text{down}}) \tag{3.14}$$

and finally v_{top} and v_{down}:

$$v_{\text{top}} = \frac{\text{sizev}}{2} - pv_{\text{top}} \tag{3.15}$$

$$v_{\text{down}} = \frac{\text{sizev}}{2 - pv_{\text{down}}} \tag{3.16}$$

The correct size of the patch is then calculated as

$$h_{\text{window}} = v_{\text{top}} - v_{\text{down}} \tag{3.17}$$

Having an image patch which adapts its size to the pedestrian head, taking into account the perspective deformation, is an useful method (see figure 3.6). The prior knowledge is maximally exploited and it allows us to avoid the formulation of more complex deformation models.

3.4 Head shape models

We use the prior knowledge on the human head shape to define multi-scale shape models. In all the used models, we assume that the head size and the patch size are proportional to each other:

$$h_{\text{head}} = \frac{h_{\text{window}}}{3} \tag{3.18}$$

$$w_{\text{head}} = \frac{h_{\text{window}}}{4} \tag{3.19}$$

where h_{head}, w_{head} and h_{window} are illustrated in figure 3.7. We describe in the following the models that we use:

Gaussian shape model

This first model is based on Gaussian shape,

$$f(x) = \begin{cases} N(\mu, \sigma^2) & \text{per } a \leq x \leq b \\ 0 & \text{otherwise} \end{cases}$$

The standard deviation σ is directly related to the head width and the μ parameter coincides with the $w_{\text{head}}/2$ position (see figure 3.7,3.8). The tails have been removed outside the interval $[a, b]$ (see figure 3.9(b)).

An image template is generated assuming the same size of the resize patch (equation 3.17). The pixel positions corresponding to the Gaussian shape have been set to 255 intensity value and all the other pixels are zero. This resulting mask is only an approximation of the real human head shape. To reduce the gap we diffuse the mask with a Gaussian discrete filter with variable length. The length of the discrete Gaussian filter should be related to the length of the filter derivative Gaussian used to do the edge extraction. A model of the thickness of the edge is needed. We use a set of discrete Gaussian filter and we test the candidate applying the head detector with a shape model diffused first with the shorter filter up to the positive result of the statistic test. The diffused mask is illustrated in figure 3.10.

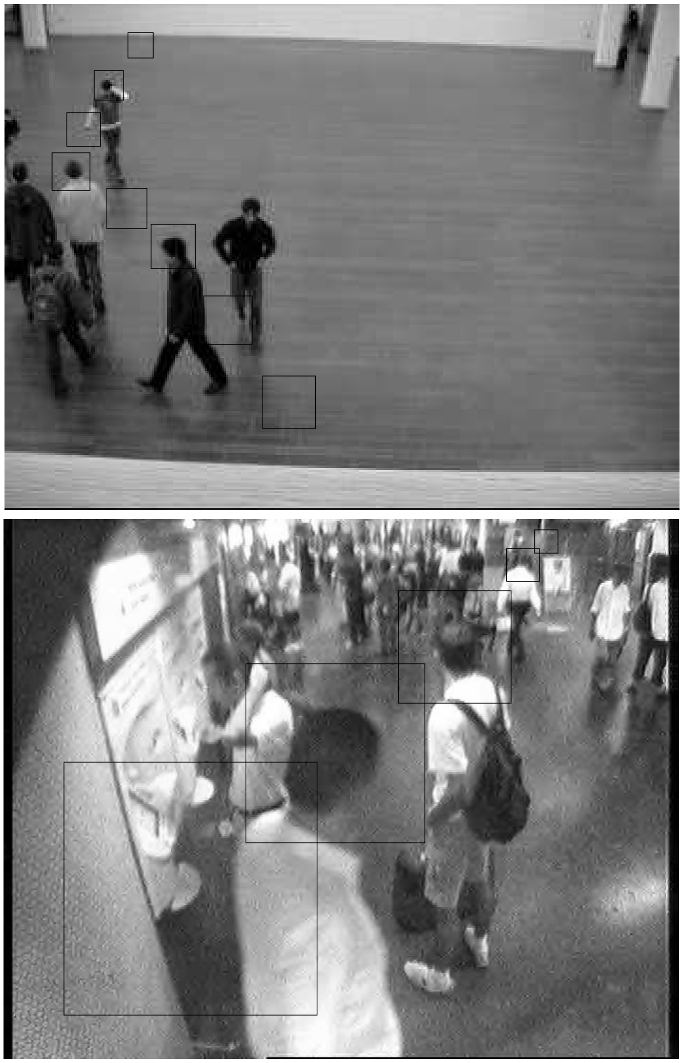


Figure 3.6: The size of the patch changes according to the perspective field.

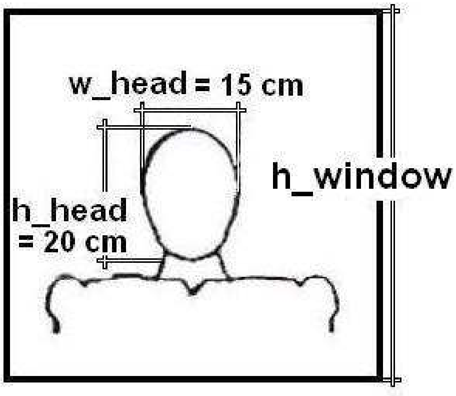


Figure 3.7: The head model is created according to the real human head size. Height and width of the head are linked to the patch size.

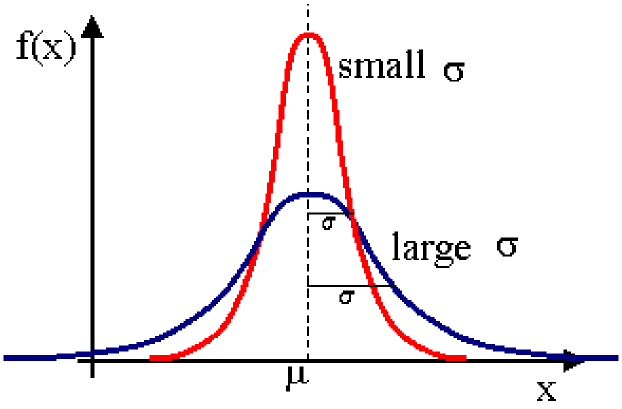
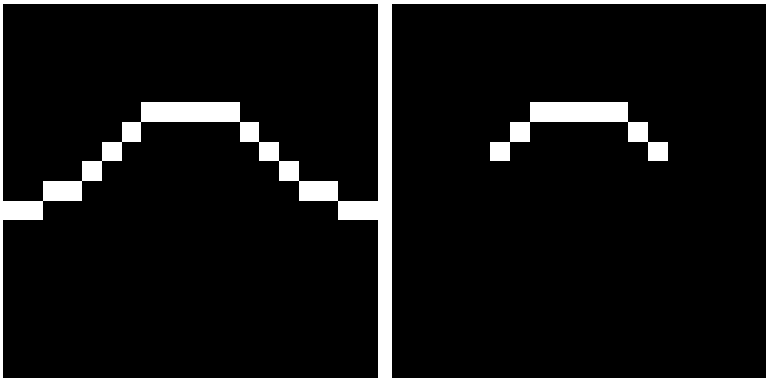


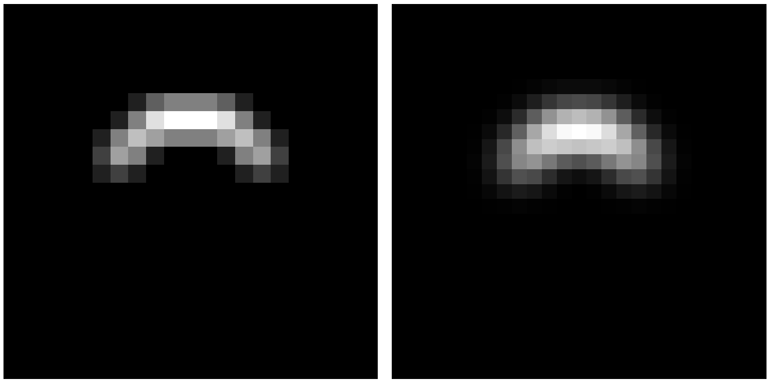
Figure 3.8: Gaussian function.



(a) The centered Gaussian shape.

(b) Gaussian model without the tails.

Figure 3.9: The first model used.



(a) Gaussian model diffused with a Gaussian filter with 3 coefficients.

(b) Gaussian model diffused with a Gaussian filter with 7 coefficients.

Figure 3.10: The first model used.

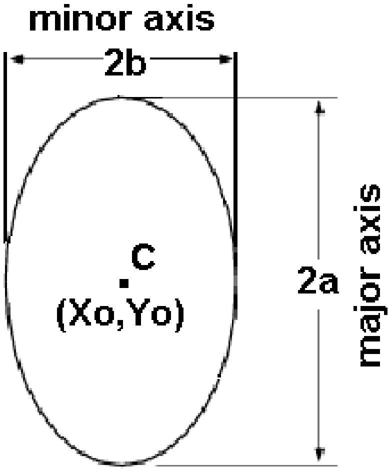


Figure 3.11: Ellipse curve.

Elliptic shape model

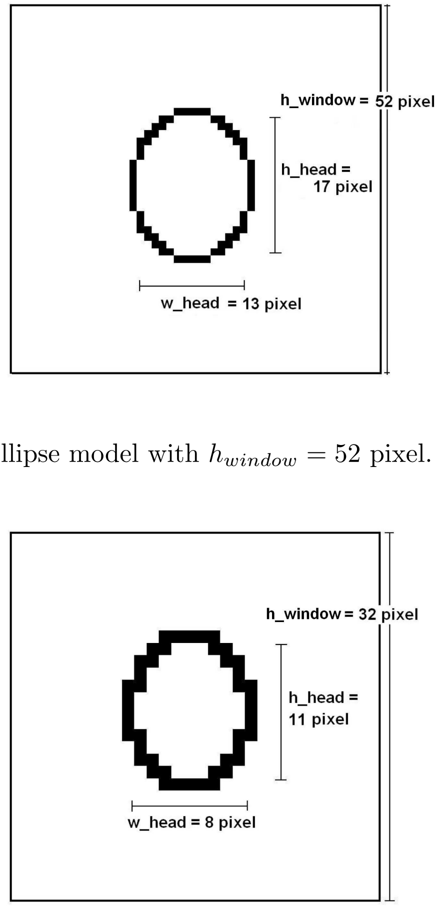
This is based on the ellipse illustrated in figure 3.11 :

$$\frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2} = 1 \tag{3.20}$$

We construct the template centering the ellipse in the resized patch where a,b directly depend on the real head size with $2a = h_{\text{head}}$ and $2b = w_{\text{head}}$ as shown in figure 3.12.
Following the same steps as for the Gaussian model we obtain the diffused elliptic mask, as shown in figure 3.13. We illustrate the use of the model in the section (see section 4.4) and (see section 4.5).

Gradient maxima constraints

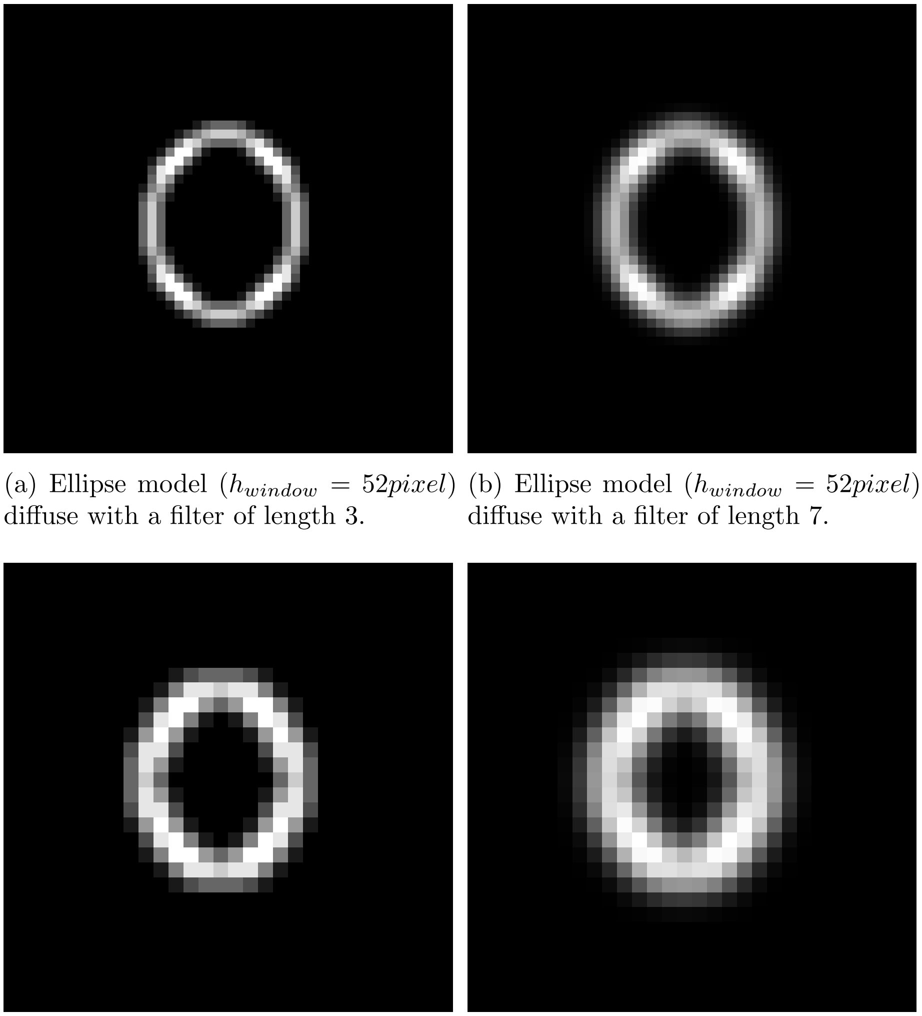
We use the vertical symmetry of the Gaussian model and both the horizontal and vertical symmetries of the elliptic model. Assuming to have already obtained the edges from the resized patch (see section 4.3), we refine the shape models constraining the edge map. We analyze the luminance values



(a) Ellipse model with $h_{window} = 52$ pixel.

(b) Ellipse model with $h_{window} = 32$ pixel.

Figure 3.12: Examples of elliptic models for two patches with different size.



(a) Ellipse model ($h_{window} = 52pixel$) diffuse with a filter of length 3.

(b) Ellipse model ($h_{window} = 52pixel$) diffuse with a filter of length 7.

(c) Ellipse model ($h_{window} = 32pixel$) diffuse with a filter of length 3.

(d) Ellipse model ($h_{window} = 32pixel$) diffuse with a filter of length 7.

Figure 3.13: Examples of elliptic models diffuse with Gaussian filter with 3 and 7 coefficients.

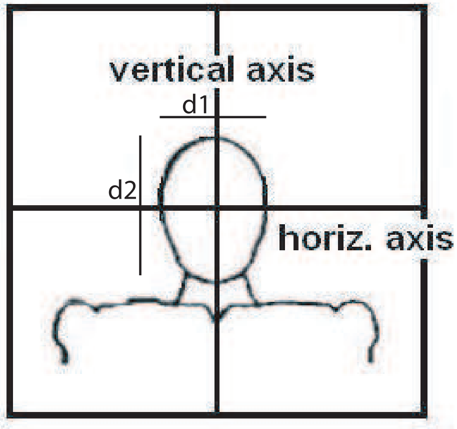
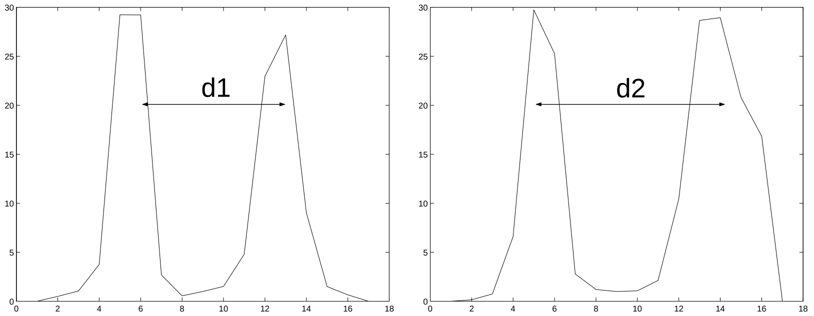


Figure 3.14: Axes of symmetry.



(a) Maxima value in the horizontal axis with $d_1 = w_{\text{head}}$.

(b) Maxima value in the vertical axis with $d_2 = h_{\text{head}}$.

Figure 3.15: Maxima intensity value horizontal and vertical considering the edge map.

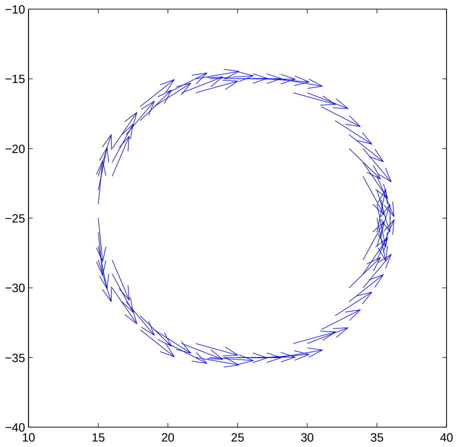
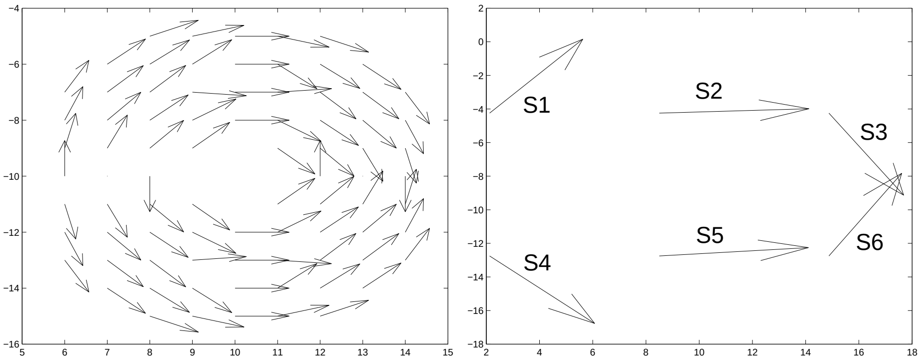


Figure 3.16: The angles of the tangent directions to each point of a circular shape, is uniformly distributed over the range $[-\pi/2, \pi/2]$.

of the edge map along the vertical and horizontal axes as illustrated in figure 3.14. The intensity maxima on these directions have to fit the distance constraints imposed by the elliptic (or Gaussian) shape models (see figure 3.15).

Curvature-based models

The last two models are based on statistical considerations on the curvature of the head shape. Let's consider the circular shape in the first model. If we look at the tangent vector t to each point on the circle (see figure3.16), we note that the angle formed by t and the horizontal direction is uniformly distributed over the range $[-\pi/2, \pi/2]$. So, we can describe the circular shape by an uniform distribution related to the direction of the tangent vector to the support of the circle. In the case of the elliptic shape we have a certain "distortion" from the uniform distribution, depending on the ellipse eccentricity. In this analysis we ignore the error introduced by this approximation. The other model is a refinement an the curvature analysis. As for the gradient maxima constraints, we assume to dispose of the edge map for the current resize patch. Moreover, we assume also to have computed the Gradient Direction Map (GDM, see section 4.5). We build the GDM associating to each position in the edge map the curvature of the gradient at that position, expressed in radians. We divide the GDM in six regions (upper-left, upper-center, upper-right, lower-left, lower-center, lower-right). We simply constraints the average-curvature values on each of these regions



(a)

(b)

Figure 3.17: GDM and the six regions.

(S1, S2, S3, S4, S5, S6) as follows:

$$\text{Upper part} = \begin{cases} 0 < S1 < \pi/2 \\ -\pi/6 < S2 < \pi/6 \\ -\pi/2 < S3 < 0 \end{cases}$$

$$\text{Lower part} = \begin{cases} -\pi/2 < S4 < 0 \\ -\pi/6 < S5 < \pi/6 \\ 0 < S6 < \pi/2 \end{cases}$$

Chapter 4

Head detector

In this chapter we describe the head detector. All the models defined in chapter 3 are combined. In section 4.1 we give an overview of the head detection algorithm. The head detection is performed on the edge map of a given image patch. Section 4.2 and 4.3 describe the automatic scale selection for edge detection. A simple template head matching approach is described in section 4.4 and its limits are discussed. In section 4.5 we explain a statistical-based shape analysis and in section 4.6 is implemented a clustering algorithm for the candidates overlapping on the same pedestrian head.

4.1 System overview

An overview of our system is shown in figure 4.1. Given an input image I and assuming to have the background image B we analyze only those patches centered around those pixels lying on the foreground, obtained by background subtraction. The *a-priori* information about the camera parameters and the head shape are combined to automatically select the right scale for edge detection (see section 4.3). Similarly, the shape models are used to generate the correct region of interest (ROI). Inside our ROI a Gradient Direction Map (GDM) is extracted associating the orientation of the edge with each position in the edge map (EM). The advantage of using the GDM against the simple EM is that GDM is illumination invariant and directly related to the shape information. The detection is performed using goodness-of-fit tests (see section 4.5). We assume infact that for an ideal head shape the histogram of its GDM follows a uniform distribution. We use both the Chi-Square and the Kolmogorov-Smirnov tests to compare the candidate GDM distribution with the reference one. After goodness-of-fit test has been performed, those patches which are positive are spatially clustered. This is justified by the fact

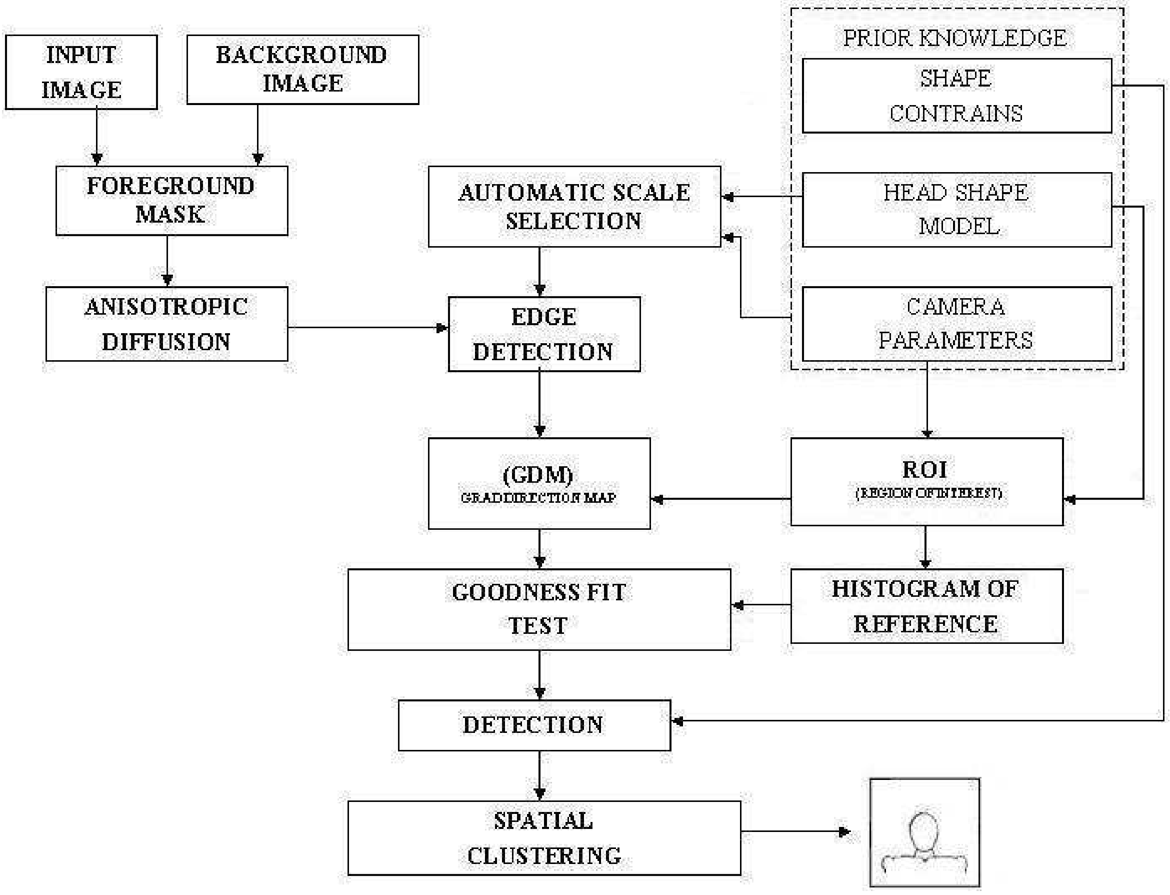


Figure 4.1: Head detector scheme.

that several positive patches overlap on the same pedestrian head. We have compared this method with a template-based matching between the EM and the shape model. The drawback of this simple approach is that it requires threshold tuning. The selection of the threshold is strongly dependent on the image at hand.

4.2 Automatic scale selection

Objects in real world appear differently, based on their scale. Image structures as well assume different characteristics based on their distance in the real world. Linear scale space theory represents a well known theoretical framework to tackle this problem. The general theory recognizes the importance of a scale-space image representation but does not provide a general means to automatic select the right scale for each position, based on the known object size. In our approach we use the calibrated monocular camera to obtain the depth information on the top-view plane and to select the right scale for extract the edges. A similar approach is used in [31], where the authors stereo vision to obtain the range data information. We chose the scale to perform edge detection based on the known object size and the top-view distance information. The right scale to be used at (x, y) is given by:

$$\sigma(x,y) \;=\; f(C_p, O_s) \tag{4.1}$$

where $\sigma(x, y)$ is the scale, C_p is the camera parameters vector and O_s is related to the known object size. We explicit this equation:

$$\sigma(x,y) \;=\; \log h_{\text{window}} - K \tag{4.2}$$

where h_{window} is the estimated patch size given for each point (x, y) by the equation 3.17, and K is a constant value depending on the object size. In our case we fix its value to $\log 3$. In other words, the scale value corresponds to the log value of the head size as it appears on the image plane, opportunely resized taking into account perspective effects. Since the range of scales that we are concerned with may be very large and in [69] the author has shown that a logarithmic sampling of the scale space is stable and so we work in the log domain.

4.3 Edge detection

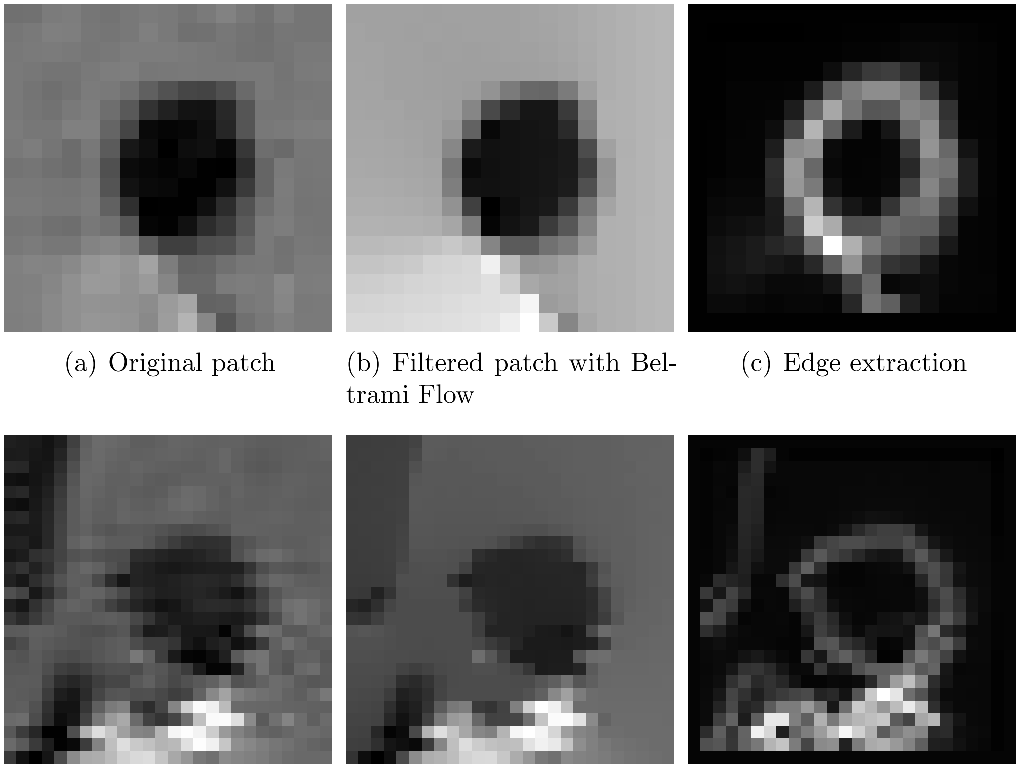
Pre processing

The anisotropic diffusion problem can be seen as a more general problem well known in m. and ph.: the generalized minimal surface computation. Given an object, we want to find the embedding surface having the minimal energy. The problem is formulated in [70] as a minimization problem (known as the Polyakov action), where the objective functional depends on the mapping between the object and the embedding surface and on the metric induced by the mapping itself. In our context, if we define the standard euclidian metric as :

$$g = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

The minimization problem leads to the Gaussian solution (isotropic diffusion). On the contrary taking the metric g based on the image gradient:

$$g = \begin{pmatrix} 1 + I_x^2 & I_x I_y \\ I_x I_y & 1 + I_y^2 \end{pmatrix}$$



(a) Original patch (b) Filtered patch with Beltrami Flow (c) Edge extraction

(d) Original patch (e) Filtered patch with Beltrami Flow (f) Edge extraction

Figure 4.2: Examples of filter patch with Beltrami Flow.

where I is a grey-level image, the result of the Polyakov action (minimized respect to I) is the Beltrami-Flow. It can be shown that the action of this kind of flow between the original image (thought as a generalized surface) and the diffused image (the "minimal" surface) is edge preserving (see [71],[72]). For this reason we use it as a pre-processing step (see figure 4.2).

Edge computation

An edge occurs where there is a discontinuity in the intensity function or a very steep intensity gradient in the image. Using this fact, if we take the derivative of the intensity values across the image and find points where the derivative is maximum, we will have marked our edges. We can calculate the gradient by simply taking the difference of grey values between adjacent pixels. This is equivalent to convolving the image with the mask $[-1, 1]$. Note that we have a problem in determining where we place the result of the

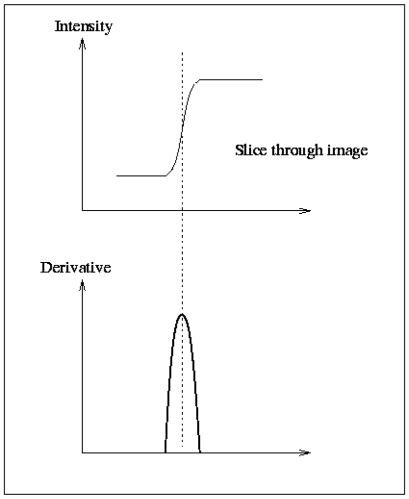


Figure 4.3: An ideal step edge and its derivative profile.

convolution. Ideally, we would like to place it between the two pixels. By using a mask that spans an odd number of pixels we end up with a middle pixel to which we can assign the convolution result. The gradient of the image function I is given by the vector:

$$\nabla I = \left[\frac{\partial I}{\partial x} + \frac{\partial I}{\partial y}\right] \tag{4.3}$$

The magnitude of this gradient is given by $\sqrt{(\frac{\partial I}{\partial x})^2 + (\frac{\partial I}{\partial y})^2}$ and its direction by $\arctan\left\{\frac{h_y \otimes a}{h_x \otimes a}\right\}$.
Note, one can use any pair of orthogonal directions to compute this gradient, although it is common to use the x and y directions. There are some kinds of gradient operators:
Robert's Cross operator, Prewitt operator, and Sobel operator among others.

Filter coefficient

We have to deal with the problem of the object scales and we can improve the edge extraction if we know the right scales of the object in the image. We apply a first derivative Gaussian filter with a variable length according to the right scale. We have to work with discrete data so a discrete version of the Gaussian filter is needed (Binomial filter). The relationship between the right scale σ (as obtained in equation 4.2) and the length L of the discrete filter is derived based on Hermite polynomial theory [73]. The discrete Hermite transform of length M approximates the analog Hermite transform of spread

$$\sigma \quad = \quad \sqrt{M/2} \tag{4.4}$$

where $M = L+1$ and M is the order of the binomial coefficient. According to the orthonormal Hermite polynomial expansion theory, given a 2D discrete signal, it can be represented with an arbitrary approximation by means of the discrete Hermite transform of length M. The coefficients of this expansion are generated by the following filter :

$$D_n(x) \quad = \quad G_n\Big(\frac{M}{2} - x\Big) \cdot V^2\Big(\frac{M}{2} - x\Big) \tag{4.5}$$

for $x = -(M/2), \ldots, (M/2)$ where G_n is the polynomial based function

$$G_n(x) \quad = \quad \frac{1}{\sqrt{C_M^n}} \sum_{k=0}^{n} (-1)^{n-k} \cdot C_{M-x}^{n-k} \cdot C_x^k \tag{4.6}$$

for $x, n = 0, \ldots, M$. $V^2(x)$ is the binomial window function:

$$V^2(x) \quad = \quad \frac{1}{2^M} \cdot C_M^x \tag{4.7}$$

for $x = 0, \ldots, M$ where C_M^x is the binomial coefficient of length M.
The order n of the filter D_n represents the derivative order. We are interested in those filter coefficients generated assuming n = 1. We report more details about the Hermite transform theory in the appendix.

4.4 Template based matching

Template matching is a generic approach to find similarity between a generic image and template. We can use it to search for specific patterns (see figure 4.4). A standard similarity measure for template matching is represented by the correlation function. Given two images f and g of size MxN, the 2D discrete correlation between them is defined as:

$$C(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) g(x + m, y + n) \tag{4.8}$$

for $x = 0, 1, ..., M - 1$ and $y = 0, 1, ..., N - 1$.
Normally, the best matching position is taken corresponding to the maximum

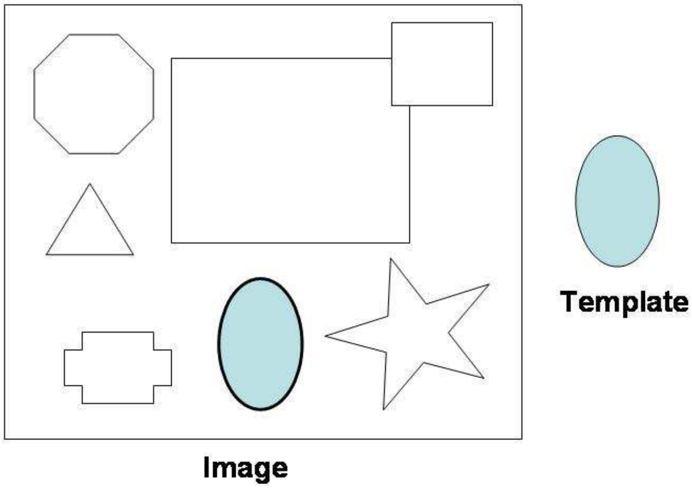


Figure 4.4: Example of matching.

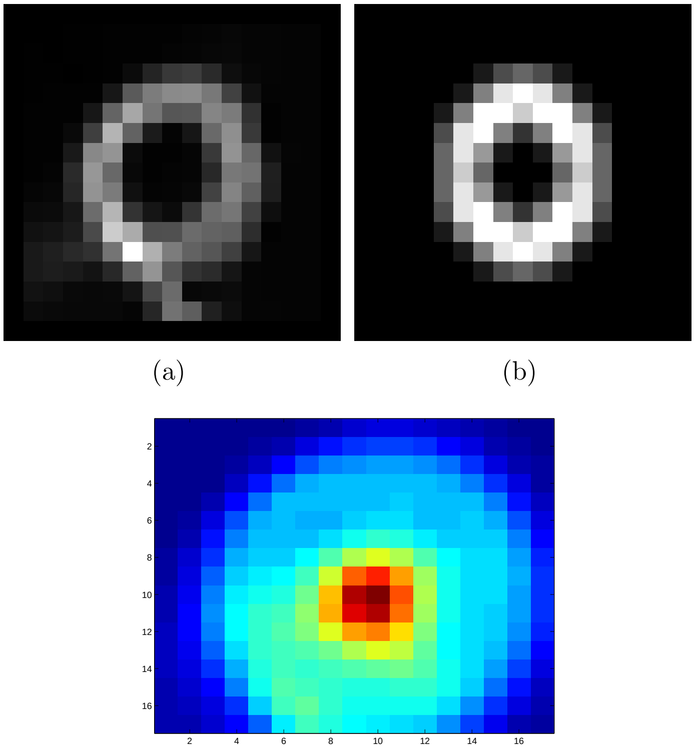
of the correlation function.

A way to implement the correlation formula is using the convolution theorem. The correlation between a pattern (M) and image (I) can be determined by

$$Corr = \mathrm{FFT}^{-1}(\mathrm{FFT(I)} \times \mathrm{FFT}^{*}(\mathrm{M})) \tag{4.9}$$

where FFT is the Fats Fourier Transform and $\mathrm{FFT}^{*}(M)$ is the complex conjugate of the Fast Fourier transform of the pattern.
The correlation result will be higher at those positions where the image structure matches the mask structure (see figure 4.5).

As already said previously in the thesis, we want the patch being centered on the pedestrian head. So, we impose two constraints on the maximum-of-correlation value:
the first one is that the coordinates of the maximum have to be in the center of the patch (or in a neighborhood opportunely defined) and the second one is that the maximum value must be higher than a certain threshold opportunely tuned. If both conditions are verified we accept the potential candidate as good. The problem of this method is that it requires threshold tuning. Unfortunately the selection of the threshold is strongly dependent on the image at hand.



(a) (b)

(c)

Figure 4.5: (a) Edge map of the current patch, (b) Diffuse mask, (c) Correlation result.

4.5 Statistical edge curvature analysis

In order to overcome the threshold-tuning problem and to obtain illuminance invariance we approach the head detection using a statistical description of the head shape. Given an image a, its gradient $\nabla a[m,n]$ is defined as:

$$\nabla a = \frac{\partial a}{\partial x}\vec{i}_x + \frac{\partial a}{\partial y}\vec{i}_y \tag{4.10}$$

$$= (h_x \otimes a)\vec{i}_x + (h_y \otimes a)\vec{i}_y \tag{4.11}$$

where $\vec{i}_x$ and $\vec{i}_y$ are units vectors in the horizontal and vertical direction, respectively. The gradient direction is computed as:

$$\psi(\nabla a) = \arctan\left\{\frac{h_y \otimes a}{h_x \otimes a}\right\} \tag{4.12}$$

where h_x and h_y are computed by convolving the image with the right filter (see section 4.3). The diffused mask f_m and the edge map EM_a of the current patch are combined to obtain the correct region of interest (ROI):

$$\mathrm{ROI} = (\mathrm{f}_m \,\mathrm{AND}\, \mathrm{EM}_a) \tag{4.13}$$

The Gradient Direction Map (GDM) over the ROI is defined as follows (see figure 4.6):

$$\mathrm{GDM}_{\mathrm{ROI}} = \Big\{\psi(\nabla a)(x,y), \forall (x,y) \in \mathrm{ROI}\Big\} \tag{4.14}$$

We compute the histogram of the $\mathrm{GDM}_{\mathrm{ROI}}$ values and we use goodness-of-fit tests to compare the current histogram with the reference histogram, corresponding to a uniform distribution, as defined in section 3.4.
The GDM is illumination invariant, corresponding to the edge curvature and it is directly related to the shape information. Moreover, the use of statistical tests (χ^2 or KS) is not dependent on a threshold value. This facts make the method robust and applicable in different scenarios.

Chi-Square Test

Definition The chi-square test is defined for the hypothesis: the data follow a specified distribution (H_0) and the data do not follow the specified distribution (H_a). For the chi-square goodness-of-fit computation, the data are divided into k bins and the test statistic is defined as

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i} \tag{4.15}$$

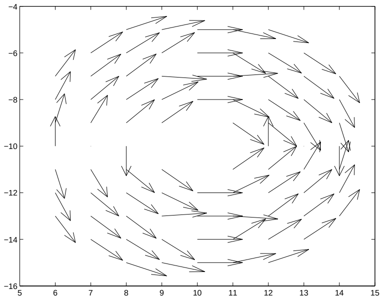


Figure 4.6: GDM defined by the gradient direction and the ROI.

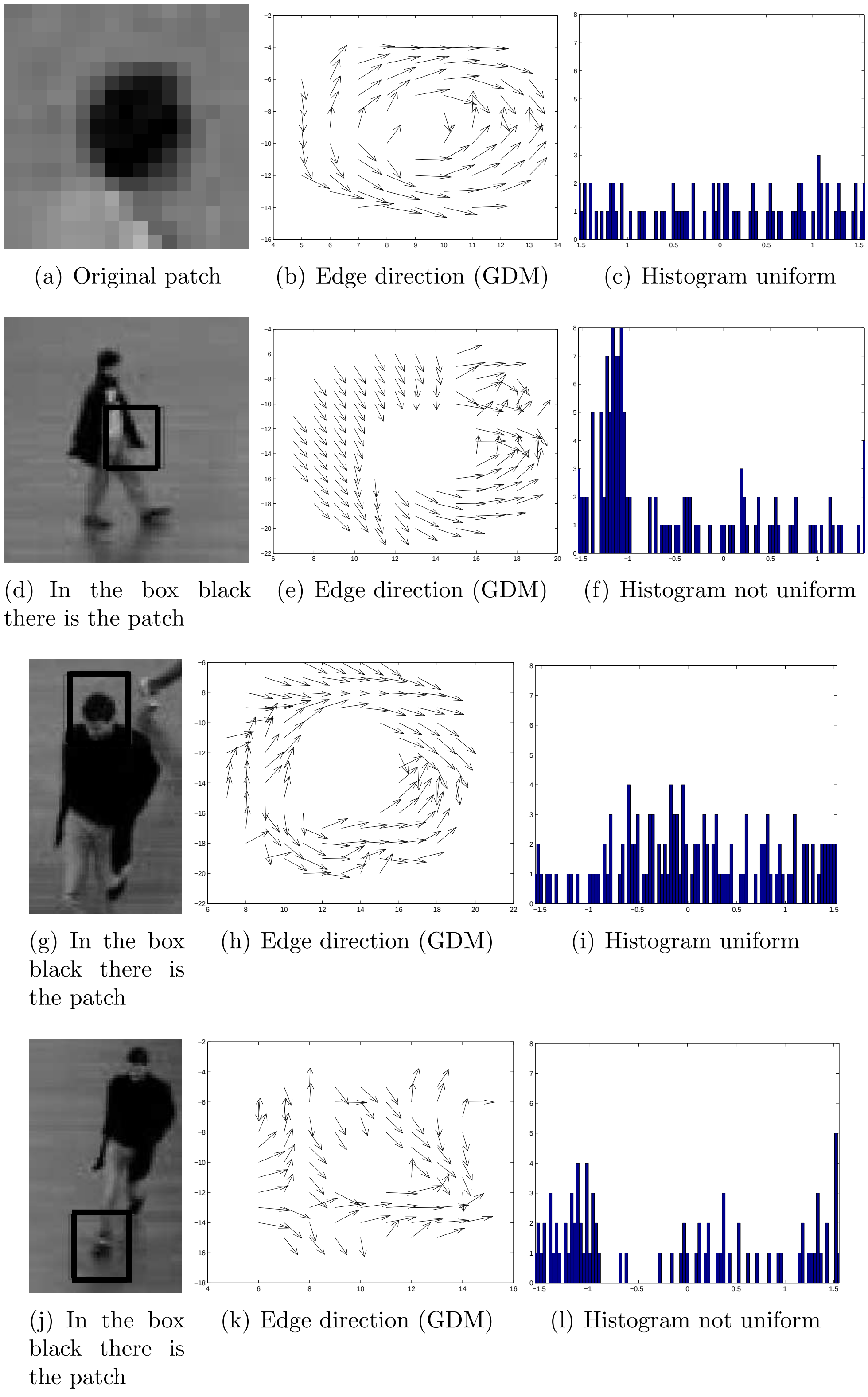
where O_i is the observed frequency for bin i and E_i is the expected frequency for bin i. The expected frequency is calculated by

$$E_i = N(F(Y_u) - F(Y_l)) \tag{4.16}$$

where F is the cumulative Distribution function for the distribution being tested, Y_u is the upper limit for class i, Y_l is the lower limit for class i, and N is the sample size.

The χ^2 test [74] is used to decide if a sample comes from a population with a specific distribution and in our case uniform distribution. It can be applied to any univariate distribution which you can calculate the cumulative distribution function. It is applied to binned data, i.e histograms. This is actually not a restriction since for non-binned data you can simply calculate a histogram or frequency table before generating the chi-square test. However, the value of the chi-square test statistic are dependent on how the data is binned. Another disadvantage of the chi-square test is that it requires a sufficient sample size in order for the chi-square approximation to be valid (it is difficult to use when the current patch contains just few edges). The actual number of observations in each bin is compared to the expected number of observations and the test statistic is calculated as a function of this difference. The number of bins and how bin membership is defined will affect the power of the test (i.e., how sensitive it is to detecting departures from the null hypothesis). Power will not only be affected by the number of bins and how they are defined, but by the sample size and shape of the null and underlying (true) distributions.

Critical Region The test statistic follows, approximately, a chi-square distribution with $(k-c)$ degrees of freedom where k is the number of non-empty



(a) Original patch

(b) Edge direction (GDM)

(c) Histogram uniform

(d) In the box black there is the patch

(e) Edge direction (GDM)

(f) Histogram not uniform

(g) In the box black there is the patch

(h) Edge direction (GDM)

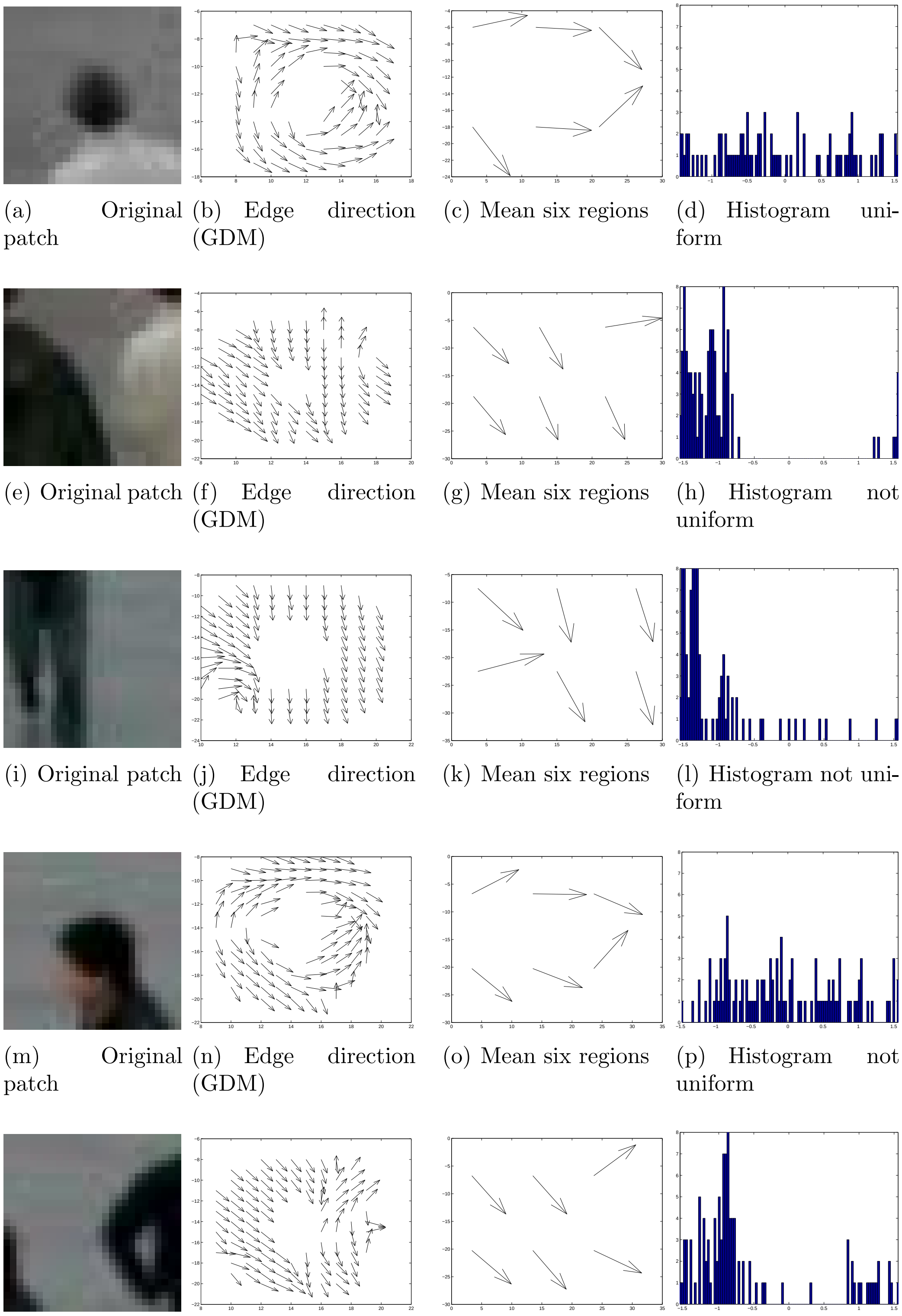
(i) Histogram uniform

(j) In the box black there is the patch

(k) Edge direction (GDM)

(l) Histogram not uniform

Figure 4.7: Comparison between rejected candidate and accepted candidate.



(a) Original patch (b) Edge direction (GDM) (c) Mean six regions (d) Histogram uniform

(e) Original patch (f) Edge direction (GDM) (g) Mean six regions (h) Histogram not uniform

(i) Original patch (j) Edge direction (GDM) (k) Mean six regions (l) Histogram not uniform

(m) Original patch (n) Edge direction (GDM) (o) Mean six regions (p) Histogram not uniform

(q) Original patch (r) Edge direction (GDM) (s) Mean six regions (t) Histogram uniform

Figure 4.8: Comparison between rejected candidate and accepted candidate.

cells and c = the number of estimated parameters (including location and scale parameters and shape parameters) for the distribution + 1. For example, for a 3-parameter Weibull distribution, $c = 4$. Therefore, the hypothesis that the data are from a population with the specified distribution is rejected if:

$$\chi^2 > \chi^2_{(\alpha,k-c)} \tag{4.17}$$

where $\chi^2_{\alpha,k-c}$ is the chi-square percent point function with k - c degrees of freedom and a significance level of α. In the above formulas for the critical regions, we follows the convention that χ^2_{α} is the upper critical value from the chi-square distribution and $\chi^2_{l-\alpha}$ is the lower critical value from the chi-square distribution. Note that this is the opposite of what is used in some texts and software programs.

To overcome the problems of the Chi-Square test we have switched to the Kolmogorov-Smirnov test [75]. Also this one is used to decide if a sample comes comes from a population with a specific distribution . An attractive feature of this test is that the distribution of the K-S test statistic itself does not depend on the underlying cumulative distribution function being tested. Another advantage is that it is an exact test (the chi-square goodness-of-fit test depends on an adequate sample size for the approximations to be valid).

Kolmogorov-Smirnov Goodness-of-fit test

Test for distributional adequacy The Kolmogorov-Smirnov ($K - S$) test is based on the empirical distribution function (ECDF). Given N ordered data points Y_1, Y_2, ..., Y_N, the ECDF is defined as

$$E_N \;=\; \frac{n(i)}{N} \tag{4.18}$$

where n(i) is the number of points less than Y_i and the Y_i are ordered from smallest to largest value. This is a step function that increases by $1/N$ at the value of each ordered data point. The graph (see figure 4.9) is a plot of the empirical distribution function with a normal cumulative distribution function for 100 normal random numbers. The K-S test is based on the maximum distance between these two curves.

Limitations of the K-S test Despite the advantages, the K-S has the following limitations: it only applies to continuous distributions, it tends to be more sensitive near the center of the distribution than at the tails and perhaps the most serious limitation is that the distribution must be fully

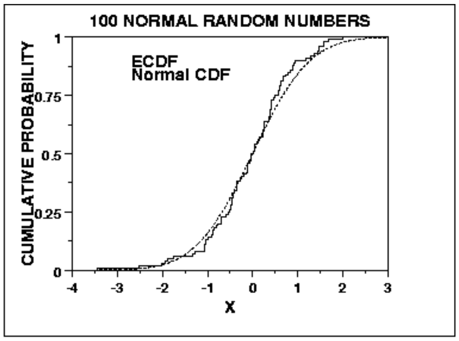


Figure 4.9: Empirical distribution function for 100 normal random numbers.

specified. That is, if location, scale, and shape parameters are estimated from the data, the critical region of the K-S test is no longer valid. It typically must be determined by simulation.

Definition The Kolmogorov-Smirnov test is defined by: the data follow a specified distribution(H_0) and the data do not follow the specified distribution (H_a). The Kolmogorov-Smirnov test statistic is defined as:

$$D = max_{1 \leq i \leq N} \left| F(Y_i) - \frac{i}{N} \right| \tag{4.19}$$

where F is the theoretical cumulative distribution of the distribution being tested which must be a continuous distribution (i.e., no discrete distributions such as the binomial or Poisson), and it must be fully specified (i.e., the location, scale, and shape parameters cannot be estimated from the data).

Critical Values The hypothesis regarding the distributional form is rejected if the test statistic, D, is greater than the critical value obtained from a table. There are several variations of these tables in the literature that use somewhat different scalings for the K-S test statistic and critical regions. These alternative formulations should be equivalent, but it is necessary to ensure that the test statistic is calculated in a way that is consistent with how the critical values were tabulated.
After goodness-of-fit test has been performed on the patches generated by foreground those which are positive are spatially clustered. This is justified by the fact that several positive patches overlap on the same pedestrian head. The clustering algorithm [76] that we have applied is in the following section.

4.6 Spatial Clustering

Clustering is the process by which discrete objects can be assigned to groups which have similar characteristics. In our case we are interesting to a spatial cluster which clustering the candidates overlap on the same pedestrian head. There are many algorithms used in clustering. The one presented here was described in [76]. Each object is represented by an ordered n-tuple of its attributes. Objects with similar characteristics must have individual attributes which are close in value. Since distance is a measure of ”closeness”, it can be used in clustering. For each cluster, there will be a single point which is designated as a hub. The placement of the hub within the cluster is determined by the algorithm. Algorithm 1 assumes a minimum of 2 clusters and a maximum of p clusters where p is the number of objects. The algorithm is as follows:

1. From the set of objects (data points), choose any one to be the first hub.
2. Find the point which is farthest from this hub, and make it the second hub.
3. For each remaining object, find the distance from the object to each hub and assign the object to the hub to which it is closer.
4(a). To determine whether it is necessary to find a third hub, do the following.
4(b). Calculate the distance from each object to its hub. If any distance is greater than R, it is necessary to find a third hub.
5. To find the third hub, find the object which is farthest from its hub (1 or 2). So, if Point 8 belongs to Hub 1, look at the distance from Point 8 to Hub 1, but not Point 8 to Hub 2.
6. Then, calculate the distance from each point to the new hub to see if that distance is less than the points distance to its current hub. If so, reassign the point to Hub 3.
7. To determine the stop value, R, find the average distance between the hubs and divide by two. Since there are three hubs, there will be three distances to average together (H1 - H2, H1 - H3, and H2 - H3).
8. Once again, see if any distance from a point to its hub exceeds R.
9. Continue this process until all points are within R units of their hub or until all points are themselves hubs.

In each hub we take as good candidate the point that better satisfies the goodness-of-fit test.

Chapter 5

Head based tracker

An overview of our tracking algorithm is shown in figure 5.1. Given a patch (t) and a search window (t+1) a correlation-based tracker has been implemented. We adopted a random sampling techniques to estimate the local maxima of the correlation function. The head detector is applied to the first five local maximum and the point that better satisfies the statistic tool is taken to track the target.

5.1 Correlation maxima computation and tracking

The tracking method we explored is based on correlation measurements. Correlation is a well-know tool for signals detection. In the framework of digital image processing, it is one of the main approaches for tracking. Computing a correlation between two functions consists in computing the product of both at each point and then integrating the result. For vectors (or, in the case of images, matrices), this is the regular dot product. It can be understood as a measure of similarity between two functions. We aim to detect pedestrians looking at their heads. More exactly, given an frame t and an hypothetical head pedestrian position $p_{(t)} = (x_{(t)}, y_{(t)})$ (on the image plane), we consider the image $r_{(t)}$ centered around $p_{(t)}$. The size of the image region $r_{(t)}$ is NxN where $N = size(patch) + cost$ (cost is the dimension of a neighborhood opportunely defined). We compute the correlation

$$Corr(r_{(t)}, r_{(t+1)}) \tag{5.1}$$

between $r_{(t)}$ and the corresponding region on the successive frame $r_{(t+1)}$. This means we compute a measure of similarity between the cropped, windowed

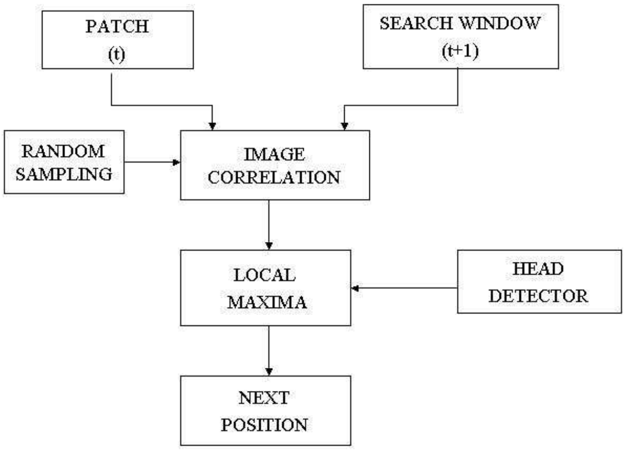


Figure 5.1: Algorithm tracking.

region around the tracker in the position p in the frame t and a slightly shifted region of same size cropped out of the following frame. We estimate the local maxima with a random sampling techniques for multi-modalities of the distributions and we consider the five higher correlation value. We check each maximum and we apply the head detector algorithm at each maximum (or in a neighborhood opportunely defined). The location $p_{(t+1)} = (x_{(t+1)}, y_{(t+1)})$ of the best matching between the two image regions is given by the maximum that better satisfies the head detector. The idea underlying a basic correlation-based tracking algorithm is to iteratively update position of the trackers according to the best-of-fit among the five higher correlation values. The vector identified by the position of $p_{(t+1)}$ with respect to $p_{(t)}$ corresponds to the displacement vector of the current image region over the two frames. In our case, the interesting thing behind this well known method is that in two consecutive frames a pedestrian can cover a limited distance, so it is reasonable to think that the searching region, used for correlation computation, contains the real target position.

Chapter 6

Experiments and Results

6.1 Data-sets

The Datasets represents the video sequences that we have used for our head detection and tracking task. For our experiments we have used two outdoor video sequences and two indoor video sequences. The camera parameters (see chapter 3) are the following, the angles are expressed in radiant:

Video	Size	h_{camera}	α	θ
Video1	240x320 pixel	5.99 m	1.0886	0.5714
Video2	576x720 pixel	10.85 m	0.5236	0.5714
Video3	576x720 pixel	2.5 m	1.2126	0.647
Video4	576x720 pixel	14.85 m	0.9	0.5714

6.2 Head detection results

This section exposes the results obtained by the head detector, we show the approach with the edge statistical analysis (see section 4.5) and with the template correlation matching (see section 4.4).

Template correlation matching

In figure 6.1,6.2,6.3 are showed the results obtain by the head detector. In this approach are combined an elliptic template (shape model, see section 3.4) and the shape constraints (see section 3.4). A threshold is tuned and the value is directly related to the auto-convolution of the template. In particular the value of the threshold is a percentage of the maximum value (MAC) of the auto-convolution result (see figure 6.1,6.2,6.3).

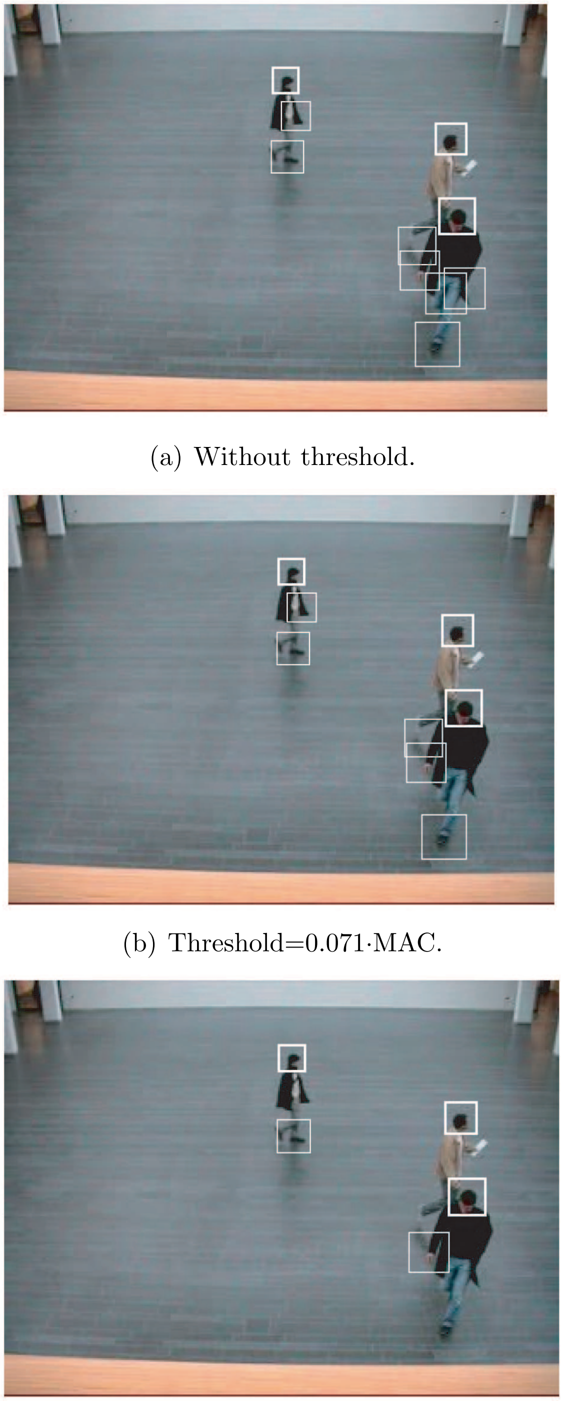
The problem of this method is that it requires threshold tuning. Unfortunately the selection of the threshold is strongly dependent on the image at hand.

Statistical edge curvature analysis

In this approach are combined an elliptic template (shape model), the shape constraints (see section 3.4) and the curvature-based models (see section 4.5). In figure 6.4, 6.8 the Chi-Square test is used as goodness-of-fit and in figure 6.5, 6.6, 6.7, 6.9, 6.10 are shown the result of the head detector where the statistic tool used is the K-S test.
In figure 6.11 are shown examples of approach with a fixed patch. In 6.11,(a) is used a squared patch of side 25 pixels, in figure 6.11,(b) is used a patch of 50 pixels and in 6.11,(c) the dimension of the patch is 200 pixels. Another example with the K-S test are shown in figure 6.12 and in figure 6.13.

6.3 Tracking results

In the first experiment of tracking we apply the simple correlation-based tracker (see section 5). The next position is the candidate that better satisfies the head detector. The first frame is manually initialized (see figure 6.14). In the figure 6.15 the first frame is initialized with the head detector, then we apply the head detector each five frame.

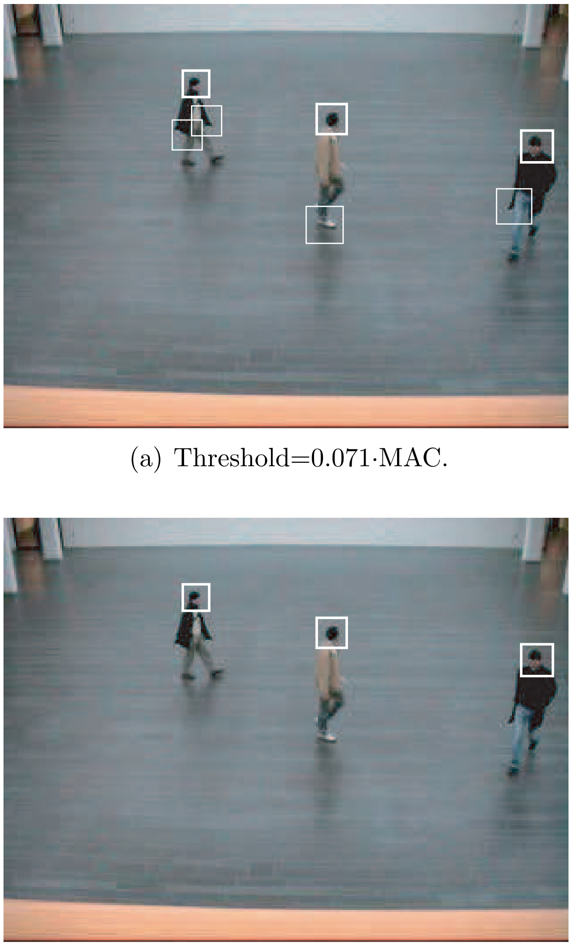


(a) Without threshold.

(b) Threshold=0.071·MAC.

(c) Threshold=0.077·MAC.

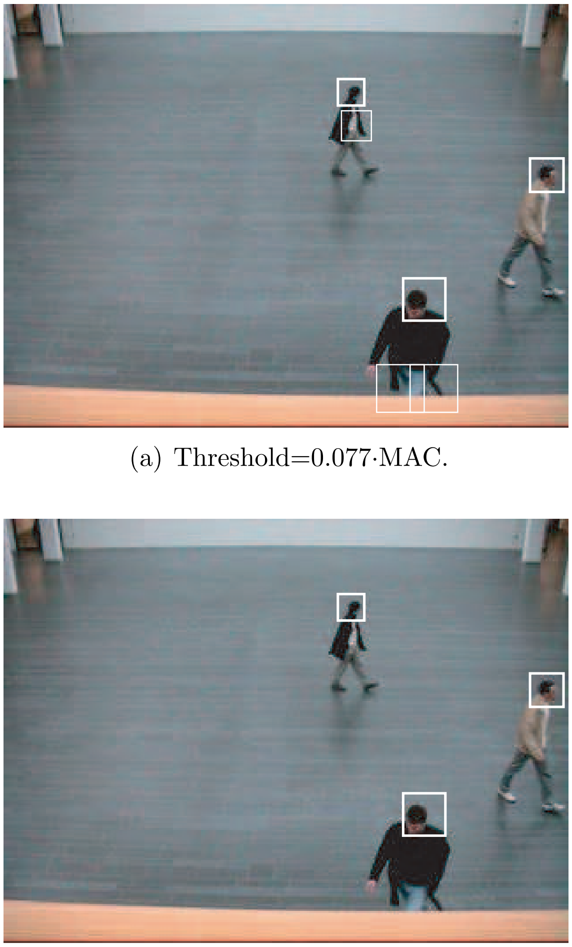
Figure 6.1: Video1: head detection with template correlation matching.



(a) Threshold=0.071·MAC.

(b) Threshold=0.083·MAC.

Figure 6.2: Video1: head detection with template correlation matching.



(a) Threshold=0.077·MAC.

(b) Threshold=0.083·MAC.

Figure 6.3: Video1 : head detection with template correlation matching.

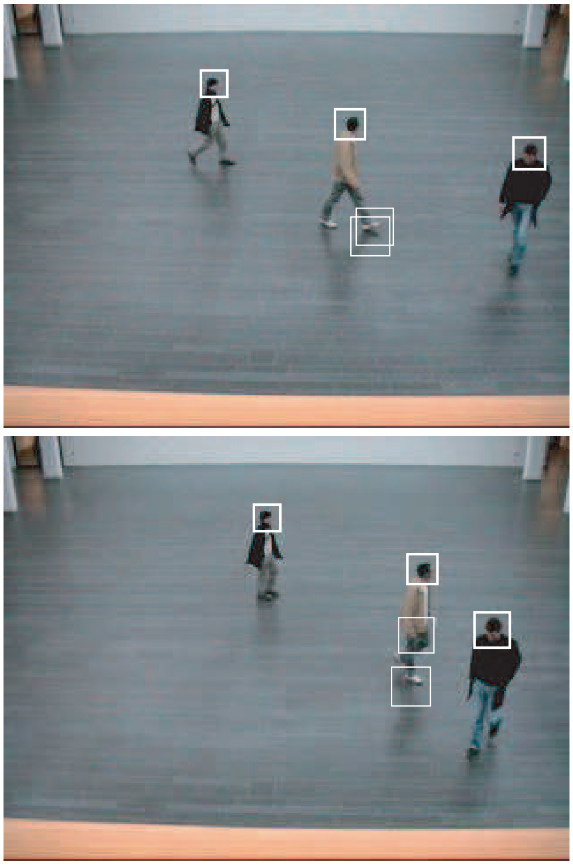


Figure 6.4: Video1: head detection with edge curvature analysis (without threshold) and Chi-Square test as statistic tool.

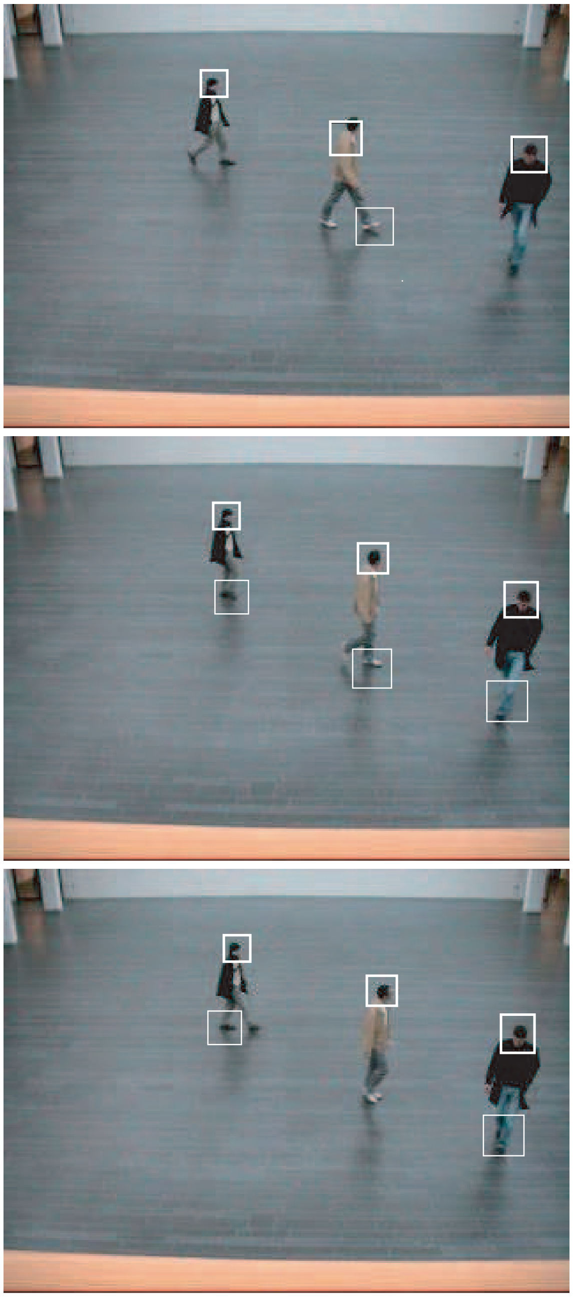


Figure 6.5: Video1 : head detection with edge curvature analysis (without threshold) and K-S test as statistic tool.

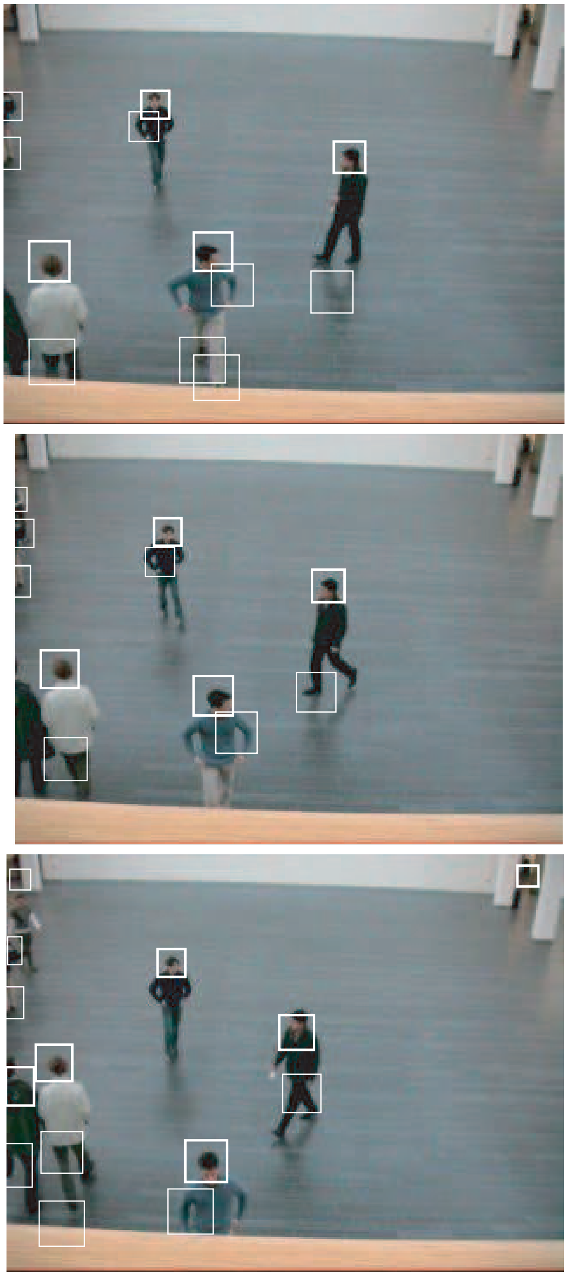


Figure 6.6: Video1 : head detection with edge curvature analysis (without threshold) and with K-S test as statistic tool.

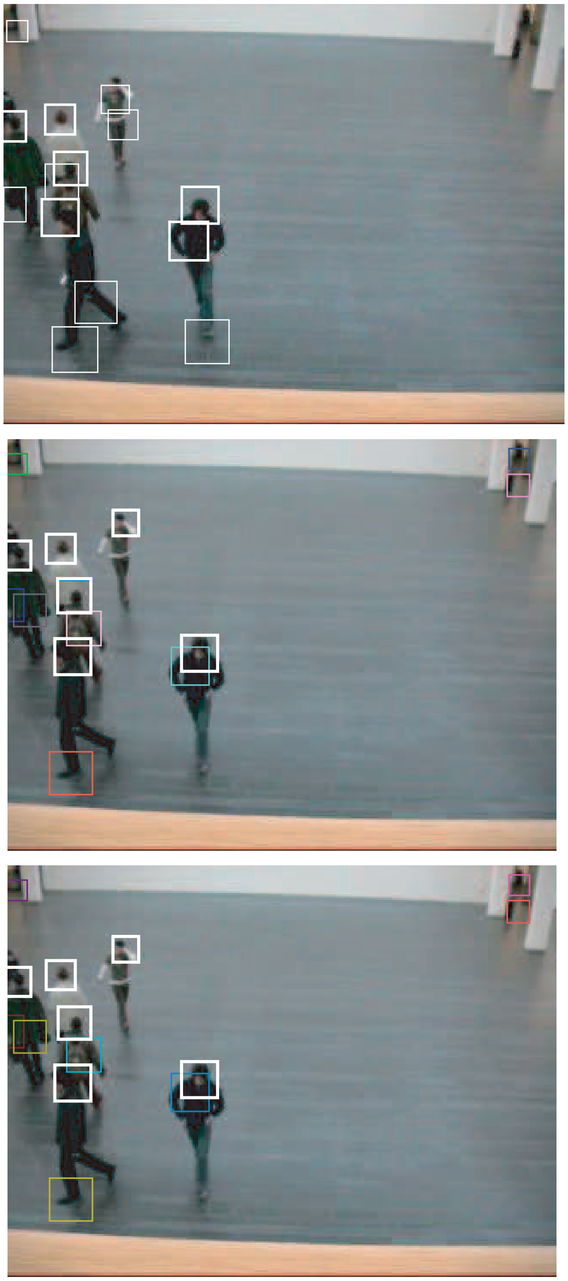


Figure 6.7: Video1 : head detection with edge curvature analysis (without threshold) and with K-S test as statistic tool.

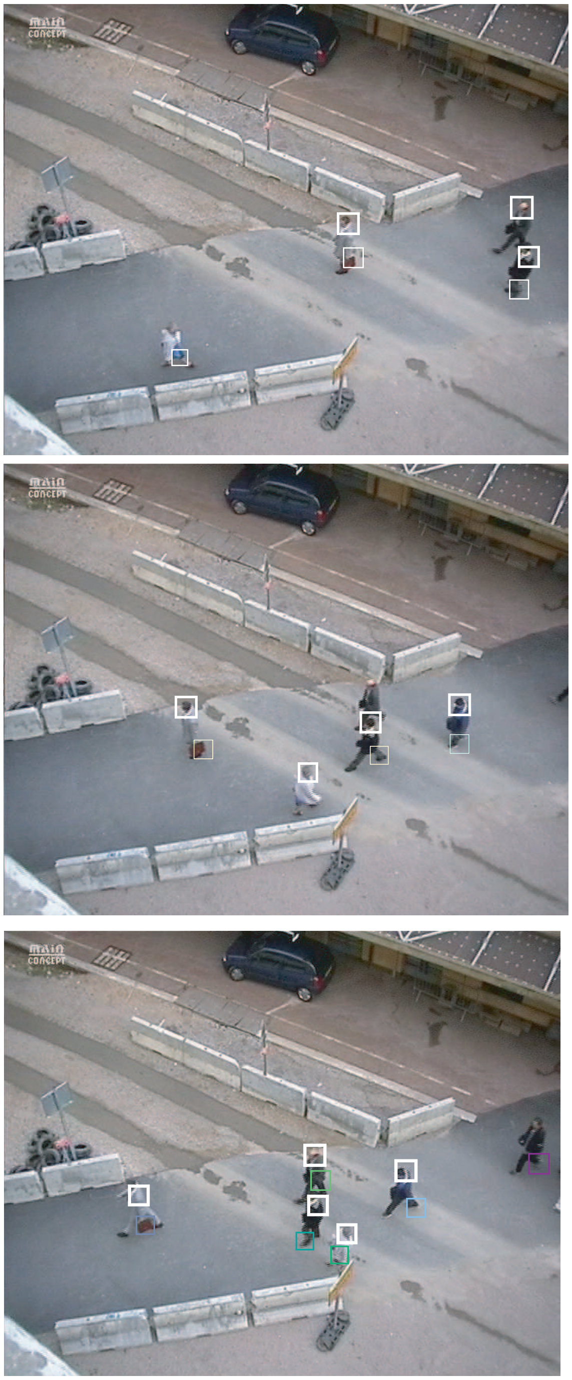


Figure 6.8: Video2 : head detection with edge curvature analysis (without threshold) and Chi-Square test as statistic tool.

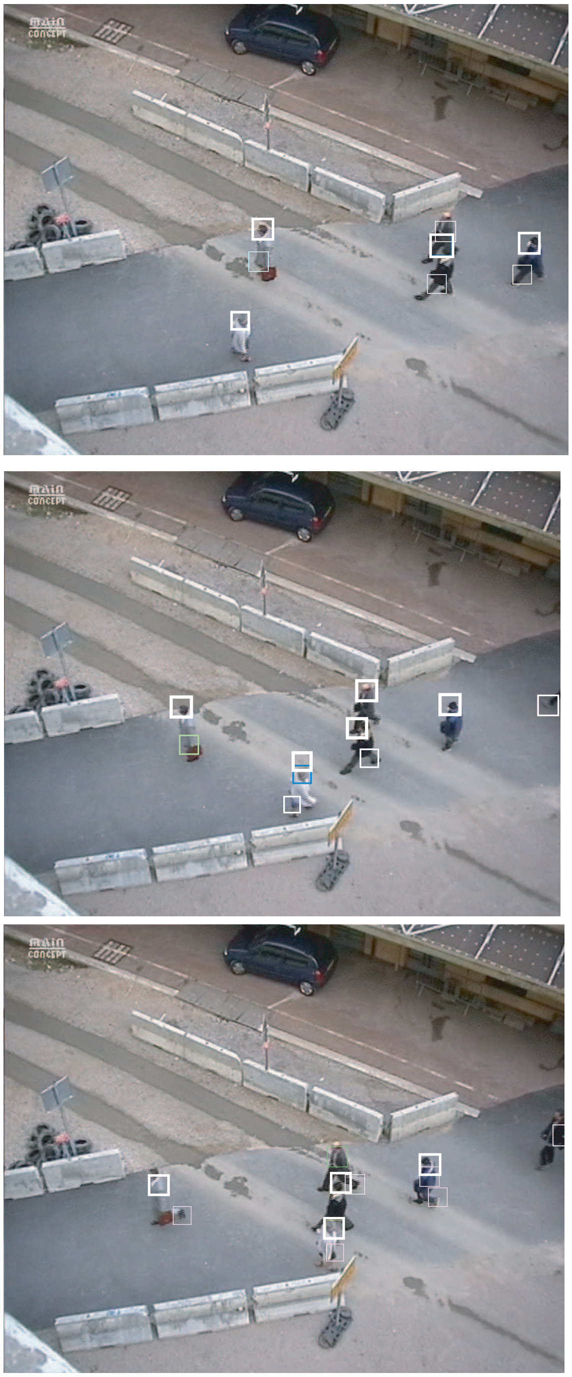


Figure 6.9: Video2 : head detection with edge curvature analysis (without threshold) and with K-S test as statistic tool.

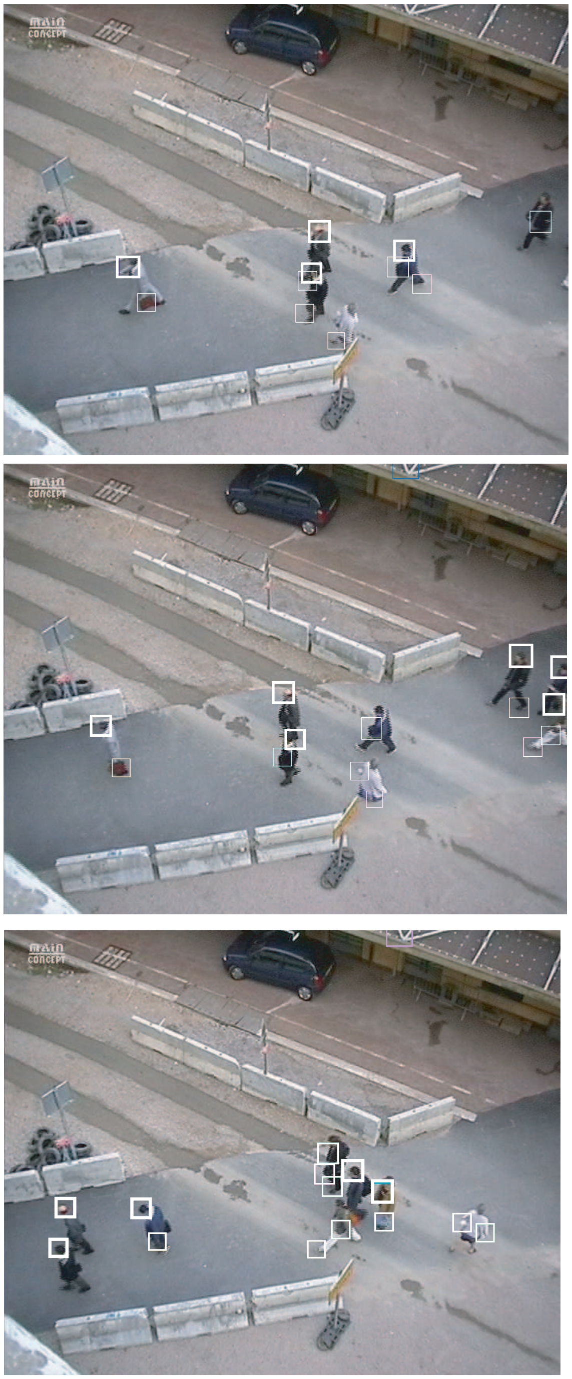
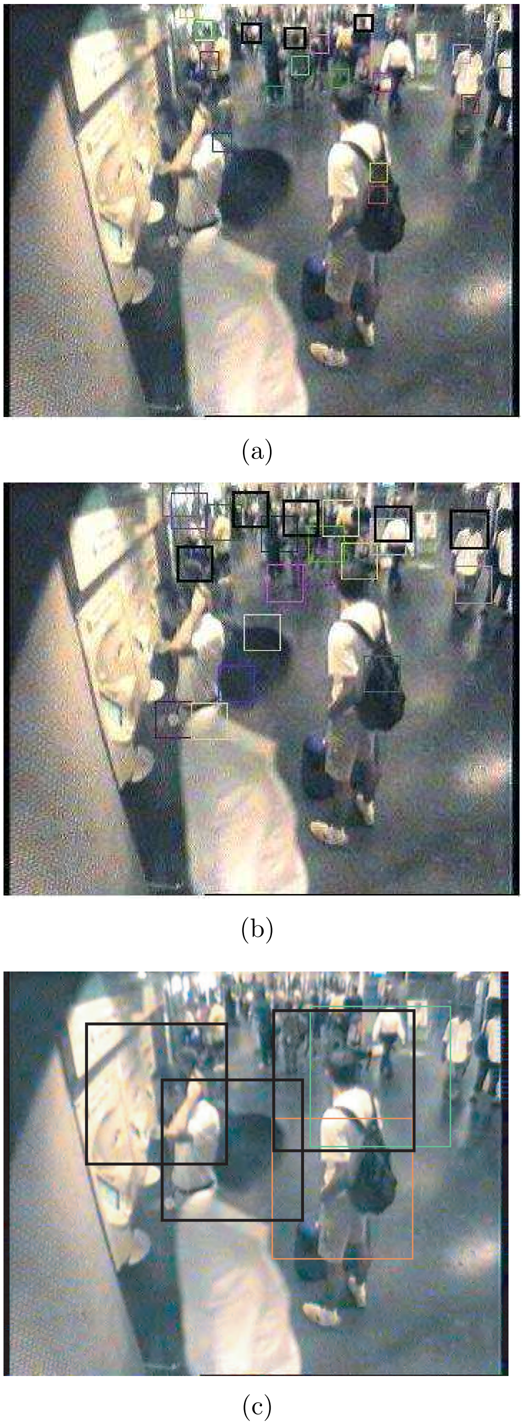


Figure 6.10: Video2 : head detection with edge curvature analysis (without threshold) and with K-S test as statistic tool.



(a)

(b)

(c)

Figure 6.11: Video3: (a) Patch of fixed side 25 pixels, (a) Patch of fixed side 50 pixels, (a) Patch of fixed side 200 pixels.

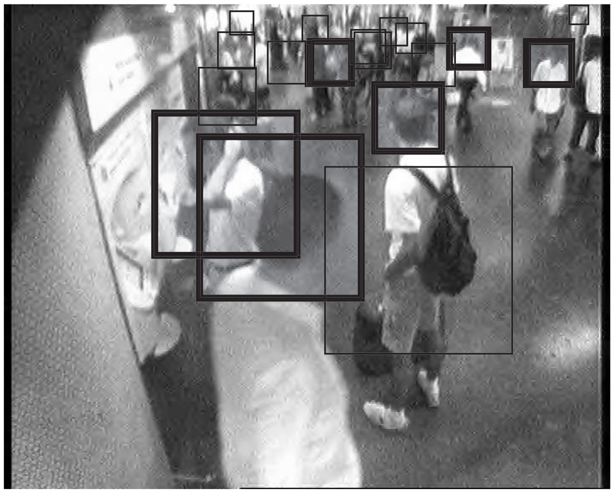


Figure 6.12: Video3: Head detector.

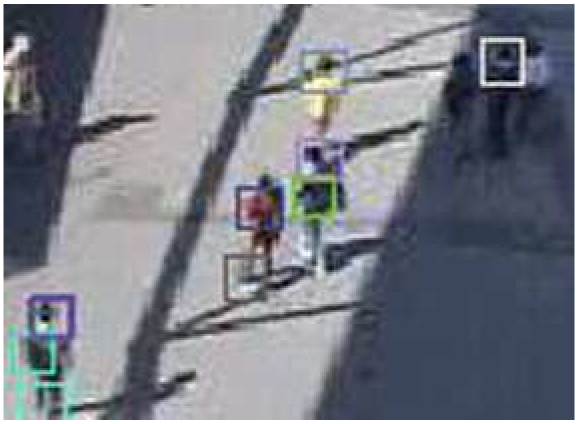
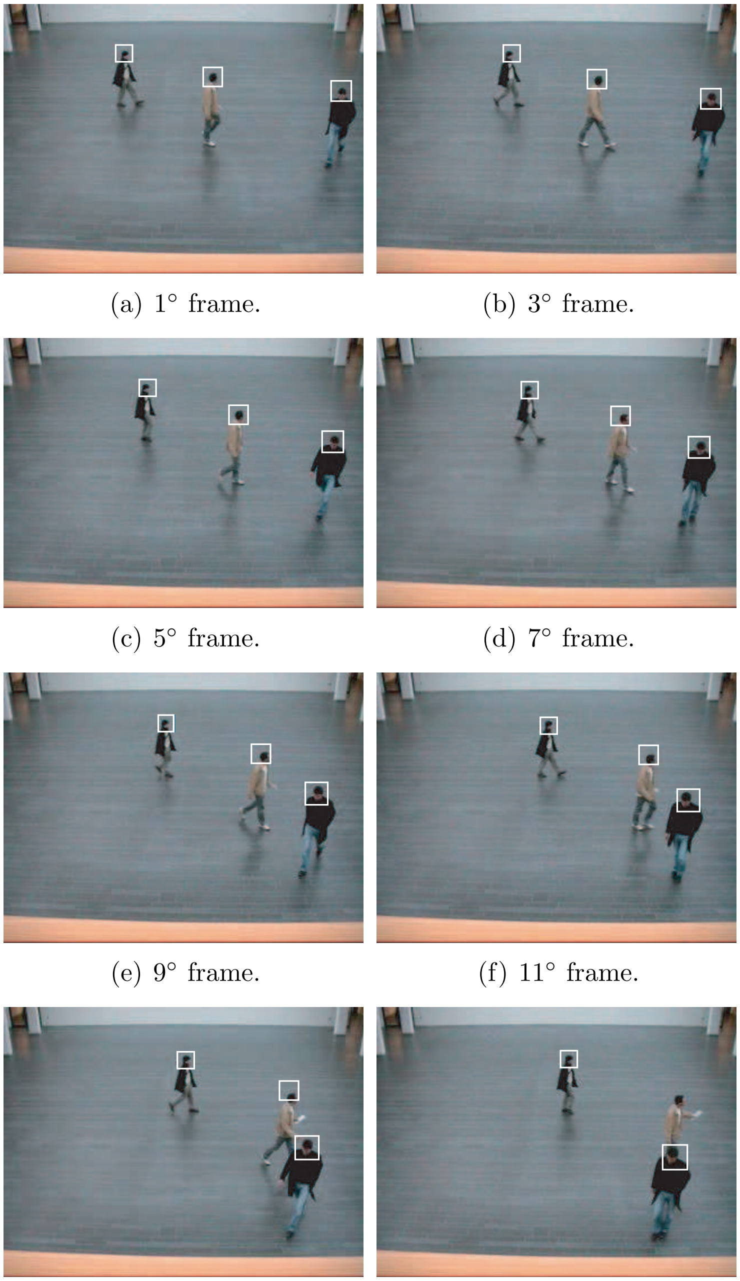


Figure 6.13: Video4: Head detector.



(a) 1° frame.

(b) 3° frame.

(c) 5° frame.

(d) 7° frame.

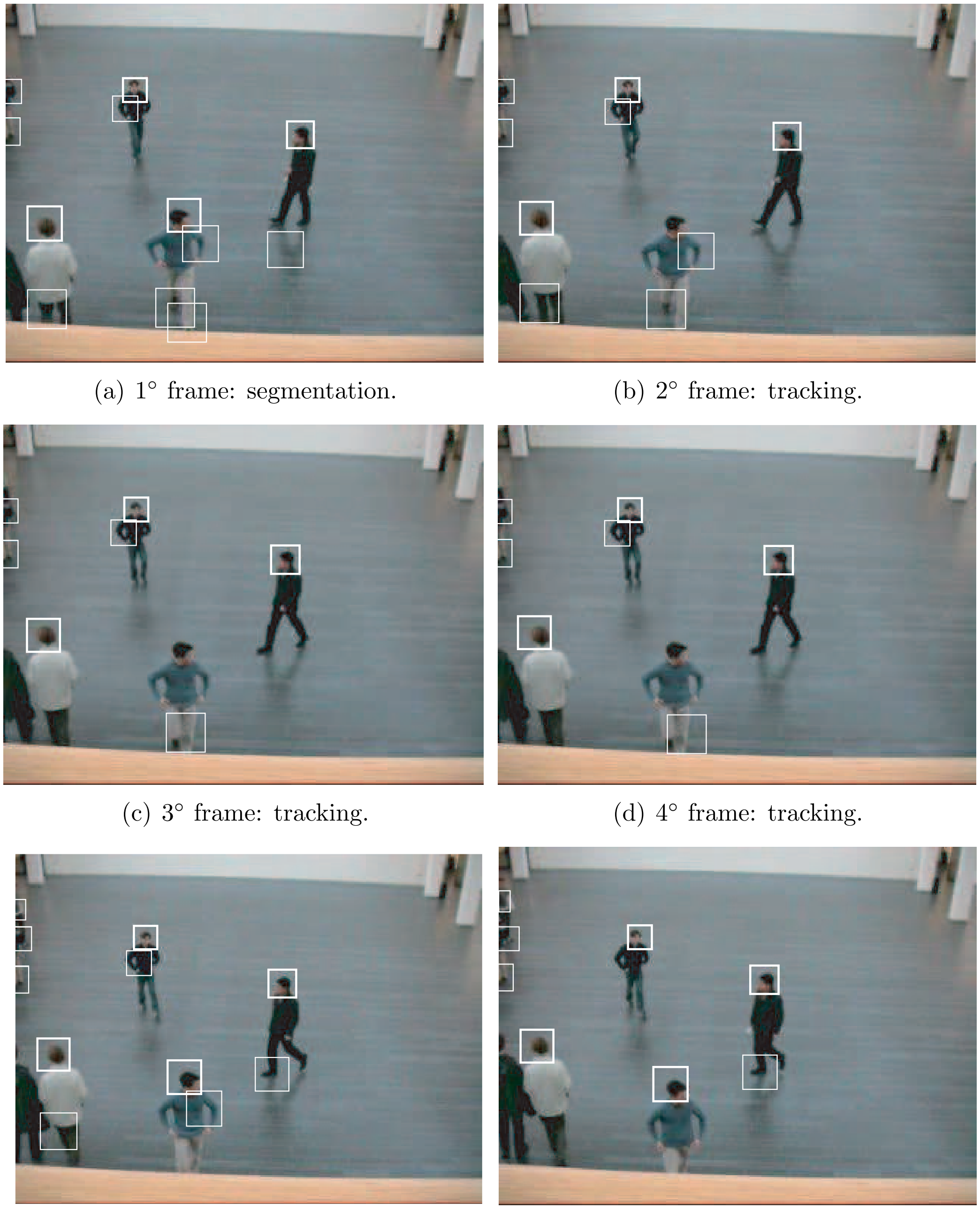
(e) 9° frame.

(f) 11° frame.

(g) 13° frame.

(h) 15° frame.

Figure 6.14: Tracking with the first frame manually initialized.



(a) 1° frame: segmentation.

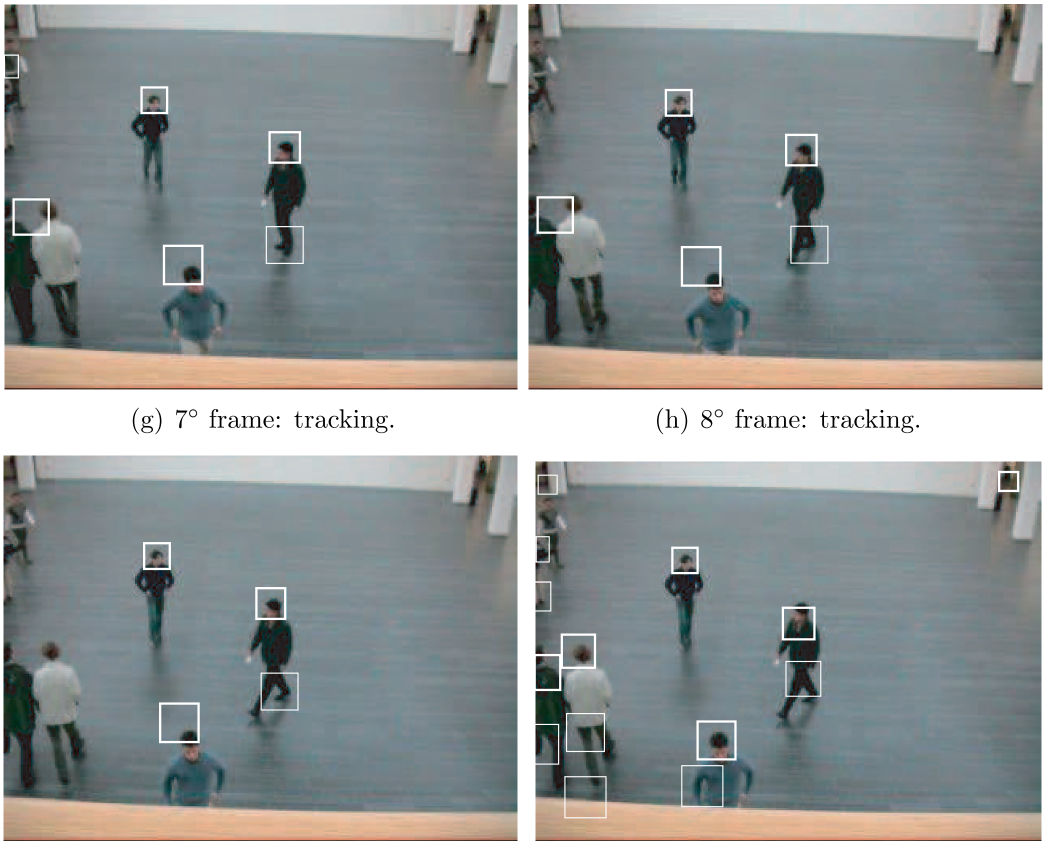
(b) 2° frame: tracking.

(c) 3° frame: tracking.

(d) 4° frame: tracking.

(e) 5° frame: segmentation.

(f) 6° frame: tracking.



(g) 7° frame: tracking.

(h) 8° frame: tracking.

(i) 9° frame: tracking.

(j) 10° frame: segmentation.

Figure 6.15: Segmentation and tracking

Conclusions and future works

In this thesis we have presented a method to use the *a priori* information for human head detection in real complex scenarios. The use of a calibrated camera allows us to select the correct scale for edge detection. To improve the edge structure we use the Beltrami anisotropic diffusion algorithm as a pre-processing step. It has been commented that the definition of the patterns is one of the critical points of a tracking system based on template matching. A part of our work has been dedicated to determine which the best way to define these templates is. So the mapping from the real head size and the scale parameter has made using a simple head shape model (ellipse), using trigonometric manipulations. The analysis of the edge structure has made using the gradient direction map which has a certain degree of illumination invariant and allows us to focus our attention on shape constraints. A statistical description of the gradient direction map is given by means of the direction histograms and goodness-of-fit statistical tests are used to validate the current candidate. We have implemented a correlation-based likelihood term, using random sampling techniques for multi-modalities of the distributions. We aim to use the head detector to give an evaluation of the different modes of the likelihood and the posterior distributions. Our algorithm is relatively computational expensive if used in a pure detection context (we have in fact to test each foreground pixel) but the computational load reduces drastically when we use it as an evaluation means. The statistical analysis of the edge curvature is resulted an useful method, because we have not fixed threshold for each video analyzed (as the correlation) and this method has a certain degree of illumination invariant.

We can refine the head shape model and we can refine the statistic analysis of the head detector using a shape model learned from data. In fact we ignore the error given by the consideration made about the circular curve using an elliptic curve. Another improvement could be to consider the information of the scale directly inside the anisotropic diffusion, doing the right edge extraction without make convolution between image and long filters. This approach could reduce the computational cost. An idea to improve the tracking algorithm is to keep and to process the candidates that don't satisfy the statistic test without displaying them, and displaying them only when the statistic test are satisfied.

Appendix

Overview of the Hermite transform theory

A digital image is usually specified by an array of point-wise intensities. For an intelligent interpretation of the image data we need to make the important information explicit. This usually implies determining spatiotemporal relationships between these intensities, and hence requires some form of local processing of the visual data. First, to make the processing local, the image is usually multiplied by a window function. The size of the window establishes the set of the image points that contribute to one basic processing step. The form of the window function determines the relative weight of each contributing image point. In order to describe the image completely, this local processing has to be repeated for a sufficient number of window positions. The form, size and spacing of the window function have to be selected. Second, for each position of the window, specific processing steps have to undertaken. As any specific choice of processing implies the search for specific patterns, selecting this process is equivalent to fixing the visual patterns which are considered most relevant *a priori*. It is very difficult to make optimal choices for the window function and processing on purely theoretical arguments. An important parameter of a window function is its size or scale. The selection of an appropriate window size makes a fundamental problem. On the other hand, in order to enable high data reduction, the window size has to be sufficiently large. On the other hand, the complexity of the analysis within each window increases rapidly with the window size. First, we can selected a window of fixed size and perform an analysis within each window that is sufficiently complex to include all visual patterns of interest. Second, we can limit the complexity of the analysis that we perform in each window, and subsequently determine the window size to describe the image locally with sufficient accuracy. Hence, instead of restricting the processing to on scale, we repeat the same processing at multiple scales and subsequently use the outputs of this processing stage to select the optimum scale at each position. If window functions of different sizes are used, then the spacing of the window functions is usually taken proportional to their size. In computer vision, the image processing problem is addressed from the point of view of interpretation, as one is mainly interested in finding the important image features such as edge and lines. The resulting processing often involves the use of the first and second order derivatives, almost always in combination with some regularizing low-pass filter. It was demonstrated that derivatives of Gaussians can model filter operation in human vision with the same accuracy as the often used Gabor filter [73]. The derivatives of Gaussians operators even have the advantage that they accomplish this modeling with fewer parameters.

First Step The first step consists to localize the original signal by multiplying it with a window function.

Second Step The second step consists of approximating the signal piece within the window by a polynomial.

Third Step The final goal is the reconstruction of the image using orthonormal polynomials. What is interesting is that when we use Gaussian local windows we deal with Hermite polynomials.

The Hermite transform theory

We consider a signal-2D (image)and we apply it a decomposition. We introduce the basic ideas on 1D analog signals. We could apply in our case the extension. This signal decomposition technique is called polynomial transform, here signals are locally approximated by polynomials. The analysis by polynomial transform is made in two steps. In the first step the original signal $L(x)$ is localized by multiplying it by a window function $V(x)$. A complete description of the signal requires that the localization process is repeated at a sufficient number of window positions and for do this we can construct a weighting function,that depend on $V(x)$:

$$W(x) \quad = \quad \sum_k V(x-kT) \tag{6.1}$$

with period T, that allow to shift the window function. Provided $W(x)$ is nonzero for all x, we get

$$L(x) \quad = \quad \frac{1}{W(x)} \sum_k L(x) \cdot V(x-kT) \tag{6.2}$$

so that we are guaranteed that the localized signals $L(x) \cdot V(x - kT)$ for all different window positions kT contain sufficient information about the original signal. The second step consist of approximating the signal piece within the window $V(x-kT)$ by a polynomial. We take the polynomial $G_n(x)$ as basis functions for the polynomial expansion. The polynomial $G_n(x)$ has degree n and is orthonormal with respect an arbitrary window function to $V^2(x)$. Under very general conditions for the original signal $L(x)$, we get that

$$V(x - kT)\Big[L(x) - \sum_{n=0}^{\infty} L_n(kT) \cdot G_n(x - kT)\Big] \quad = \quad 0 \tag{6.3}$$

that mean that decomposition on basis G_n make a good approximation of the original signal $L(x)$ with

$$L_n(kT) \quad = \quad \int_{-\infty}^{+\infty} L(x) \cdot G_n(x - kT) \cdot V^2(x - kT) \tag{6.4}$$

The approximation error between a signal and a polynomial can be made arbitrary small taking the degree of the polynomial expansion sufficiently high, as is well know from Taylor expansions. This implies that the description of the localized signal $L(x) \cdot V(x - kT)$ can, up to an arbitrary small approximation error, be reduced to specifying a finite set of polynomial coefficients $L_n(kT)$. Combining (6.3) and (6.2), we get the following expansion for the complete signal

$$L(x) \quad = \quad \sum_{n=0}^{\infty} \sum_{k} L_n(kT) \cdot P_n(x - kT) \tag{6.5}$$

where

$$P_n(x - kT) \quad = \quad \frac{G_n(x)V(x)}{W(x)} \tag{6.6}$$

Equation (6.4) implies that the coefficients $L_n(kT)$ can be derived from the signal $L(x)$ by convolving with the filter functions

$$D_n(x) = G_n(-x)V^2(-x) \tag{6.7}$$

followed by sampling at multiples of T. This mapping from the original signal $L(x)$ to the polynomial coefficients $L_n(kT)$ is called a forward polynomial transform. The signal reconstruction from these coefficients can be done according to the equation 6.5 and it is called inverse polynomial transform. We use a special case [73] where the local window is Gaussian

$$V(x) \quad = \quad \frac{1}{\sqrt{\sqrt{\pi}\sigma}} e^{(\frac{-x^2}{2\sigma^2})} \tag{6.8}$$

The orthogonal polynomials that are associated with $V^2(x)$ are the Hermite polynomials. The reasons because we use Gaussian window are manifold among the others there are that : the properties of the Hermite transform can be easily derived and evaluated, the Gaussian windows which are separated by twice the spread σ are a good model for the overlapping receptive fields found in physiological experiments, it will turn out that the Hermite transform involves filter functions that are derivatives of Gaussian and the Gaussian window minimizes the product of uncertainties in the spatial and frequency domain.

Bibliography

[1] K.D.Baker T.D.Grove and T. N. TAN. Color based object tracking. In *14th International Conference on Pattern Recognition*, 1998.

[2] D. Adalsteinsson and J. Sethian. A fast level set method for propagating interfaces, 1995.

[3] R.R. Kohler A.R. Hanson E.M. Riseman J.R. Beveridge, J. Griffith. Segmenting images using localized histograms and region merging, 1989.

[4] Ale&#353; Leonardis, Alok Gupta, and Ruzena Bajcsy. Segmentation of range images as the search for geometric parametric models. *Int. J. Comput. Vision*, 14(3):253–277, 1995.

[5] Song Chun Zhu, Tai Sing Lee, and Alan L. Yuille. Region competition: Unifying snakes, region growing, energy/bayes/MDL for multi-band image segmentation. In *ICCV*, pages 416–, 1995.

[6] C. Bouman and M. Shapiro. A multiscale random field model for bayesian image segmentation. *IP*, 3(2):162–177, March 1994.

[7] R. M. Haralick and L. G. Shapiro. Image segmentation techniques. *Comput. Vision Graphics Image Process.*, pages 29:100–132, 1985.

[8] R. Cristi H. Derin, H. Eelliott and D. Geman. Bayes smoothing algorithms for segmentation of binary images modeled by markov random fields. *IEEE Trans. on Pat. Anal. and Machine Intel.*, pages 707–720, 1984.

[9] D. Marr. *Vision.* W.H.Freeman and Company, 1982.

[10] D. Marr and E. C. Hildreth. Theory of edge detection. *Proceedings of the Royal Society, London B*, 207:187–217, 1980.

[11] J. F. Canny. Finding edges and lines in images. Master's thesis, MIT, 1983.

[12] J.F.Canny. A computational approach to edge detection. *IEEE Trans.Patt.Anal.Mach.Intell.*, 8:679–698, 1986.

[13] R. Deriche. Using Canny's criteria to derive a recursively implemented optimal edge detector. *The International Journal of Computer Vision*, 1(2):167–187, May 1987.

[14] Margaret M. Fleck. Multiple widths yield reliable finite differences (computer vision). *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(4):412–429, 1992.

[15] S. Thiruvenkadam Y. Chen, H.D. Tagare and E.A. Geiser. Using prior shapes in geometric active contours in a variational framework. *International Journal of Computer Vision*, 50(3):315–328, December 2002.

[16] A. Pentland, R. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases, 1994.

[17] Christos Faloutsos, Ron Barber, Myron Flickner, Jim Hafner, Wayne Niblack, Dragutin Petkovic, and William Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3(3/4):231–262, 1994.

[18] M. Leo P. Spagnolo. G. Attolico. A. Distante. Shape based people detection for visual surveillance systems. 2003.

[19] Massimo Bertozzi, Alberto Broggi, Roland Chapuis, Frédéric Chausse, Alessandra Fascioli, and Amos Tibaldi. Shape-based pedestrian detection and localization. In *Procs. IEEE Intl. Conf. on Intelligent Transportation Systems 2003*, pages 328–333, Shangai, China, October 2003.

[20] Jen-Hui Chuang, Jin-Fa Sheu, Chien-Chou Lin, and Hui-Kuo Yang. Shape matching and recognition using a physically based object model. 25(2):211–222, April 2001.

[21] G.S. Cox. Template matching and measures of match in image processing.

[22] P. Remagnino, P. Brand, and R. Mohr. Correlation techniques in adaptive template matching with uncalibrated cameras, 1994.

[23] Luigi Di Stefano and Stefano Mattoccia. Fast template matching using bounded partial correlation. *Mach. Vision Appl.*, 13(4):213–221, 2003.

[24] Hemant D. Tagare. Deformable 2-d template matching using orthogonal curves. *Tech. Rep. 95-4*, 1995.

[25] Stan Liu, Lifeng; Sclaroff. Index trees for efficient deformable shape-based retrieval. March 22 2000.

[26] T.Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):77–116, 1998.

[27] T.Lindeberg. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. *International Journal of Computer Vision*, 11:283–318, December 1993.

[28] T.Lindeberg. On scale selection for differential operators. In *Proc. 8th Scandinavian Conference on Image Analysis*, pages 857–866, 1994.

[29] T.Lindeberg. Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):117–154, 1998.

[30] C.F.Olson. Variable-scale smoothing and edge detection guided by stereoscopy. In *Proceedings of Computer Vision and Pattern Recognition*, pages 80–85, 1998.

[31] Clark F. Olson. Adaptive-scale filtering and feature detection using range data. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 22(9):983–991, 2000.

[32] D. Terzopoulos A. Witkin and M. Kass. Constraints on deformable models: Recovering 3d shape and nonrigid motion. *Artificial Intelligence*, 36(1):91–123, 1988.

[33] T. Kaneko and O. Hori. Feature selection for reliable tracking using template matching. *CVPR03*, pages 796–802, 2003.

[34] N. Johnson and D. Hogg. Lerning the distribution of object trajectories for events recognition. 1996.

[35] G.Kitagawa. Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996.

[36] A. Blake M. Isard. Condensation - conditional density propagation for visual tracking. *International Journal on Computer Vision*, 1(29):5–28, 1998.

[37] C.R. Wren and A.P. Pentald. Dynamic models of human motion. *In proceding of FG98*, 1998.

[38] F. Jurie and M. Dhome. Real time 3d template matching. *CVPR01*, pages 791–796, 2001.

[39] D. De Carlo and D. Metaxas. Optical flow constraints on deformable models with application to face tracking. *Int.J.of Computer Vision*, 38(2):99–127, 2000.

[40] B. Bascle and Rachid Deriche. Region tracking through image sequences. In *ICCV*, pages 302–307, 1995.

[41] T. J. Broida, S. Chandrashekhar, and R. Chellappa. Recursive 3-D motion estimation from a monocular image sequence. *IEEE Transactions on Aerospace and Electronic Systems*, 26(4):639–656, 1990.

[42] N. Brady and N. O'Connor. Object detection and tracking using an em-based motion estimation and segmentation framework. *ICIP*, A:925–928.

[43] T J Broida and R Chellappa. Estimation of object motion parameters from noisy images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(1):90–99, 1986.

[44] M. Gelgon and P. Bouthemy. A region-level graph labeling approach to motion-based segmentation. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 514–519, Puerto-Rico, June 1997.

[45] F. Meyer and P. Bouthemy. Region-based tracking using affine motion models in long image sequences. *CVGIP : Image Understanding*, 60(2):119–140, September 1994.

[46] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[47] M.Isard A.Blake. Active contours. 1.

[48] L. D. Cohen. On active contour models and balloons. *Computer Vision, Graphics, and Image Processing. Image Understanding*, 53(2):211–218, 1991.

[49] Rupert Curwen and Andrew Blake. Dynamic contours: real-time active splines. pages 39–57, 1993.

[50] Michael Isard and Andrew Blake. Contour tracking by stochastic propagation of conditional density. In *Proceedings of the 4th European Conference on Computer Vision-Volume I*, pages 343–356. Springer-Verlag, 1996.

[51] D. Metaxas and D. Terzopoulos. Constrained deformable superquadrics and nonrigid motion tracking. *CVPR*, 91:337–343.

[52] D. Terzopoulos and R. Szeliski. Tracking with Kalman snakes. In *Active Vision*, pages 3–20. MIT Press, Cambridge, MA, 1992.

[53] Dieter Koller, Joseph Weber, and Jitendra Malik. Robust multiple car tracking with occlusion reasoning. In *ECCV (1)*, pages 189–196, 1994.

[54] F. Leymarie and M. D. Levine. Tracking deformable objects in the plane using an active contour model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(6):617–634, 1993.

[55] Andrew Blake and Roberto Cipolla. Robust estimation of surface curvature from deformation of apparent contours. In *Proceedings of the First European Conference on Computer Vision*, pages 465–474. Springer-Verlag, 1990.

[56] Roberto Cipolla and Andrew Blake. Motion planning using image divergence and deformation. pages 189–201, 1993.

[57] R. Cipolla and A. Blake. The dynamic analysis of apparent contours. pages 616–623, Los Alamitos, CA, 1990. IEEE Computer Society Press.

[58] Nicholas Ayache, Isaac Cohen, and Isabelle Herlin. Medical image tracking. pages 285–302, 1993.

[59] Michael J. Black. Combining intensity and motion for incremental segmentation and tracking over long image sequences. In *Proceedings of the Second European Conference on Computer Vision*, pages 485–493. Springer-Verlag, 1992.

[60] F. Heitz and P. Bouthemy. Multimodal estimation of discontinuous optical flow using markov random fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(12):1217–1232, December 1993.

[61] S. Ayer, P. Schroeter, and J. Bigun. Segmentation of moving objects by robust motion parameter estimation over multiple frames. In J. O. Eklundh, editor, *Computer Vision–ECCV94*, volume II, pages 316–327. Springer, 1994.

[62] B. Bascle, P. Bouthemy, R. Deriche, and F. Meyer. Tracking complex primitives in an image sequence. In *12th Int. Conf. on Pattern Recognition, ICPR'94*, Jerusalem, Israel, October 1994.

[63] Stan Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 232–237, June 1998.

[64] J. Ivins and J. Porrill. Constrained active region models for fast tracking in color image sequences. *CVIU*, 72(1):54–71, October 1998.

[65] D.Harwood T.Horpraser and L.S.Davis. A ststistical approach for real-time robust background subtracktion and shadow detection. In *IEEE ICCV99 Frame Rate Workshop*, 1999.

[66] Marc-Andr Ameller, Bill Triggs, and Long Quan. Camera pose revisited - new linear algorithms.

[67] M. Antone and S. Teller. Automatic recovery of relative camera rotations for urban scenes. pages 282–289.

[68] Long Quan and Zhong-Dan Lan. Linear n-point camera pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):774–780, 1999.

[69] Jan J. Koenderink. The structure of images. *Biological Cybernetics*, (50):363–370, 1984.

[70] A. M. Polyakov. Quantum geometry of bosonic strings. *Physics Letters B*, 103:207–210, July 1981.

[71] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(7):629–639, 1990.

[72] B. M. ter Haar Romeny. Geometric–driven diffusion in computer vision. *Kluwer Academic Publishers*.

[73] J.B. Martens. The hermite transform–theory. *IEEE Transactions on Signal Processing*, 38(9):1595–1605, 1990.

[74] George W Snedecor and William G Cochran. *Statistical Methods*. Iowa State University Press, Ames, 1989.

[75] Laha Chakravarti and Roy. In *Handbook of Methods of Applied Statistics, Volume I*, pages 392–394. John Wiley and Sons, 1967.

[76] Robert and Clason. Applications: Finding clusters: An application of the distance concept. pages 301–307, 1990.