Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2008, Article ID 250780, 23 pages
doi:10.1155/2008/250780

Research Article

Tracking Using Continuous Shape Model Learning in the Presence of Occlusion

M. Asadi and C. S. Regazzoni

Department of Biophysical and Electronic Engineering, University of Genoa, Via All'Opera Pia 11a, 16145 Genoa, Italy

Correspondence should be addressed to C. S. Regazzoni, carlo@dibe.unige.it

Received 1 April 2007; Revised 2 October 2007; Accepted 17 January 2008

Recommended by Frank Ehlers

This paper presents a Bayesian framework for a new model-based learning method, which is able to track nonrigid objects in the presence of occlusions, based on a dynamic shape description in terms of a set of corners. Tracking is done by estimating the new position of the target in a multimodal voting space. However, occlusion events and clutter may affect the model learning, leading to a distraction in the estimation of the new position of the target as well as incorrect updating of the shape model. This method takes advantage of automatic decisions regarding how to learn the model in different environments, by estimating the possible presence of distracters and regulating corner updating on the basis of these estimations. Moreover, by introducing the corner feature vector classification, the method is able to continue learning the model dynamically, even in such situations. Experimental results show a successful tracking along with a more precise estimation of shape and motion during occlusion events.

1. INTRODUCTION

In security surveillance issues, tracking methods may be classified into two main groups: feature-based and model-based. In the model-based category, learning and updating the model, in order for it to adapt to unavoidable changes in the target, plays an important role in increasing the precision of tracking. Since the scene undergoes different changes, clutter, distracters, and occlusions may appear, all of which may distract the tracker. For example, when occlusion occurs, not only the estimation of the new position of the object is affected, since it has a model-updating phase, the model estimation can also be influenced by the occluder. This causes the model to contain the most recent information about the shape of the occluder, which may lead to a distraction toward the occluder in successive frames. To overcome such problems, the algorithm must at least be able to recognize such undesirable situations to stop learning the model. Although this avoids the model from being affected by the unwanted data, if an occlusion lasts for a long enough time, the object may undergo some changes that are not considered in the model. This may again lead to a failure in tracking, after occlusion ends, even if the tracker passes the occlusion successfully. This is the rationalization for the following assertion: it is much better that the model is *partially* learned, when an occlusion is realized. Partial learning here means that the parts of the target that are not under occlusion may help the model to consider the most recent changes. In this case, in each frame of a successive set, some parts of the object are being learned while the other parts are waived. This paper presents such an approach, in which the model is based on corners information. In the literature, several corner model-based tracking methods have been proposed, while none of them considers a partial learning.

Oberti et al. [1] proposed an algorithm to track multiple objects by modeling them using corner information [2]. The approach is applied to the output of a detection-and-tracking system to learn the object model adaptively when the object is completely isolated. The learning phase is stopped when two or more objects are close, and the most recent model is used to individualize the targets. Marcenaro et al. presented a tracking and classification method that uses a similar method as a part of the tracking [3]. Before commencing the model, they use a simple background subtraction to find the changes, and based on the overlap of two blobs in two

successive frames, the object is tracked and identified. The model is used only when it is ready and an overlap has occurred between two blobs in the current frame. In this case, the model can help in distinguishing between the objects.

A corner-based method combined with a Kalman filter is used by Gabriel et al. [4]. In this method, each object in the image is represented by several interest points obtained using a color Harris detector [5]. Then, a joint geometric and appearance model is formed for the object by a 12D vector. To track the object in the next frame, the position of the object center is predicted using the Kalman filter and the color Harris detector is applied to the region. To find the match of any corner, they compare every corner in the current region of interest with all the corners of the model using Mahalanobis distance.

Wei and Piater have presented a paper that uses a mixture particle filter to analyze the feature points clusters [6]. In this method, the Harris and the Kanade-Lucas-Tomasi (KLT) corner detectors [7] are applied to each frame, in order to detect feature points (FPs) and their velocities. Then, the FPs are clustered based on their spatial location and the temporal coherency (similar motion). Each cluster is considered as a mixture component modeled by an individual particle filter. The last step is to apply the expectation-maximization (EM) algorithm [8] to recluster the clusters by merging overlapped clusters and spatially splitting disjointed clusters.

In the aforementioned methods, corner information is used to detect the object, or to help the filters in observation and prediction correction. Rosenberg and Werman [9]. presented a method for object motion detection and tracking in which the local motion is represented as a probability distribution matrix. A set of points, not necessarily corners, inside the object is selected. Considering an area around any point, the probability of the displacement of the point between two frames is calculated using the probability distribution matrix of that area, which is computed using the color information. At the end, the global object motion is computed by averaging the motion of all the points. This can cause error accumulation due to outliers. Furthermore, the color information can be affected during occlusion. Instead, in this paper, the global motion is first estimated directly from the nonlinear voting space that can lead to correction of the local motion of the corners. Moreover, in case of occlusion, it can be decided whether any single corner belongs to the object.

The paper presented by Wiskott combines Gabor and Mallat wavelets for a "segmentation from motion" process [10]. A combination of two different types of wavelets is used to overcome the aperture problem. First, the image flow field between two successive frames is computed, based on the Gabor wavelets. Then, a histogram over the flow vectors is evaluated to choose certain peaks as motion hypotheses. After this, just edges are evaluated. Each motion hypothesis is checked for each edge pixel. Finally, each edge pixel is categorized if it belongs to a given motion hypothesis. However, this method mainly uses motion information and fails during occlusion [10].

This paper presents a Bayesian framework for a corner model-based method for tracking nonrigid objects in the presence of occlusions. The main contribution of the paper is that the method is able to learn the model continuously, even in the case of occlusions. It includes automatic decision making, which determines when to completely learn the model, when to change the learning strategy due to the cluttered scene, and how to learn the model during occlusion. Actually, based on the analysis in a multimodal voting space, the algorithm decides if there is a suspicious situation. Then, it classifies the corners into good, mixed-good, and malicious corners, in order to apply different strategies on each class. This allows the model to continue learning in either certain parts of the object or the entire object. This is done regardless of whether the suspicious situation is due to an occlusion or other clutter in the scene.

The rest of the paper is organized as follows. In Section 2, the Bayesian framework is explained. Section 3 explains the object position estimation and tracking. Section 4 discusses continuous learning during occlusions and explains the learning strategy. Section 5 provides some discussion on the continuous learning strategy. The experimental results are shown and discussed in Section 6. Finally, the conclusions and the prospects for future work appear in Section 7.

2. BAYESIAN FRAMEWORK

In this section, the probabilistic Bayesian framework is introduced for the considered problem. Prior to that, for a better understanding, the model is briefly explained.

2.1. Target model representation

Starting from the reference image frame, the user selects, within a bounding box, an area containing a target to be tracked in the sequence. The target is modeled at any time t as a joint representation of its global position $\underline{X}_{p,t}$ and of a vectorial shape model $\underline{X}_{s,t}$.

The global target position is initialized as the center of the bounding box containing the target, even though any other point in the image reference system could be chosen. The shape *model* consists of a set of point elements that can be associated with the local object shape information extracted within the bounding box. In our work, corners estimated by a corner extractor [2] are considered local information. The set of M point elements representing the object shape is described as a Generalized Hough Transform (GHT) Table [11, 12] composed by M entries. Each GHT entry, say m, provides the following:

(i) the relative coordinates $\underline{DX}_t^m = (dx_t^m, dy_t^m)$ of the shape element m (*model corner*) with respect to the reference point $\underline{X}_{p,t} = (x_{\mathrm{ref},t}, y_{\mathrm{ref},t})$, considered as a 2D vectorial element of the object shape model. The relative coordinates are computed by subtracting the reference point coordinates from the absolute coordinates of the model element m in the image plane using $dx_t^m = x_t^m - x_{\mathrm{ref},t}$, $dy_t^m = y_t^m - y_{\mathrm{ref},t}$, where the pair $\underline{X}_{c,t}^m(x_t^m, y_t^m)$ is the coordinates of the model element m in the image plane at time t;

Table 1: GHT table as the model at time t.

$(\underline{X}^1_{s,t})$:	dx^1_t	dy^1_t	P^1_t
$(\underline{X}^2_{s,t})$:	dx^2_t	dy^2_t	P^2_t
	$\vdots$		
$(\underline{X}^m_{s,t})$:	dx^m_t	dy^m_t	P^m_t

(ii) a characteristic vector $W^m_t(\underline{DX}^m_t)$ describing additional information associated with each model element.

In this work, the characteristic vector is reduced to a scalar value, $W^m_t(\underline{DX}^m_t) = P^m_t(\underline{DX}^m_t)$, defined as the *persistency* value associated with a model element m. The persistency value represents how stable a given model element is over time in terms of the number of video frames in which it has been supported by performed observations. In general, a characteristic vector $W^m_t(\underline{DX}^m_t)$ allows one to associate different types of local information with the shape element m that can be locally observed at time t such as color, texture, and so forth.

From the above definition, the target shape model (i.e., the GHT) at time t can be defined as

$$\underline{X}_{s,t} = \{\underline{X}^m_{s,t}\}_{1\le m\le M} = \{[\underline{DX}^m_t, W^m_t(\underline{DX}^m_t)]\}_{1\le m\le M}. \quad (1)$$

In this paper, due to the choices explained above, the GHT representing the target shape model can be written as

$$\underline{X}_{s,t} = \{[\underline{DX}^m_t, P^m_t(\underline{DX}^m_t)]\}_{1\le m\le M}. \quad (2)$$

Since the persistency value is a scalar, for simplicity it is considered that $P^m_t(\underline{DX}^m_t) = P^m_t$. Therefore, the model becomes

$$\underline{X}_{s,t} = \{[\underline{DX}^m_t, P^m_t]\}_{1\le m\le M}. \quad (3)$$

Table 1 shows the GHT table $\underline{X}_{s,t}$ as the object shape model.

In this way, the shape model can be considered to be composed by a number of elements that can vary in time. As the shape model is a part of the state of the target, it could be more difficult to apply a mathematical consistency for modeling the dynamic evolution of the object shape in time using such a variable dimension model. Therefore, an alternative, but equivalent, fixed dimension representation is chosen to represent the GHT table.

The GHT at time t is represented as a multivalued 2D matrix (an image) with the same size as the original bounding box, $\underline{W_{X,t}} = (W_{x,t}, W_{y,t})$ containing the object, where $W_{x,t}$ is the dimension along the x axis and $W_{y,t}$ is the dimension along the y axis. The GHT image is centered at the reference point $\underline{X}_{p,t}$. In this way, each integer element r_t of the GHT image is such that $r_t \in \{(i,j) : i \in (-W_{x,t}/2, +W_{x,t}/2), j \in (-W_{y,t}/2, +W_{y,t}/2)\}$. The reference point $\underline{X}_{p,t}$ is associated with the GHT reference point $r^{\text{ref}}_t = (0,0)$. A zero value at any position r_t indicates that no shape model element m is included as a GHT entry m such that $r_t = \underline{DX}^m_t$. On the other hand, if there exists an entry m in the GHT table, then the GHT image value at position $r_t = \underline{DX}^m_t$ will be equal to P^m_t.

0	0	0	0	0	0	0	0	0
0	0	P^1_t	0	0	P^2_t	0	0	0
0	0	0	0	P^3_t	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	P^4_t	0	0	0	0	0	0
0	0	0	0	0	P^5_t	0	0	0
0	0	0	0	0	0	0	0	0

Figure 1: An example of GHT image at time t.

Table 2: The observation set.

$(\underline{Z}^1_t)$:	x^1_t	y^1_t
$(\underline{Z}^2_t)$:	x^2_t	y^2_t
	$\vdots$	
$(\underline{Z}^n_t)$:	x^n_t	y^n_t

As an example, a GHT image obtained from a GHT table with $M = 5$ at time t is shown in Figure 1. In this example, the bounding box is $\underline{W_{X,t}} = (W_{x,t} = 9, W_{y,t} = 7)$, and the reference point is indicated on a gray background.

It is clear that the GHT image can be obtained from the GHT table and vice versa. As a consequence, in the rest of the paper we will refer to the two representations in the same way as the shape model $\underline{X}_{s,t}$ unless the context requires a further specification.

To summarize, the joint representation of the object model at time t is defined as a vector $\underline{X}_t = \{\underline{X}_{s,t}, \underline{X}_{p,t}\}$, of fixed dimension $\dim(\underline{X}_t) = W_{x,t} * W_{y,t} + 2$.

2.2. Observations set representation

In a given frame (time t), an observation set $\underline{Z}_t = \{\underline{Z}^n_t\}_{1\le n\le N} = \{[x^n_t, y^n_t]\}_{1\le n\le N}$ is acquired to constrain tracking results to real data. The observation set $\underline{Z}_t$ is composed of the coordinates in the image plane of each observed local shape element n inside a window of the image, (i.e., the nth corner extracted from an image frame at time t in a local area), with no loss of generality. The size of the window could be fixed at a size at least equal to the GHT image size, $\underline{W_{X,t}} = (W_{x,t}, W_{y,t})$, and it can be considered as centered at a generic reference point $(\underline{X}_{p,t})$. As in the model, one can equivalently speak of an observation set $(\underline{Z}_t)$ and of an observation image (Q) representing the position of the observed shape elements in the absolute image reference system at time t. Table 2 indicates the observation set containing N observed shape elements represented as a table of variable dimension.

In Figure 2. A part $\underline{W_{X,t}} = (W_{x,t} = 9, W_{y,t} = 7)$ of the observation image, centered at a generic reference point

0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0
0	0	1	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	1	0

Figure 2: Window of interest of the observation binary image at time t.

$(\underline{X}_{p,t})$ is shown where a value 1 at locations $\underline{O}_t = \{\underline{O}_t^n \mid \underline{O}_t^n = \underline{Z}_t^n - \underline{X}_{p,t} = (i_t^n, j_t^n) : i_t^n \in (-W_{x,t}/2, +W_{x,t}/2), j_t^n \in (-W_{y,t}/2, +W_{y,t}/2)\}$ indicates the presence of an entry n in the observation set. A zero value is associated with any other location, including all not being contained in the window, but in the whole image. A function q can be defined, associated with image Q, such that $q_{\underline{X}_{p,t}}(\underline{O}_t^n) = q(\underline{O}_t^n + \underline{X}_{p,t}) = q(\underline{Z}_t^n)$, and $q_{\underline{X}_{p,t}}(\underline{O}_t^n) = 1$ and $q_{\underline{X}_{p,t}}(\underline{O}_t^n) = 0$ in the case an entry is present or not, at location $\underline{Z}_t^n$, respectively.

Again, there is a one to one correspondence between the observation set and the observation image. It should be noted that the reference point around which observations are extracted is not to be considered as fixed but it must be estimated as a part of the object status. Therefore, dynamic observation extraction is needed to generate multiple observations images at a given time, t.

2.3. The Bayesian joint position and shape model

The probabilistic framework for the proposed tracking method can be defined starting from the above representations of the model and of the observations. In this paper, both the model variables and the observation variables are considered as random variables. However, no noise model for the used corner extractor will be discussed, as this will be out of the scope of the paper.

In the probabilistic framework used here to specify the presented algorithm, the goal of the tracker is to estimate the posterior $p(\underline{X}_t \mid \underline{Z}_t, \underline{X}_{t-1})$ where $\underline{X}_t$ and $\underline{X}_{t-1}$ are the states of the object at times t and $t-1$, respectively, and $\underline{Z}_t$ is the set of observations at time t. Using Bayesian filtering approach one can write

$$\begin{aligned} p(\underline{X}_t \mid \underline{Z}_t, \underline{X}_{t-1}) &= p(\underline{X}_{p,t}, \underline{X}_{s,t} \mid \underline{Z}_t, \underline{X}_{p,t-1}, \underline{X}_{s,t-1}) \\ &= p(\underline{X}_{s,t} \mid \underline{Z}_t, \underline{X}_{p,t-1}, \underline{X}_{s,t-1}, \underline{X}_{p,t}) \\ &\quad \cdot p(\underline{X}_{p,t} \mid \underline{Z}_t, \underline{X}_{p,t-1}, \underline{X}_{s,t-1}). \end{aligned} \tag{4}$$

Maximizing each of the two terms at the right-hand side of (4) separately provides a suboptimal solution to the problem of maximizing the posterior of $\underline{X}_t$. In this paper, we deal with an approach to obtain such a suboptimal solution. The first term in (4) is related to the shape-based model at the current time conditioned on the availability of observations, previous shape and global motion/position of the object at the current time. The second term is related to the current global position model conditioned on the observations, previous object position, and previous object shape. First, the object global position (tracking) is estimated (Section 2.4). Then, having the object global position, its shape is estimated (updating the model, Section 2.5). These two terms are investigated in the following subsections.

2.4. The global position model

The current position posterior (second term in the right-hand side of (4)) can be written as

$$p(\underline{X}_{p,t} \mid \underline{Z}_t, \underline{X}_{p,t-1}, \underline{X}_{s,t-1}) = \frac{p(\underline{X}_{p,t}, \underline{Z}_t, \underline{X}_{p,t-1}, \underline{X}_{s,t-1})}{p(\underline{Z}_t, \underline{X}_{p,t-1}, \underline{X}_{s,t-1})}. \tag{5}$$

Maximizing the left-hand side of (5) is equivalent to maximizing the numerator of the right-hand side fraction, provided that one considers the denominator as a normalizing constant with respect to $\underline{X}_{p,t}$. It can be shown that, under the hypothesis of independence between shape $\underline{X}_{s,t}$ and global motion $\underline{X}_{p,t}$ of a given tracked object, a Bayesian network of dependencies between involved variables can be written such that the numerator of (5) will become

$$\begin{aligned} &p(\underline{X}_{p,t}, \underline{Z}_t, \underline{X}_{p,t-1}, \underline{X}_{s,t-1}) \\ &\quad = k \cdot p(\underline{X}_{p,t} \mid \underline{X}_{p,t-1}) \cdot p(\underline{Z}_t \mid \underline{X}_{p,t-1}, \underline{X}_{s,t-1}, \underline{X}_{p,t}). \end{aligned} \tag{6}$$

Again, the global position model is decomposed into two terms: the position prediction model (global motion model, first term) and the observation model (second term). The variable k in (6) is a constant, and if we consider the shape to be independent from the global position and motion, k can be written as

$$k = p(\underline{X}_{p,t-1}) \cdot p(\underline{X}_{s,t-1} \mid \underline{X}_{p,t}, \underline{X}_{p,t-1}) = p(\underline{X}_{p,t-1}) \cdot p(\underline{X}_{s,t-1}). \tag{7}$$

In this paper, the position prediction model $p(\underline{X}_{p,t} \mid \underline{X}_{p,t-1})$ emphasizes two major points: (a) the object cannot move faster than a given speed (in pixels), and (b) defining different prediction models gives different weights to different global object positions in the plane. For example, in this paper a simple global position prediction model of a uniform windowed type kernel is used such that

$$K_{\text{gp}}(\underline{X}_{p,t}, \underline{X}_{p,t-1}) = \begin{cases} 1 & \text{if } \left(|x_{p,t} - x_{p,t-1}| \le \dfrac{W_x}{2}\right) \\ & \wedge \left(|y_{p,t} - y_{p,t-1}| \le \dfrac{W_y}{2}\right), \\ 0 & \text{elsewhere,} \end{cases} \tag{8}$$

where K_{gp} is a rectangular uniform kernel centered on $\underline{X}_{p,t-1}$ (the object previous position), and it gives the value one to each point inside the kernel, and the value zero to the points outside the kernel. This kernel guarantees that the new object global position lies inside the kernel, and hence the kernel limits the object speed. The subscript gp indicates the "global position" term. Dimension of this kernel is defined as

$$W_{\text{gp}} = \dim(K_{\text{gp}}) = W_x \cdot W_y, \tag{9}$$

where W_x and W_y are scalar dimensions along x and y of K_{gp}. Now, using (8) and (9) the global position prediction model is defined:

$$p(\underline{X}_{p,t} \mid \underline{X}_{p,t-1}) = \begin{cases} \dfrac{1}{W_{\text{gp}}} & \text{if } K_{\text{gp}}(\underline{X}_{p,t}, \underline{X}_{p,t-1}) = 1, \\ 0 & \text{elsewhere.} \end{cases} \tag{10}$$

Formula (10) defines a uniform probability for all positions inside the kernel to be considered as the new object position. This is done by assuming that the movement of the object in two successive frames is such that the two consequent objects appearances overlap each other. It is possible to give different weights to different positions inside the kernel by defining a different probability or a different kernel (e.g., a Gaussian kernel).

Having the probability assigned to every point in the kernel, the probability of the observation model (second term in (4)) is defined in this paper as follows:

$$p(\underline{Z}_t \mid \underline{X}_{p,t-1}, \underline{X}_{s,t-1}, \underline{X}_{p,t}) = \frac{1}{W_Z} \cdot K_Z(\underline{X}_{p,t}, \underline{Z}_t, \underline{X}_{s,t-1}, \underline{X}_{p,t-1}),$$

$$\text{where } W_Z = \sum_{Z} K_Z(\underline{X}_{p,t}, \underline{Z}_t, \underline{X}_{s,t-1}, \underline{X}_{p,t-1}),$$

$$K_Z(\underline{X}_{p,t}, \underline{Z}_t, \underline{X}_{s,t-1}, \underline{X}_{p,t-1}) = \frac{e^{V_t(\underline{X}_{p,t}, \underline{Z}_t, \underline{X}_{s,t-1}, \underline{X}_{p,t-1})} - 1}{e^{V_t(\underline{X}_{p,t}, \underline{Z}_t, \underline{X}_{s,t-1}, \underline{X}_{p,t-1})}}, \tag{11}$$

where $V_t(\underline{X}_{p,t}, \underline{Z}_t, \underline{X}_{s,t-1}, \underline{X}_{p,t-1})$ is the number of votes (see Section 3.1) to a potential object position (see (12)) at time t and the related observation model computed for the set of elements found at step t. W_Z is the summation of all possible observations configurations based on the previous shape model, and the previous and the current object position. Note that the current object position in the current time is a variable for which different observations configurations are obtained. The function $K_Z(\cdot)$ is a kernel on the shape subspace that filters the observations based on different possible object positions. The goal is to maximize the multiplication of (10) and (11) to maximize (6).

Equation (11) says that if there is no vote for a given position ($V_t(\underline{X}_{p,t}, \underline{Z}_t, \underline{X}_{s,t-1}, \underline{X}_{p,t-1}) = 0$), the probability value will equal zero. Instead, if the number of votes for a given position tends to infinity, the probability value will be equal to one. The choice of the probability of the global elements configuration (11) is to some extent heuristic, and it is possible that one selects other functions using the same general model. However, different probabilistic models that can be chosen should be developed starting from a function that relates the whole set of observations and the global object position, at a given time t, by using knowledge of the whole shape model (i.e., the GHT table) available at time $t-1$. The function $V_t(\underline{X}_{p,t}, \underline{Z}_t, \underline{X}_{s,t-1}, \underline{X}_{p,t-1})$ is defined here as follows:

$$V_t(\underline{X}_{p,t}, \underline{Z}_t, \underline{X}_{s,t-1}, \underline{X}_{p,t-1}) = \sum_{n=1}^{N} S_n(\underline{X}_{p,t}, \underline{Z}_t^n, \underline{X}_{s,t-1}, \underline{X}_{p,t-1}), \tag{12}$$

where N is the number of observation elements. $S_n(\cdot)$ is a function of a given hypothesized object position $\underline{X}_{p,t}$ and an observation element. In other words, every single observation contributes to a probable object position according to $S_n(\cdot)$ by increasing the probability of that position. However, the influence of each observation is modulated by its relative position with respect to shape model elements at time $t-1$. As a consequence, the function $S_n(\cdot)$ is defined as follows:

$$S_n(\underline{X}_{p,t}, \underline{Z}_t^n, \underline{X}_{s,t-1}, \underline{X}_{p,t-1}) = \sum_{m=1}^{M} K_R(d_{m,n}(\underline{X}_{s,t-1}^m, \underline{Z}_t^n)). \tag{13}$$

In formula (13), M is the number of the model elements at time $t-1$. The right side of the equation shows that every model element contributes to increase the probability of a given global object position with respect to a metric $d_{m,n}$. The variable $d_{m,n}$ evaluates the distance between the shifted value of a model element m at time $t-1$ and an observation element n at time t. This shift is done by the possible shifts of the whole object to the new position $\underline{X}_{p,t}$. This means that, for a rigid object, the movement of each single given corner is equal to the movement of the whole object. For a nonrigid object, this movement will be slightly different. Therefore, the higher the local nonrigidity of the object is, the larger the difference between the motion of the whole object and the motion of every single corner will be. Hence, a rigidity measure can be defined using the metric $d_{m,n}$. In the aforementioned situation, if the distance between the shifted value of a model element and an observation element falls within the radius R_R of a rigidity kernel $K_R(\cdot)$, it will contribute to $V_t(\cdot)$:

$$d_{m,n}(\underline{X}_{s,t-1}^m, \underline{Z}_t^n) = \left|\left|\underline{X}_{c,t-1}^m + \underline{d}_{\text{ref},t}(\underline{X}_{p,t}) - \underline{Z}_t^n\right|\right|^2, \tag{14}$$

where $\underline{d}_{\text{ref},t}$ is the motion of the object at time t to the new position $\underline{X}_{p,t}$. Since the new position has not been fixed yet, for every possible $\underline{X}_{p,t}$ a different motion vector can be computed:

$$\underline{d}_{\text{ref},t}(\underline{X}_{p,t}) = \underline{X}_{p,t} - \underline{X}_{p,t-1}. \tag{15}$$

Therefore, one can write

$$\begin{aligned} &d_{m,n}(\underline{X}_{s,t-1}^m, \underline{Z}_t^n) \\ &\quad = \left|\left|\underline{X}_{c,t-1}^m + \underline{d}_{\text{ref},t}(\underline{X}_{p,t}) - \underline{Z}_t^n\right|\right|^2 \\ &\quad = \left|\left|(\underline{DX}_{t-1}^m + \underline{X}_{p,t-1}) + (\underline{X}_{p,t} - \underline{X}_{p,t-1}) - (\underline{DX}_t^n + \underline{X}_{p,t})\right|\right|^2 \\ &\quad = \left|\left|\underline{DX}_{t-1}^m - \underline{DX}_t^n\right|\right|^2 = d_{m,n}(\underline{X}_{s,t-1}^m, \underline{X}_{s,t}^n). \end{aligned} \tag{16}$$

The choice of the kernel in (13) is a heuristic issue. Note that if a Kronecker delta kernel (17) is used, then (12) also becomes the definition of a fixed null rotation generalized Hough transform using $\underline{X}_{s,t}$ as a GHT table. While if a uniform kernel (18) is introduced, a regularized voting area (see Algorithm 1) is used within the GHT to allow

```
for n = 1 to N
    for m = 1 to M
        v_x(t) = x_t^n − dx_{t−1}^m
        v_y(t) = y_t^n − dy_{t−1}^m
      for r_x, r_y = − 2 to 2
        V_t(v_x(t) + r_x, v_y(t) + r_y) = V_t(v_x(t) + r_x, v_y(t) + r_y) + 1
      end
    end
end
```

Algorithm 1: Voting procedure and regularization.

local distortion effects to be compensated during global position estimation. A Gaussian kernel (19) also allows effects to be compensated during global position estimation, but it gives different weights to different positions lied in the radius of the kernel. If a Gaussian kernel is used in the voting procedure (formulae (11) to (13)), different positions inside the kernel will have different influences on the probability function in (11). Therefore, for the aforementioned voting framework a uniform kernel has been used to allow distortion in any given direction within the radius of the kernel without prejudging on the weight of the position. The three different types of the kernels are defined as follows:

$$K_\delta(d_{m,n}) = \delta(d_{m,n}) = \begin{cases} 1 & \text{if } d_{m,n} = 0, \\ 0 & \text{if } d_{m,n} \neq 0, \end{cases} \tag{17}$$

$$K_U(d_{m,n}) = \begin{cases} 1 & \text{if } d_{m,n} \leq R_R, \\ 0 & \text{otherwise}, \end{cases} \tag{18}$$

$$K_G(d_{m,n}) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{d_{m,n}}{2\sigma^2}\right). \tag{19}$$

In (19), σ^2 is the variance of the Gaussian kernel. The delta function does not allow nonrigidity in the object, while a uniform kernel allows every single model element to contribute to multiple possible global position hypotheses within a radius of R_R. For this reason, in this work, a uniform kernel has been chosen with limited extension, as it is presumed that relatively slow frame-to-frame movements of objects of interest will occur, and they will allow only small local shape distortions. In this way, the proposed suboptimal algorithm looks at the solution as the value $\underline{X}_{p,t} = \underline{X}^*_{p,t}$ that maximizes the product in (6).

2.5. The shape-based model

Having found the new estimated global position of the object, the shape must be estimated. This means to apply a strategy to maximize the probability of the posterior $p(\underline{X}_{s,t} \mid \underline{Z}_t, \underline{X}_{p,t-1}, \underline{X}_{s,t-1}, \underline{X}_{p,t})$ (first term at the right-hand side of (4)) where all terms in the conditional part have been fixed. With a reasoning approach similar to the one related to (5) and (6), one can write

$$P(\underline{X}_{s,t} \mid \underline{Z}_t, \underline{X}_{p,t-1}, \underline{X}_{s,t-1}, \underline{X}_{p,t}) = \frac{p(\underline{X}_{s,t}, \underline{Z}_t, \underline{X}_{p,t-1}, \underline{X}_{s,t-1}, \underline{X}_{p,t})}{p(\underline{Z}_t, \underline{X}_{p,t-1}, \underline{X}_{s,t-1}, \underline{X}_{p,t})}, \tag{20}$$

$$p(\underline{X}_{s,t}, \underline{Z}_t, \underline{X}_{p,t-1}, \underline{X}_{s,t-1}, \underline{X}_{p,t}) = k' \cdot p(\underline{X}_{s,t} \mid \underline{X}_{s,t-1}) \cdot p(\underline{Z}_t \mid \underline{X}_{s,t}, \underline{X}_{p,t-1}, \underline{X}_{s,t-1}, \underline{X}_{p,t}), \tag{21}$$

$$k' = p(\underline{X}_{s,t-1}) \cdot p(\underline{X}_{p,t}, \underline{X}_{p,t-1} \mid \underline{X}_{s,t}, \underline{X}_{s,t-1}) = p(\underline{X}_{s,t-1}) \cdot p(\underline{X}_{p,t}, \underline{X}_{p,t-1}), \tag{22}$$

where the first term at the right-hand side of (21) is the shape prediction model, and the second term is the shape updating observation model. The variable k' in (22) is a constant, and if we consider the shape to be independent from the object global position and motion, k' can be defined as (22). Assuming small changes in the object shape in two successive frames and assuming motion to be independent from shape and its local variations, it is reasonable to have the shape at time t be similar to the shape at time $t-1$. Therefore, using a shape kernel in a similar way as in (8) between the elements of the shape models in two successive frames one can write

$$p(\underline{X}_{s,t} \mid \underline{X}_{s,t-1}) = \begin{cases} \frac{1}{W_{\text{gs}}} \cdot K_{\text{gs}}(\underline{X}_{st}, \underline{X}_{st-1}) & \text{if } K_{\text{gs}}(\underline{X}_{s,t}, \underline{X}_{s,t-1}) > 0, \\ 0 & \text{elsewhere}, \end{cases} \tag{23}$$

$$W_{\text{gs}} = \sum_{\underline{X}_{s,t}} K_{\text{gs}}(\underline{X}_{s,t}, \underline{X}_{s,t-1}), \tag{24}$$

where W_{gs} in (24) is the summation of the global shape kernels in the shape subspace containing just those elements of the predicted object shape model that lay into the kernel. In other words, W_{gs} is the partition function used to normalize the kernel to become a probability. The subscript gs stands for the term "global shape". Formula (23) results in a nonuniform probability for different predicted shape models. Although (23) can be written as one term,

$p(\underline{X}_{s,t} \mid \underline{X}_{s,t-1}) = K_{\mathrm{gs}}(\underline{X}_{s,t}, \underline{X}_{s,t-1})/W_{\mathrm{gs}}$, separating the two conditions in (23) emphasizes that only the possible shape configurations that lay within the kernel are considered. The kernel used in (23) is a kernel defined on the shape space:

$$K_{\mathrm{gs}}(\underline{X}_{s,t}, \underline{X}_{s,t-1}) = \prod_{m} K_{\mathrm{ls},m}(\underline{X}^m_{s,t}, \eta^m_{s,t}), \tag{25}$$

$$\eta^m_{s,t} = \{\underline{X}^j_{s,t-1} : |\underline{DX}^m_t - \underline{DX}^j_{t-1}| \le R_R\}. \tag{26}$$

$\underline{\eta}^m_{s,t}$ is the set of all model elements at time $t-1$ that lay inside the rigidity kernel K_R with the radius R_R centered on $\underline{X}^m_{s,t}$. The predicted shape model may have different configurations satisfying (25). If one considers the kernel in (25) as generated independently by m elements of the shape model, then the global shape kernel K_{gs} can be written in terms of the local shape kernel $K_{\mathrm{ls},m}$ of each model element as

$$K_{\mathrm{gs}}(\underline{X}_{s,t}, s\underline{X}_{s,t-1}) = \prod_{m} K_{\mathrm{ls},m}(\underline{X}^m_{s,t}, \eta^m_{s,t}), \tag{27}$$

$$\begin{aligned} &K_{\mathrm{ls},m}(\underline{X}^m_{s,t}, \eta^m_{s,t}) \\ &\quad = \sum_{j:\underline{X}^j_{s,t-1} \in \eta^m_{s,t}} K^j_{\mathrm{ls},m}(\underline{X}^m_{s,t}, \underline{X}^j_{s,t-1}) \\ &\quad = \sum_{j:\underline{X}^j_{s,t-1} \in \eta^m_{s,t}} K_R(d_{m,j}(\underline{X}^m_{s,t}, \underline{X}^j_{s,t-1})) \cdot K^j_{P,m}(\underline{X}^m_{s,t}, \underline{X}^j_{s,t-1}), \end{aligned} \tag{28}$$

$$K^j_{P,m}\left(\underline{X}^m_{s,t}, \underline{X}^j_{s,t-1}\right) = \begin{cases} 1 & \text{if } (P^m_t - P^j_{t-1}) \in R^{j(m)}_P, \\ 0 & \text{elsewhere,} \end{cases} \tag{29}$$

$$R^m_P = \bigcup_j R^{j(m)}_P. \tag{30}$$

In (28), the local shape kernel has been related to $\eta^m_{s,t}$ and therefore to the rigidity kernel radius R_R. The local nonrigidity of the model elements could presumably be due to the shape distortion of every single element in the GHT table, since a noise-free corner extractor with no displacement of corner position is supposed here. Despite the fact that they can represent interesting extensions of the method proposed here, corner extractor models are out of the scope of this paper. Moreover, the local shape kernel has also been related to the rigidity kernel and the persistency kernel. The rigidity kernel used in (28) is a Gaussian kernel (19) that, like the uniform kernel (18), allows some degree of local distortion, but, unlike the uniform kernel, it gives different weights to different positions based on their distance from the center of the kernel. This is useful when there is more than one model element in the neighborhood of a given observation. In this case, the model element closest to the observation is considered to be the same element at time $t-1$ that has been distorted to the new position at time t. Therefore, the Gaussian kernel is used in this work in the shape updating phase (formula (60)). In (29), R_P is the radius of the persistency, P, kernel. The persistency kernel is related to the persistency values of the elements within two successive frames. This kernel implies that, in addition to the local nonrigidity of the elements, the persistency values of shape model elements at two consequent time instants must also have a relation to each other with respect to their associated characteristic vectors. This is because although the persistency value of a new model element that is included in the GHT table is set to an initial value P_I, when the element first appears, it evolves during time to allow the whole shape model to adapt to new 2D shapes assumed by the object with respect to the sensor. Therefore, to realize the possible changes in the persistency value of a given model element, within two successive frames, different cases of the persistency values have to be considered. All possible changes in the persistency values between two following frames form the persistency kernel. To make the persistency kernel as time-invariant as possible with respect to persistency values, the changes are introduced as the difference of the persistency values:

$$R^{j(m)}_P = \{P^m_t - P^j_{t-1} \mid \ \forall j : \underline{X}^j_{s,t-1} \in \eta^m_{s,t}\}. \tag{31}$$

Considering Figures 1 and 2, five different cases for the model elements are defined as follows, by considering the two possible events that can occur, that is, an observation is either present or not at the position $\underline{O}^m_t$.

(i) A given element $j \neq m$ exists in the model at time $t-1$ with a persistency value greater than or equal to a threshold P_{th}. In case an observation is present at $\underline{O}^m_t$ at time t ($q_{\underline{X}^*_{p,t}}(\underline{O}^m_t) = 1$), the persistency of the shape model element m will be the one of point j increased by one, in case it will be decided that a local shape distortion is moving point j to the point m in the interval between time $t-1$ and time t. Therefore,

$$\begin{aligned} &\text{if } (P^{j\neq m}_{t-1} \ge P_{\mathrm{th}}),\ (\underline{X}^j_{s,t-1} \in \eta^m_{s,t}),\ (q_{\underline{X}^*_{p,t}}(\underline{O}^m_t) = 1), \\ &\text{then } R^{j(m)}_P = P^m_t - P^j_{t-1} = (P^j_{t-1} + 1) - P^j_{t-1} = 1. \end{aligned} \tag{32}$$

(ii) A given element m exists in the model at time $t-1$ with a persistency value greater than or equal to a threshold P_{th}. In case an observation is present at $\underline{O}^m_t$ at time t ($q_{\underline{X}^*_{p,t}}(\underline{O}^m_t) = 1$), the persistency of the shape model element m will be the one of point m increased by one, in case it is decided that no local distortion will occur at m, and the shape model is reinforced by the observation. Therefore,

$$\begin{aligned} &\text{if } (P^m_{t-1} \ge P_{\mathrm{th}}),\ (q_{\underline{X}^*_{p,t}}(\underline{O}^m_t) = 1), \\ &\text{then } R^{m(m)}_P = P^m_t - P^m_{t-1} = (P^m_{t-1} + 1) - P^m_{t-1} = 1. \end{aligned} \tag{33}$$

(iii) A given element m exists in the model at time $t-1$ with a persistency value greater than a threshold P_{th}. In case no observation is present at $\underline{O}^m_t$ at time t ($q_{\underline{X}^*_{p,t}}(\underline{O}^m_t) = 0$), the persistency of the shape model element m will be decreased by one, as no observation will support the presence of the shape element already estimated at time $t-1$ at position m. Therefore,

$$\begin{aligned} &\text{if } (P^m_{t-1} > P_{\mathrm{th}}),\ (q_{\underline{X}^*_{p,t}}(\underline{O}^m_t) = 0), \\ &\text{then } R^{m(m)}_P = P^m_t - P^m_{t-1} = (P^m_{t-1} - 1) - P^m_{t-1} = -1, \end{aligned} \tag{34}$$

however, in the event that a local distortion occurs, moving the point m at $t-1$ to a position $j \neq m$ at time t, the persistency of the shape model element m can be decided to be decreased to zero (i.e., the model element can disappear at time t). Therefore,

$$\begin{aligned} &\text{if } \left(P_{t-1}^{m} > P_{\text{th}}\right),\ \left(q_{\underline{X}_{p,t}^{*}}\left(\underline{O}_{t}^{m}\right) = 0\right), \\ &\text{then } R_{P}^{m(m)} = P_{t}^{m} - P_{t-1}^{m} = 0 - P_{t-1}^{m} = -P_{t-1}^{m}. \end{aligned} \tag{35}$$

(iv) A given element m exists in the model at time $t-1$ with a persistency value equal to the threshold. In case no observation is present at $\underline{O}_{t}^{m}$ at time t, the persistency of the shape model element m will be decreased to zero (i.e., the model element will disappear at time t), as no observation will support the presence of the shape element already estimated as the very uncertain position m at time $t-1$. Therefore,

$$\begin{aligned} &\text{if } \left(P_{t-1}^{m} = P_{\text{th}}\right),\ \left(q_{\underline{X}_{p,t}^{*}}\left(\underline{O}_{t}^{m}\right) = 0\right), \\ &\text{then } R_{P}^{m(m)} = P_{t}^{m} - P_{t-1}^{m} = 0 - P_{\text{th}} = -P_{\text{th}}. \end{aligned} \tag{36}$$

(v) A given element m does not exist in the model at time $t-1$ (its persistency value equals to zero). In case an observation is present at $\underline{O}_{t}^{m}$ at time t ($q_{\underline{X}_{p,t}^{*}}(\underline{O}_{t}^{m}) = 1$), a new shape model element will be created at $\underline{O}_{t}^{m}$ with a minimal initial certainty represented by P_I. Therefore,

$$\begin{aligned} &\text{if } \left(P_{t-1}^{m} = 0\right),\ \left(q_{\underline{X}_{p,t}^{*}}\left(\underline{O}_{t}^{m}\right) = 1\right), \\ &\text{then } R_{P}^{m(m)} = P_{t}^{m} - P_{t-1}^{m} = P_{I} - 0 = P_{I}. \end{aligned} \tag{37}$$

(vi) A given element m does not exist in the model at time $t-1$ (its persistency value equals to zero). In case no observation is present at $\underline{O}_{t}^{m}$ at time t and no shape model element is present at m at time $t-1$, no new shape element will be created as a consequence of this configuration (note that in case of no shape element at m at time $t-1$, an element can be always created at m at time t also in this configuration due to a decision in favor of a local distortion from an adjacent point j at time $t-1$). Therefore,

$$\begin{aligned} &\text{if } \left(P_{t-1}^{m} = 0\right),\ \left(q_{\underline{X}_{p,t}^{*}}\left(\underline{O}_{t}^{m}\right) = 0\right), \\ &\text{then } R_{P}^{m(m)} = P_{t}^{m} - P_{t-1}^{m} = 0. \end{aligned} \tag{38}$$

Consequently, the values that the persistency radius can take are

$$\begin{aligned} R_{P}^{m} &= R_{P}^{j(m), j \neq m} \bigcup R_{P}^{m(m)} \\ &= \{1\} \bigcup \{1, -1, P_{\text{th}}, P_{I}, 0, -P_{t-1}^{m}\} \\ &= \left\{1, -1, P_{\text{th}}, P_{I}, 0, -P_{t-1}^{m}\right\}. \end{aligned} \tag{39}$$

The set R_P^m depends only on the location due to the presence of the element $-P_{t-1}^{m}$. This dependency arises from the fact that shape distortion is allowed in every point of the GHT image. According to the above model, only a finite set (even though quite large) of possible new GHTs ($\underline{X}_{s,t}$) can be obtained as equally probable from the GHT table at the previous time $t-1$. This step, therefore, represents a shape model prediction phase. After prediction of the shape model at time t, the shape model can be updated by an appropriate observation model that can be used to filter the finite set of possible new GHTs, and to select one of the possible predicted new shape models ($\underline{X}_{s,t}$). To this end, the second term in (21), that is, the shape updating observation model $p(\underline{Z}_t \mid \underline{X}_{s,t}, \underline{X}_{p,t-1}, \underline{X}_{s,t-1}, \underline{X}_{p,t})$ is used. Again, consider Figure 2 (observation set binary image), but with the difference that this time the global object position has been estimated and it is fixed, $\underline{X}_{\text{ref},t} = \underline{X}_{p,t}^{*}$. Since the observation set is independent from the corner extractor, its probability can be written as a product of two terms:

$$\begin{aligned} &p\left(\underline{Z}_t \mid \underline{X}_{s,t}, \underline{X}_{p,t-1}, \underline{X}_{s,t-1}, \underline{X}_{p,t}\right) \\ &\quad = \prod_{\underline{Z}_t^n \in \underline{Z}_t} p\left(q\left(\underline{Z}_t^n\right) = 1 \mid \underline{X}_{p,t-1}, \underline{X}_{s,t-1}, \underline{X}_{s,t}, \underline{X}_{p,t}^{*}\right), \\ &\prod_{\underline{Z}_t^n \in \underline{Z}_t} p\left(q\left(\underline{Z}_t^n\right) = 0 \mid \underline{X}_{p,t-1}, \underline{X}_{s,t-1}, \underline{X}_{s,t}, \underline{X}_{p,t}^{*}\right). \end{aligned} \tag{40}$$

Since $\underline{O}_t^n$ is the pair of the relative coordinates of the observation $\underline{Z}_t^n$ (Section 2.2) with respect to the fixed estimated global object position $\underline{X}_{\text{ref},t} = \underline{X}_{p,t}^{*}$, $\underline{O}_t^n$ and $\underline{Z}_t^n$ are equivalent and they can be used alternately. Consequently, $\underline{O}_t$ and $\underline{Z}_t$ will also become equivalent, as $\underline{O}_t = \{\underline{O}_t^n \mid \underline{O}_t^n = \underline{Z}_t^n - \underline{X}_{p,t}^{*}\}$. Therefore, (40) becomes

$$\begin{aligned} &p\left(\underline{Z}_t \mid \underline{X}_{s,t}, \underline{X}_{p,t-1}, \underline{X}_{s,t-1}, \underline{X}_{p,t}\right) \\ &\quad = \prod_{\underline{O}_t^n \in \underline{O}_t} p\left(q_{\underline{X}_{p,t}^{*}}\left(\underline{O}_t^n\right) = 1 \mid \underline{X}_{s,t}, \underline{X}_{s,t-1}, \underline{X}_{p,t}^{*}, \underline{X}_{p,t-1}\right), \\ &\prod_{\underline{O}_t^n \notin \underline{O}_t} p\left(q_{\underline{X}_{p,t}^{*}}\left(\underline{O}_t^n\right) = 0 \mid \underline{X}_{s,t}, \underline{X}_{s,t-1}, \underline{X}_{p,t}^{*}, \underline{X}_{p,t-1}\right). \end{aligned} \tag{41}$$

As explained in Section 2.2, the observation image can be represented as a set of zero-valued points (that are not a part of the observations set) and one-valued positions (the observations set elements). These two sets partition the windowed part of the image into two areas (zero-valued and one-valued partitions). To show the interindependency (the independency among the two partitions) and the intraindependency (the independency among the elements of each partition) to verify (40), one can investigate the two cases separately (using (32)–(39)) and write the following formulae to describe the observation model for shape updating:

$$R_{P}^{n,1}\left(\underline{\eta}_{s,t}^{n}\right) = \begin{cases} P_I & \text{if } \forall\, j : \underline{X}_{s,t-1}^{j} \in \underline{\eta}_{s,t}^{n} \\ & P_{t-1}^{j} = 0,\ q_{\underline{X}_{p,t}^{*}}\left(\underline{O}_t^n\right) = 1, \\ 1 & \text{if } \exists\ j : \underline{X}_{s,t-1}^{j} \in \underline{\eta}_{s,t}^{n} \\ & P_{t-1}^{j} \geq P_{\text{th}},\ q_{\underline{X}_{p,t}^{*}}\left(\underline{O}_t^n\right) = 1, \end{cases} \tag{42}$$

$$R_{P}^{n,0}\left(\underline{\eta}_{s,t}^{n}\right) = \begin{cases} 0 & \text{if } P_{t-1}^{n} = 0,\ q_{\underline{X}_{p,t}^{*}}\left(\underline{O}_t^n\right) = 0, \\ -1 & \text{if } P_{t-1}^{n} > P_{\text{th}},\ q_{\underline{X}_{p,t}^{*}}\left(\underline{O}_t^n\right) = 0, \\ -P_{t-1}^{n} & \\ -P_{\text{th}} & \text{if } P_{t-1}^{n} = P_{\text{th}},\ \ q_{\underline{X}_{p,t}^{*}}\left(\underline{O}_t^n\right) = 0, \end{cases}$$

Table 3: Kernels that have been used in the implemented algorithms (described in the next sections) to estimate different probabilities in (6) and (21), along with the related probability terms, related kernel radii, and a short explanation about the effect of each kernel on the model.

Probability	Kernel		Radius	Note
$p(\underline{X}_{p,t} \mid \underline{X}_{p,t-1})$	K_{gp}		$R_{\text{gp}} = \dfrac{\sqrt{W_x^2 + W_y^2}}{2}$	Global uncertainty over the motion of the whole object
$p(\underline{Z}_t \mid \underline{X}_{p,t-1}, \underline{X}_{s,t-1}, \underline{X}_{p,t})$	$K_R = K_U$		$R_R = 2\sqrt{2}$ R_R is a constant value	Uncertainty in shape distortion of single GHT elements from one frame to the next (rigidity of observations with respect to the old shape)
$p(\underline{X}_{s,t} \mid \underline{X}_{s,t-1})$	K_{gs} $K_{\text{ls},m}$ $K^j_{\text{ls},m}$	$K_R = K_G$	$R_R = 2\sqrt{2}$	Uncertainty in shape distortion of single GHT elements from one frame to the next (rigidity of observations with respect to the old shape)
		K_P	$R_P^m = R_P^{j(m),j\neq m} \bigcup R_P^{m(m)}$, $R_P^{j(m),j\neq m} = \{1\}$, $R_P^{m(m)} = \{1, -1, P_{\text{th}} = 1, P_I = 5, 0, -P_{t-1}^m\}$	A set of possible differences between old persistency values in a neighborhood of a point m with respect to values that can be obtained at time t in point m
$p(\underline{Z}_t \mid \underline{X}_{s,t}, \underline{X}_{p,t-1}, \underline{X}_{s,t-1}, \underline{X}_{p,t})$	$K_R = K_G$		$R_R = 2\sqrt{2}$	Uncertainty in shape distortion of single GHT elements from one frame to the other (rigidity of observations with respect to the old shape)
		K_P	$R_P^n = R_P^{n,0}(\underline{\eta}_{s,t}^n) \bigcup R_P^{n,1}(\underline{\eta}_{s,t}^n) = \{1, -1, P_{\text{th}}, P_I, 0, -P_{t-1}^n\}$	A set of possible differences between old persistency values in a neighborhood of a point m with respect to values that can be obtained at time t in point m. The difference from the previous case (the last row) is that the kernel values are partitioned in another way

where $R_P^{n,0}$ and $R_P^{n,1}$ are a posteriori kernel sets that indicate which possible values of the searched solution are possible after one has observed whether an observation is present ($R_P^{n,1}$) at location n, or it is not present ($R_P^{n,0}$). As it can be seen from (42):

$$R_P^{n,0}\left(\underline{\eta}_{s,t}^n\right) \bigcap R_P^{n,1}\left(\underline{\eta}_{s,t}^n\right) = \varnothing, \tag{43}$$

$$R_P^{n,0}\left(\underline{\eta}_{s,t}^n\right) \bigcup R_P^{n,1}\left(\underline{\eta}_{s,t}^n\right) = \{1, -1, P_{\text{th}}, P_I, 0, -P_{t-1}^n\} = R_P^n. \tag{44}$$

This means that the probability of having an observation in a point n can be associated with a set (43) of possible new shape model elements at n, whose intersection is null with respect to the set of possible new shape model elements that can be obtained if no observation is present at n. On the other hand, the union of the two a posteriori sets for each element n is equal to the a priori kernel set (31) of shape elements that can be reached if no observation is done.

On the basis of above definitions, the two terms in (41) can be written as

$$p\left(q_{\underline{X}^*_{p,t}}(\underline{O}_t^n) = 1 \mid \underline{X}_t^n : (P_t^n - P_{t-1}^j) \in R_P^{n,1}\left(\underline{\eta}_{s,t}^n\right), \underline{X}_{s,t-1}^j \in \underline{\eta}_{s,t}^n\right) = 1, \tag{45}$$

$$p\left(q_{\underline{X}^*_{p,t}}(\underline{O}_t^n) = 0 \mid \underline{X}_t^n : |P_{t-1}^j - P_t^n| \in R_P^0\left(\underline{\eta}_{s,t}^n\right), \underline{X}_{s,t-1}^j \in \underline{\eta}_{s,t}^n\right) = 1. \tag{46}$$

As one has to search for the most probable solution of (21) after having observed corners $\underline{Z}_t$ at time t, it comes

out that the set of possible solutions of shape updating (i.e., the optimal updated GHT table $\underline{X}^*_{s,t}$) can be found by jointly maximizing (23) and (40). As in both cases, one can easily understand that there exist a limited number of multiple solutions, both in case of having observed a corner is present or it is not present (from (32) to (38)) for the shape prediction, and from (42) for the observation model), then it can be deduced that the maximum can be obtained only for those values that belong to both solution sets.

For example, if an observation is present, the set of solutions maximizing (28) is given by S_1 that is represented by all possible persistency values that can be attributed to a local shape model element at time t, given the previous persistency values in its neighborhood at time $t-1$ that provide a kernel value equal to 1 in (29). At the same time, a configuration, where observation element $\underline{Z}^n_t$ is present, cannot be maximized unless (45) holds. The set of new shape model elements that satisfy (45) is equal to S_2. As a consequence, the set of possible persistency values that can be associated with the mth shape model elements is given by the intersection of the two solution sets:

$$\begin{aligned} S_1 &= \arg\max_{\underline{X}^m_{s,t}} K_{\mathrm{ls},m}\left(\underline{X}^m_{s,t}, \eta^m_{s,t}\right) \\ &= \left\{P^m_t : \left(P^m_t - P^j_{t-1}\right) \in R_P^{j(m)}, X^{j(m)}_{s,t-1} \in \eta^m_{s,t}\right\}, \\ S_2 &= \arg\max_{\underline{X}^m_{s,t}} p\left(\underline{Z}_t \mid \underline{X}_{s,t}, \underline{X}_{p,t-1}, \underline{X}_{s,t-1}, \underline{X}_{p,t}\right) \\ &= \left\{P^m_t : \left(P^m_t - P^j_{t-1}\right) \in R_P^{n,1}\left(\underline{\eta}^n_{s,t}\right)\right\}, \\ S &= S_1 \bigcap S_2. \end{aligned} \tag{47}$$

A similar reasoning by substituting (45) with (46) can be achieved if no observation is present.

For the aforementioned system, there can be multiple solutions characterized by the same maximum probability value. To select among them, it is possible to pick up randomly one for each model element or to design some alternative strategy. In the current work, a windowed Gaussian kernel is used in (28) that is cut to zero for location farther than the rigidity kernel, whose radius is here fixed to $R_R = 2\sqrt{2}$. In this way, a first ordering in set S_1 can be used that ranks possible solutions on the basis of the rigidity kernel function value computed at a distance value between the model location m with respect to the nth observation. In this way, low shape distortion solutions are preferred with respect to ones that cause a higher distortion. In every case, as in a discrete image, there may exist more points with the same distance from a given point (e.g., points belonging to a 4-neighborhood with respect to the center). In such cases, a random choice over a restricted set can be still necessary as a second ordering step.

2.6. Model discussion

As can be seen in the above discussion (and synthesized from Table 3), the proposed approach can be considered as a two-stage hierarchical Bayesian model, where the global object position is considered first and the shape model is then estimated. Each stage can be represented as a prediction-updating model that uses different situation-dependent kernels. In this paper, the kernels indicated in Table 3 have been used and the algorithms described in the next sections have been applied to perform the estimation. In the algorithms that will be described, some parts will be introduced that cannot be fully included in the above descriptions (and that can therefore appear as a heuristic selection of parameters). In particular, this is true for Section 4, where multiple possible motions of different object subparts are assumed. For the sake of simplicity, this assumption is not fully described in the above Bayesian model. However, it is reasonable to say that these parts can be considered as direct, albeit nontrivial, extensions of the above theory.

3. OBJECT POSITION ESTIMATION

The shape model forming and the observations set were already explained in Section 2. This section briefly explains how to implement the estimation and tracking phase to be compliant with the Bayesian framework. To this end, the voting mechanism and the analysis of the voting space are introduced. Following, this section provides the hypotheses set selection and validation issues in order to estimate the object global position (tracking). More details can be found in our previous works [13, 14].

3.1. Voting mechanism

The voting mechanism implements (11)–(14). Having the model at time $t-1$, and the observations set at time t, the observations vote based on the model corners for the reference point in different positions of the voting space. The voting space is a two-dimensional space with Cartesian coordinates corresponding with the image plane with all the positions values set to zero, but it increases for every vote to that position or to a close position. Increasing the values in the surrounding area of the voted position is done to compensate for the potential deformation of the relative coordinates of a given model corner. This compensatory action is called *regularization*. The voting procedure along with regularization is shown in Algorithm 1. Voting and regularization steps produce a 2D histogram in the voting space, in which each bin represents a position in the image plane, and the value of that bin presents the number of votes to that position.

In Algorithm 1, v_x, v_y are the x and y values of the voted position coordinate, and V_t is the voting space at time t. The pair (r_x, r_y) indicates a regularization process around every voted position, for example, $-2 \le r_x, r_y \le 2$ covers a 5×5 area around the voted position (having a rigidity radius $R_R = 2\sqrt{2}$ and considering that the voted positions are coordinates with integer values, a 5×5 area is covered). As mentioned before, this is to compensate, somehow, for the small changes from frame to frame in the relative coordinates of the corners with respect to the reference point.

The voting procedure introduced here corresponds with formulae (12) to (14). Having a look at (11)–(14), one

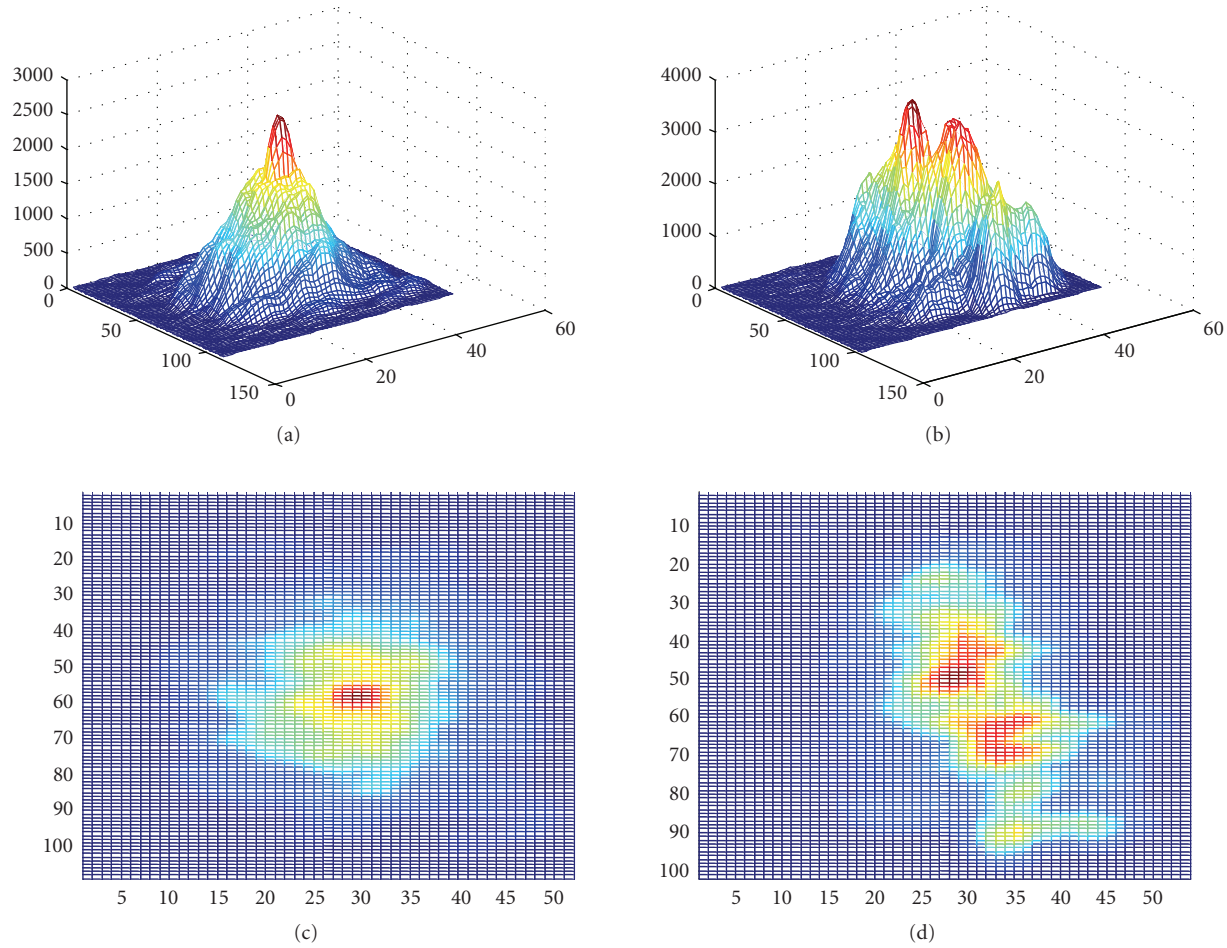


Figure 3: (a) The voting space histogram of the reference point related to the first frame (Frame #536) of the sequence that appears in Figure 11; (b) the voting space histogram related to Frame #570 of the same sequence, (c)–(a) from a top view, and (d)–(b) from a top view.

can see that the higher the number of votes, the higher the probability in formula (11). However, different kernels can be defined regarding the kernel used in (11). If it is defined as the Kronecker delta function (17), the only way a model corner and an observation corner will contribute to $V_t(\cdot)$ in (12) is when they have a null distance. If it is defined as a uniform kernel (18), a model corner and an observation corner will contribute to $V_t(\cdot)$ in (12), when their distance falls within a circle with a given radius centered on the Kronecker delta function. In this case, using a uniform kernel, different distances within a range predefined by fixing the radius R_R will generate equal contributions. The uniform kernel behaves the same as the regularization mask (r_x, r_y) in Algorithm 1, which has been used in this work.

3.2. Analyzing the voting space

Since most object corners vote for the reference point based on the relation between them (16), and vote for the wrong positions randomly, it is reasonable to consider the point with the maximum number of votes as the point that is most likely the reference point (11). This is equivalent to the point with the highest probability in (11). However, this is in normal cases where no distracter is available. In the absence of a distracter, the observations generate a unimodal voting space histogram (Figure 3(a)). In the presence of a distracter, a multimodal histogram is generated (Figure 3(b)). The reason is that the new corners related to the occluder, distracter, or clutter participate in voting. Having many irrelevant corners in the observations strongly affects the voting space leading to a multimodal histogram (Figure 3(b)) that, in turn, affects the position of the reference point. This is clearer in Figure 3. Figure 3 shows the voting space histograms related to the sequence shown in Figure 11. The tracked sequence in Figure 11 starts at Frame #535. Figure 3(a) shows the voting space histogram related to Frame #536, where no distracter is available. In this case, it is clear from the figure that the histogram is a unimodal one. The peak of this histogram contains the most number of votes, maximizing the probability in (11). Figure 3(c) shows the same histogram from a top view, where the red parts in the graph correspond to the area around the peak. Figure 3(b) is related to the same sequence, but Frame #570, where the object undergoes an occlusion. Figure 3(b) shows a multimodal histogram generated from the voting procedure. The multimodality is due to the occluder. Figure 3(d) shows

the same histogram from a top view, where the red parts are around the peaks. As seen in these two (Figures 3(c) and 3(d)), it is somehow difficult and risky to choose the mode with the maximum number of votes as the position that maximizes (11). This means that some points due to the distracter, clutter, or occluder have obtained high numbers of votes, making them highly probable to be chosen as the new estimated reference point. Therefore, the modes with high numbers of votes must be selected as a set of reference point hypotheses. Then, the hypotheses are evaluated to choose one of them as the new reference point. This is the first step toward a good enough estimation. This is explained in Sections 3.3 and 3.4.

3.3. Selecting the hypotheses set

There are different strategies for finding the modes in a multimodal histogram. For simplicity, a region dividing strategy has been chosen here. In this strategy, the voting space is divided into four regions as follows. First, it is divided into two regions vertically such that the summations of all vote numbers in the two regions have minimal differences from each other. For a $M \times N$ voting space, we have

$$X = \arg\min_{L} \left(\sum_{y=1}^{M} \sum_{x=1}^{L} V_t(x,y) - \sum_{y=1}^{M} \sum_{x=L+1}^{N} V_t(x,y) \right), \quad (48)$$

where $1 \le L \le N$, $V(x,y)$ is the number of the votes at point (x,y), and X is the x-coordinate of the vertical line that divides the voting space into two parts. In the same way, every subregion is divided again into two regions but horizontally (Figure 2). After dividing the voting space into four subregions, the absolute maximum for each subregion is found. Note that it is not necessary that all four subregions have the same number of votes. Actually, having the same number of votes is rather impossible due to the discrete nature of the voting space. Instead, Formula (48) tries to find subregions with minimum difference in the number of votes. In the case of no clutter, the regions are almost symmetric and the maxima relate mostly to the same mode, and hence, are very close (Figure 4(a)). In the case of clutter, the histogram is a multimodal one. This causes different modes to lay into different subregions. In other words, the maxima are attracted toward the clutter (Figure 4(b)). Now, finding the peak in each subregion, at least four absolute maxima $M\_\text{set} = \{\underline{X}^*_{p,t,h} \mid h = 1 \cdots 4\}$ are found. These maxima form a hypotheses set.

Figure 4, as mentioned above, shows the region dividing results obtained from applying this procedure on the histograms in Figure 3. Figure 4(a) is related to Figures 3(a) and 3(c). Figure 4(b) is related to Figures 3(b) and 3(d). The next step is to validate the maxima.

3.4. Maxima validation

The four maxima found at the previous step must be validated. To this end, the displacement vectors related to the previous reference point and current maxima are taken into account (Figure 5). Figure 5 shows six maxima: the reference

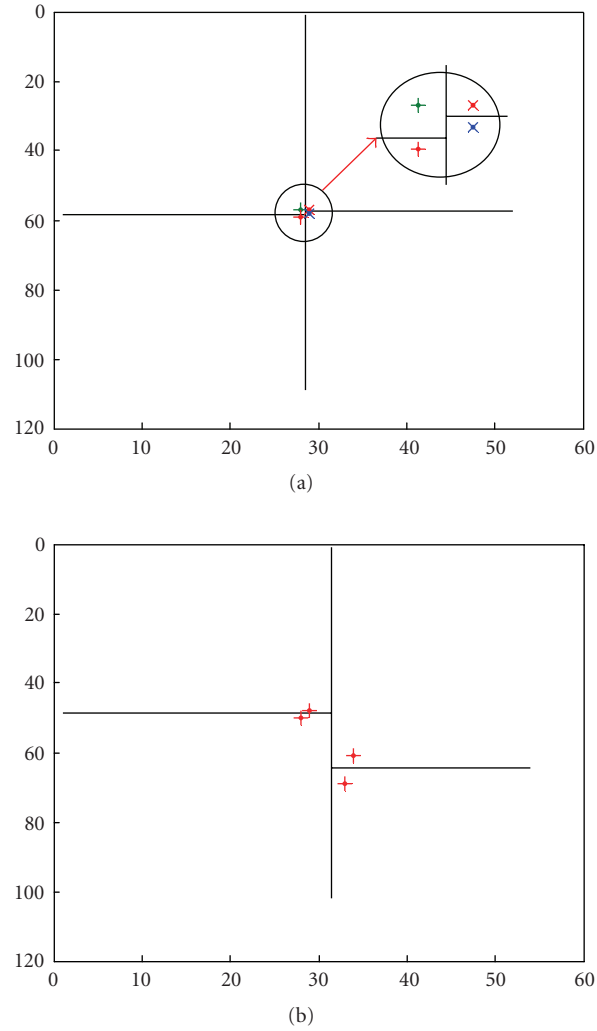


Figure 4: Dividing the voting space into four subregions. The absolute maximum in each region is found. (a) Region dividing along with the maxima related to Figure 3(a); (b) region dividing related to Figure 3(b).

points in the last frame and two frames ago, shown by $\underline{X}_{p,t-1}$ and $\underline{X}_{p,t-2}$, and four maxima in the current frame, shown by $\underline{X}^*_{p,t,h}$. Among the four maxima, the maximum that forms the closest displacement vector (like Formula (15)) to the one of the previous frame is chosen as the most probable reference point (Figure 5):

$$\begin{aligned} \underline{d}_{\text{ref},t-1}(\underline{X}_{p,t-1}) &= \underline{X}_{p,t-1} - \underline{X}_{p,t-2}, \\ \underline{d}_{\text{ref},t,h}(\underline{X}^*_{p,t,h}) &= \underline{X}^*_{p,t,h} - \underline{X}_{p,t-1}, \quad 1 \le h \le 4, \\ h^* &= \arg\min_{h} \text{dist}_e(\underline{d}_{\text{ref},t,h}(\underline{X}^*_{p,t,h}), \underline{d}_{\text{ref},t-1}(\underline{X}_{p,t-1})), \end{aligned} \quad (49)$$

where $\underline{d}_{ref,t-1}(\cdot)$ is the reference point displacement vector between two successive frames at times $t-1$ and $t-2$, $\underline{d}_{\text{ref},t,h}(\underline{X}^*_{p,t,h})$ is the displacement vector of hth maximum, dist_e is the Euclidean distance, and finally h^* is the index of the winner maximum that is chosen as the reference point of the current frame. Figure 5 shows this procedure: the blue

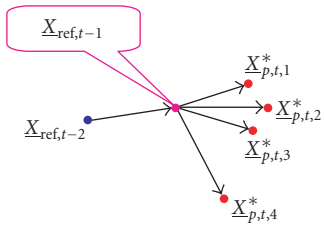


FIGURE 5: $\underline{X}_{\text{ref},t-2}$ and $\underline{X}_{\text{ref},t-1}$ are the reference points at times $t-2$ and $t-1$. The vector between these two points forms the displacement vector of the reference point. The red points are four maxima at time t, and hence, they form four displacement vectors. The closest one to the previous displacement vector is chosen.

point is the reference point at time $t-2$ and the pink one relates to the previous frame. The vector between them shows the last displacement vector of the reference frame. In the current frame, there are 4 maxima (the red points) forming four potential displacement vectors.

However, there may be two or more maxima, very close to each other. In such cases, there is no other alternative to validate them and to prefer one of them. Therefore, to shift the bounding box to the new optimal point (tracking), one trusts the aforementioned policy (formula (49), Figure 5).

The other alternative to select one maximum as the most probable reference point is to use the displacement vector (49) along with the number of the votes to each maximum, in the following way; the vote numbers of the maxima are normalized to the maximum number of votes:

$$V_h = \frac{V_t(\underline{X}^*_{p,t,h})}{\max\{V_t(\underline{X}^*_{p,t,h})\}} \quad 0 < V_h \le 1,\ h = 1 \cdots 4, \tag{50}$$

where $V_t(\underline{X}^*_{p,t,h})$ is the number of votes to the maximum $\underline{X}^*_{p,t,h}$, and V_h is the normalized number of votes for maximum $\underline{X}^*_{p,t,h}$.

Then, the distance between the previous displacement vector and each displacement vector, formed by each maximum, is normalized to the minimum distance among them, to compute a normalized proximity for each maximum:

$$\text{prox}_h = \frac{\text{dist}_e(\underline{d}_{\text{ref},t,h^*}(\underline{X}^*_{p,t,h^*}), \underline{d}_{\text{ref},t-1}(\underline{X}_{p,t-1}))}{\text{dist}_e(\underline{d}_{\text{ref},t,h}(\underline{X}^*_{p,t,h}), \underline{d}_{\text{ref},t-1}(\underline{X}_{p,t-1}))} \tag{51}$$
$$0 \le \text{prox}_h \le 1,\ i = 1 \cdots 4,$$

where prox_h is the normalized proximity value between the hth maximum displacement vector and the previous reference point displacement vector. Now, using both the normalized number of votes (50) and the normalized proximity (51) the winner maximum can be found as

$$W = \arg\max_h(\alpha\, V_h + \beta\, \text{prox}_h), \tag{52}$$

where α and β are the weights controlling the influence of the normalized terms. Regardless of the weights, the higher the number of votes for a given maximum, the higher the first term, and the closer the displacement vector of a given maximum to the previous reference point displacement vector, the higher the second term. The ideal case is when a given maximum has both factors simultaneously. Although (11) shows the probability based on the number of votes, it is possible to extend the Bayesian framework to also take into account the displacement vector. Actually, it can be shown that the current formula (52)is in compliance with the current Bayesian framework.

Having $\alpha = 0$ and $\beta = 1$ in (52), the only effective term is the proximity (that is a measure of the displacement vector, or the speed and direction of the motion) and formula (51) is obtained. Inversely, having $\alpha = 1$ and $\beta = 0$, the only effective term is the number of votes (that is a measure of the object shape) and Formula (11) is obtained. The weakness of using only formula (11) (having just the first term) was already analyzed in Figures 3 and 4. It is also shown in Figure 11 (left column). The weakness of using just the second term (formula (51)) is that it is only good for uniform object speeds. If the speed of the object increases, formula (51) may avoid the tracker to follow the object with the same speed rate.

To track the object, the maximum that is chosen as the winner using (52), is considered as the new object reference point, and the object bounding box is shifted to the new position to be centered on the winner. The next step is to update the model (learning).

4. MODEL UPDATING

After finding the target, the model must be updated so that the changes in the number of the corners and their relative coordinates can be applied to the model, making the model ready for use in the next frame. To this end, to use the four maxima found in the last step in order to handle occlusion events, first maxima are classified and after that the observations are classified. This makes different input classes of corners feed in the updating module. The updating module treats different classes in different ways. Details can be found in the following subsections.

4.1. Maxima corners classification based on their votes

After finding the new reference point (called the *winner maximum*), it is possible to classify maxima to distinguish the potential maxima introduced by the distracter. To this end, the Euclidean distances between all other maxima and the winner maximum are computed, and every maximum having a distance more than a threshold from the winner maximum is considered as a maximum belonging to other modes most likely generated by a distracter. Such a maximum is called a "*far maximum*" and forms the *far maxima set*. Other maxima close to the winner maximum form the "*pool of winners*" and are shown by W_set:

$$W\_\text{set} = \{\underline{X}^*_{p,t,h} \mid \text{dist}_e(\underline{X}^*_{p,t,h}, \underline{X}^*_{p,t,W}) \le \text{thr},\ 1 \le h \le 4\}, \tag{53}$$

where W is the winner maximum index according to (52). Other maxima form the set of far maxima:

$$F\_\text{set} = M\_\text{set} - W\_\text{set}. \tag{54}$$

Now, again taking a look at (53) and (54), one can understand that having an empty set of $F_$set is equivalent with Figure 4, a unimodal voting space. Conversely, having at least one member in this set is equivalent to a multimodal voting space. In the case of an empty set of $F_$set (no occlusions detected), the model is updated using all observations according to Section 4.3 However, in the case that an occlusion is detected (the set is not empty), all observations must be classified, based on their votes to the maxima, to distinguish between observations that belong to the distracter and other observations. To do this, the corners that have voted for all the considered maxima can be classified to three classes in order to remove the corners related to the occluder to be able to learn the model more precisely. These three classes are: good corners, mixed-good corners, and malicious corners. The classes are defined in the following way.

Good corners

The corners that have voted for at least one maximum in the "pool of winners" and have not voted for any maximum in the "far maxima" set. This class is shown by $g_$set:

$$g\_\text{set} = \bigcup_{i=1\cdots N_{W\_\text{set}}} V\_\text{set}_i - \bigcup_{j=1\cdots N_{F\_\text{set}}} V\_\text{set}_j, \tag{55}$$

where V_set_i is the set of all corners that have voted for the ith maximum (Figure 6), and N_{W_set} and N_{F_set} are the number of maxima in the "pool of winners" and "far maxima set," respectively.

Mixed-good corners

The corners that have voted for at least one maximum in the "pool of winners" and have also voted for at least one maximum in the "far maxima" set. This class is shown by mg$_$set:

$$\text{mg}\_\text{set} = \bigcap \left(\bigcup_{i=1\cdots N_{W\_\text{set}}} V\_\text{set}_i, \bigcup_{j=1\cdots N_{F\_\text{set}}} V\_\text{set}_j \right). \tag{56}$$

Malicious corners

The corners that have voted for at least one maximum in the far maxima set and have not voted for any maximum in the pool of winners:

$$m\_\text{set} = \bigcup_{j=1\cdots N_{F\_\text{set}}} V\_\text{set}_j - \bigcup_{i=1\cdots N_{W\_\text{set}}} V\_\text{set}_i. \tag{57}$$

Figure 6 shows the Venn diagram of the voter sets. The four dots are the four maxima, and each circle represents the set of the corners voting for one maximum. For example, the blue circle represents all corners that have voted for the blue

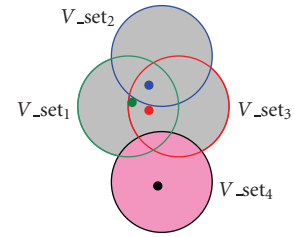


Figure 6: Venn diagram of four maxima and their voters: blue, green, and red are close maxima, and hence, members of the "*pool of winners*". The black represents a far maximum. Each color circle represents a set of voters for a maximum with the same color.

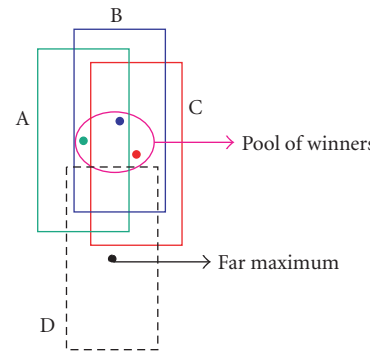


Figure 7: The bounding boxes are centered at the maxima. The union of the corners inside the bounding boxes A, B, and C are taken into consideration.

maximum. In the same figure, the three close maxima (pool of winners) and a far maximum (the black one) are shown.

Figure 8 shows the definitions of the three sets in (55) to (57). The pink oval labeled "G" can be considered as the good corners set, the black circle labeled as "MG" can be considered as the mixed-good corners set, and finally the brown oval labeled "M" can be considered as the malicious corners set.

Figures 9(a) and 9(c) show the classification of corners voting for the maxima in Frame #570 of Figure 11. The second row of Figure 10 shows another view of the corners classification. The first column contains all observations. The other columns show the good corners in blue, the mixed-good corners in yellow, and the malicious corners in red.

4.2. Forming the corner set to update the model

Up to now the voters have been classified. Actually, the classification introduced in the previous subsection (Formulae (55) to (57)) considers just the corners that have voted for the maxima. Yet, one step remains before updating. As mentioned above (in Section 3.4), to track the object, the bounding box is shifted to the most probable reference point. This means that the size of the bounding box is fixed. Updating the model is slightly different. Having a pool of winners, it is possible to have several bounding boxes, each of which is centered at one maximum in the pool of winners.

This is done in order to take into consideration all corners in the current position of the target and is shown in Figure 7. The motivation is that each maximum in the pool of winners could be the reference point, and hence, in order to avoid the situation in which the reference point has been incorrectly selected, the algorithm takes into account the corners in all those bounding boxes. The maxima in Figure 7 correspond to the maxima in Figure 6. In Figure 7, the pink oval indicates the pool of winners, and the bounding boxes A, B, and C are centered on the maxima in the pool of winners. The fourth maximum that is shown in black color is a far maximum. The dashed line shows a possible bounding box centered at the far maximum (D). To update the model, all the corners (observations) inside the bounding boxes A, B, and C (but not D) are considered. Actually, a union is taken on the sets of corners inside these three bounding boxes:

$$\text{Corners\_set} = \bigcup_{i=\text{A,B,C}} C\_\text{set}_i, \tag{58}$$

where C_set_i is the corners set inside the ith bounding box.

Considering the set "Corners_set", containing the corners around the target position and the three classes introduced before, it may happen that all or some corners of each class are present in the "Corners_set". Therefore, the number of corners in each class may be reduced and the classes are updated: the class of good corners is considered as those corners of the good class that are present in the "Corners_set". Note that the preliminary class of good corners and the updated class of good corners are shown by the yellow color in Figures 9(a) and 9(b), respectively, (equivalent to the black color in Figures 9(c) and 9(d), respectively). In the same way, the classes of mixed-good corners and malicious corners are considered the same as those corners of the mixed-good class and malicious class that are present in the "Corners_set". Other corners inside the "Corners_set" are considered "neutral corners". The blue color in Figure 9 shows the mixed-good class, where in Figures 9(a) and 9(c) this class has not been updated yet. These four classes, the three updated classes along with the neutral class, are shown in Figure 8. As it can be seen, the good corners are voting for the pool of winners, the mixed-good corners are voting for both the pool of winners and the far maxima set, the malicious corners are voting for the far maxima set, and finally the neutral set. These four classes are used to update the model.

4.3. Learning the model when occlusion is not detected

To update the model, the relative coordinates of all corners inside the "Corners_set" are calculated $\{\underline{DX}_t^k \mid \underline{DX}_t^k = (dx_t^k, dy_t^k)\}_{k=1\cdots K}$ with respect to the new reference point (the winner maximum) $(\underline{X}_{\text{ref},t}^* = \underline{X}_{p,t,W}^* = (x_{\text{ref},t}^*, y_{\text{ref},t}^*))$:

$$\begin{aligned} dx_t^k &= x_t^k - x_{\text{ref},t}^*, \\ dy_t^k &= y_t^k - y_{\text{ref},t}^*. \end{aligned} \tag{59}$$

Then, the relative coordinates of every model corner are compared to the ones of all these corners using the Euclidean distance (16), and if the minimum distance is less than a threshold ($R_R = 2\sqrt{2}$), the corner having this minimum distance is considered to be the same model corner (this is somehow similar to weighting the positions based on their distances using a Gaussian kernel (19)):

$$\begin{aligned} &\forall\ m = 1 \cdots M \text{ do}, \\ &\quad \text{dist}_{m,k}(t) = \sqrt{\left(dx_{t-1}^m - dx_t^k\right)^2 + \left(dy_{t-1}^m - dy_t^k\right)^2}, \\ &\qquad\qquad\qquad\qquad 1 \le k \le K, \\ &\quad \text{if } \min(\text{dist}_{m,k}(t)) < \|(r_x, r_y)\|, \\ &\quad k^* = \arg\min_{1\le k\le K} \{\text{dist}_{m,k}(t)\}, \\ &\quad \text{else, } \underline{X}_{s,t-1}^m \text{ does not have any association,} \end{aligned} \tag{60}$$

where arg means the argument (here: the corner index) among K corners that minimizes the distance. The variable k^* is the index of the associated corner candidate with the mth corner in the model. The symbol $\|\cdot\|$ is the norm function. The threshold $\|(r_x, r_y)\|$ that has been used here represents approximately the same area as the regularization window, where r_x usually equals r_y. This is because if any given model corner is supposed to have a movement in any optional direction for maximum $\|(r_x, r_y)\|$ pixels, then the regularization window must have the same value to be able to cover it. Moreover, to associate this corner with itself in two successive frames, the maximum distance between them must be $\|(r_x, r_y)\|$. If the mth corner in the model has an association k^*, the corner m is updated using the information of the associated corner:

$$\begin{aligned} dx_t^m &= dx_t^{k^*}, \\ dy_t^m &= dy_t^{k^*}, \\ P_t^m &= P_{t-1}^m + 1. \end{aligned} \tag{61}$$

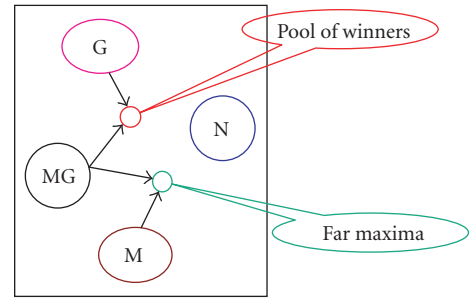


Figure 8: The four different classes of corners. "G": good corners that have only voted for the pool of winners, and they are present inside the "Corners_set". "MG": mixed-good corners that have voted for both the pool of winners and the far maxima set, and they are present inside the "Corners_set". "M": malicious corners that have only voted for the far maxima, and they are present inside the "Corners_set". "N": neutral corners that have not voted for the pool of winners or the far maxima.

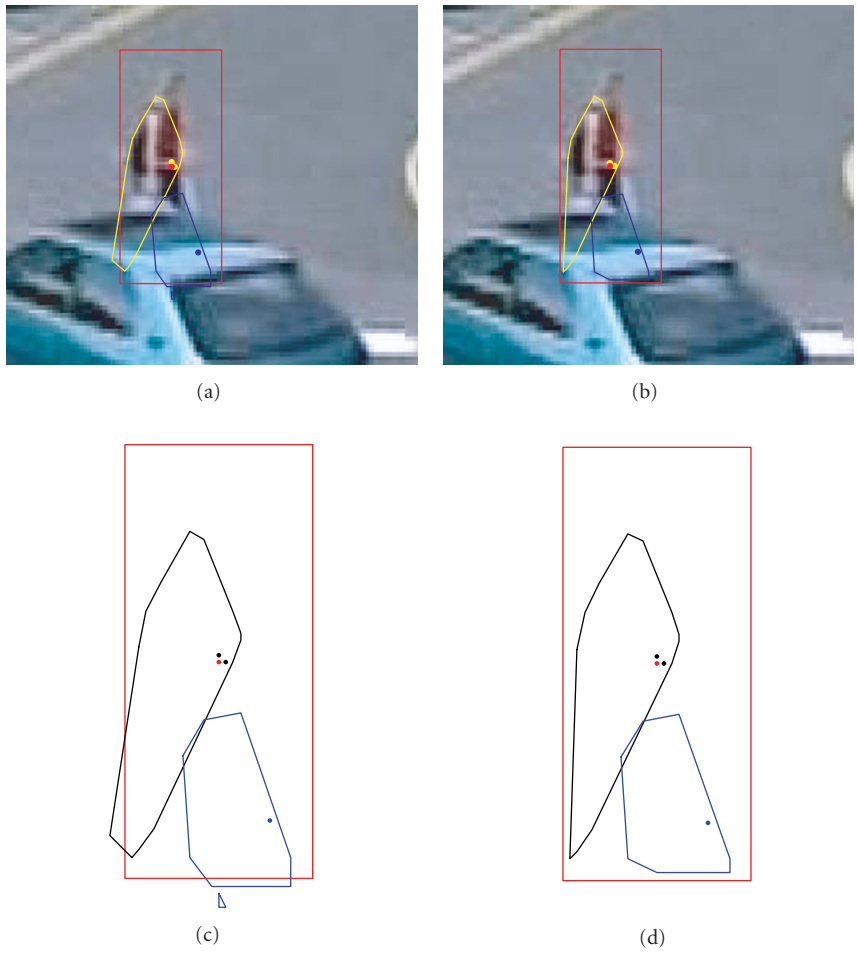


Figure 9: Frame #570 of the sequence appeared in Figures 11(a) and 11(b); the yellow convex: good corners, the blue convex: mixed-good corners, and the red rectangle: bounding box. (a) Classification after voting and (b) classification before updating; there is no malicious corner in this frame; the red point is the winner maximum; the red and yellow points are members of the pool of winners; the blue point is a far maximum; (c) and (d) the same as (a) and (b), respectively, changing the color of the good corners from yellow to black for a better contrast. Note that parts of the classes in (c) are out of the bounding box. They have been updated again (d) to contain the corners of the classes that are present in the Corners_set.

If a given corner m does not have any association, its persistence value will decrease by one:

$$\text{if } k^* = \{\}, \quad P_t^m = P_{t-1}^m - 1. \tag{62}$$

Once the persistency of a given corner goes below a threshold (e.g., $P_{\text{th}} = 1$), this corner is removed from the model. Setting the initial value of persistency to a relatively high value (e.g., $P_I = 5$) and the GHT inclusion threshold to a lower value (e.g., $P_{\text{th}} = 1$) gives the corner an opportunity equal to some number of frames (living time). This is because it may happen that a given corner, due to flickering or any reason, may disappear in the next frame but using this living time it is not removed from the model.

After examining all corners in the previous model, all other observations in the "Corners_set" that were not associated with the model corners, are added to the model, and their persistency values are set to the initial value of the persistency. To sum this up, the updating strategy performs the following four steps:

(i) updating all the information of the model corners with their associated observations;

(ii) decreasing the persistency values of the model corners that do not have any association;

(iii) removing model corners that have a persistency value less than the threshold;

(iv) adding the unassociated observations to the model with the initial persistency value.

4.4. Learning the model when occlusion is detected

To update the model during occlusion, as it was already mentioned in Section 4.2, malicious corners are removed to avoid the model being affected by the corners that do not vote

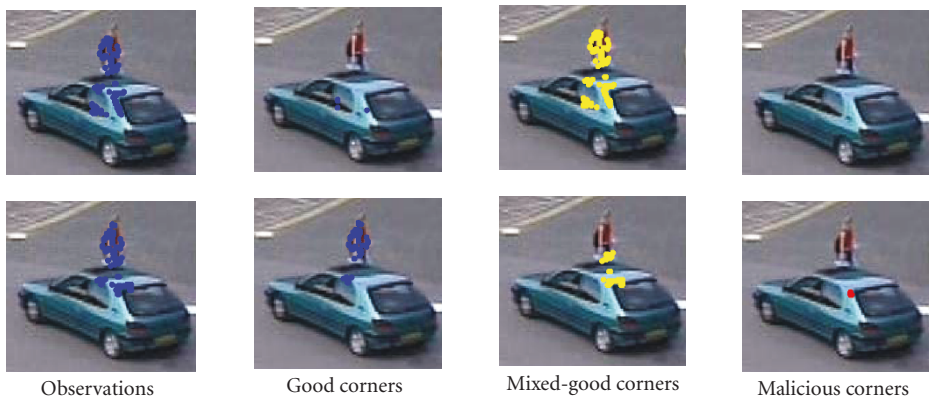


FIGURE 10: First step corners classification in Frame #570, columns from left to right: all observations: all the corners used to vote in the frame (blue), good corners (blue), mixed-good corners (yellow), and malicious corners (red). First row: updating the model during occlusion along with adding the unassociated corners to the model (corresponds to the second column of Figure 11). Second row: updating without adding the unassociated corners (corresponds to the third column of Figure 11). The first updating policy has not been able to classify precisely. There is no malicious corner using this policy (first row, last column).

for the reference point but generate a high number of votes in other positions of the voting space (this is automatically done when "Corners_set" is formed, see Section 4.2). Then, both the good corners class and the mixed-good corners class are considered to update the model corners in the same way as described in Section 4.3. The only difference between the current learning strategy (during occlusion) and the one described in Section 4.3 is that Formula (62) (decreasing the value of the persistency) is not applied here. The reason is that if a model corner does not have any association during occlusion, it may be due to the fact that the associated corner has been occluded, and hence decreasing the value of persistency may cause the model corner to be removed from the model. This can generate problems when the object passes the occlusion. Two different policies can be used regarding the unassociated corners: to add or not to add them to the model. Since some of the unassociated observations, or maybe all of them, belong to the occluder, the model is affected by the occluder model when they are added. The fourth class, neutral corners, can also be treated in two ways. First, they may be considered in the updating procedure discussed above. This is shown in Figure 11, Column 2 where the neutral corners have been considered in the learning phase, and the unassociated corners have been added to the model. Since many irrelevant corners infect the model and the current approach cannot distinguish them, the tracker fails in Frame #580, 20 frames after being involved in the occlusion. The second way to treat the neutral corners is simply to waive them because of the problem discussed above. In this policy, the algorithm trusts the persistent corners available in the model. The results of tracking and partially learning the model, waiving the neutral corners and not adding the unassociated corners, are shown in Figure 11, Column 3, where the tracker was able to track the object successfully. Figure 10 shows tracking using the two policies of the partial learning phase. Both of the rows show the corner classification after the voting mechanism (the first step of classification) in two different situations: in the first row, the tracker updates the model along with adding the unassociated corners to the model, while in the second row it uses learning, but without adding the unassociated corners to the model. As it can be seen in the figure, in the first row, classification is not precise. This is because the models are different. The reason that the models are different is related to the learning phase, as there are many irrelevant unwanted corners in the model. It means that the success of the method is tightly related to the classification as well as the learning phase: a bad classification leads to a bad learning, even if the learning phase has been properly chosen and vice versa. In the figure, as the classification method is the same, all the difference relates to the learning policies. The first row of this figure relates to the second column in Figure 11 where the tracker fails to track the target. From Figure 10, the reason can be clear after 20 frames during occlusion (560–580), and having an unsuitable learning policy, the error accumulation in the model causes the failure in tracking.

The advantage of corner classification and partial learning is to avoid the corners belonging to the distracters of being added to the model, in other words, the model is partially learned using the parts that are not occluded.

Again, similar to other parts of the algorithm, it can be shown that the learning strategy during occlusion is compliant with the Bayesian framework. Although some observations are discarded (malicious corners), the compliancy to the Bayesian framework is achieved through:

(i) adaptive shape noise (e.g., occluder, clutter, and distracter) model estimation by considering distribution of position observation model (20);

(ii) filtering observations $\underline{Z}_t$ to produce a reduced observation set $\underline{Z}'_t$;

(iii) substitute $\underline{Z}'_t$ in (20) to compute an alternative solution $\underline{X}'_{s,t}$.

Figure 11: Tracking results in three different modes. First column: tracking, occlusion detection, and stop learning during occlusion. Second column: tracking using partial learning along with the addition of unassociated corners to the model. Third column: tracking using partial learning, but without the addition of unassociated corners to the model.

The above-mentioned procedure simply says that discarded observations are noise. After discarding them (filtering done using classification), a new observation set is formed. This new noise-free observation set using (20) gives a different probability and a different model. However, it can be shown that since the new observations set is more precise, the resultant model will also be more precise.

5. CONTINUOUS LEARNING STRATEGY: DISCUSSION

In normal cases (no occlusion, distracter, or clutter), the updating strategy discussed above leaves the persistent corners of the object in the model and adds the newly found corners to the model. The advantage of this policy is that the model is adaptively learned by considering the new corners and updating the persistent ones. Since the method is able to detect occlusion and to change the learning strategy during the detected occlusion, it is able to partially learn the model using the visible parts of the object and adapting this part to the changes. Removing the occlusion detection module from the algorithm, it is clear that the method fails, since having many irrelevant corners in the model strongly affects the voting space leading to a multimodal histogram (Figure 3(b)) that affects, in turn, the position of the reference point. After detecting the occlusion, different decisions may be made. The first decision totally stops the model learning when the voting space histogram shows some unwanted objects in the scene close to the target and starts the model learning phase again when the target passes the clutter. The problem of this strategy, when to stop and when to start learning again, rises when the clutter (e.g., occluder) lasts for a long time. In this case, then model that has not been updated for a long time can not handle object tracking after passing the clutter, because the object has undergone some changes during the clutter. It may even fail during the clutter. The first column of Figure 11 shows such a policy. In this sequence, starting from frame Number 535 (reference frame), the object has been tracked until an occluder (the car) arrives close to the object (at frame Number 560). After tracking other 20 frames of the object under occlusion without updating the model, the tracker fails at frame Number 580.

The second strategy is to update the model partially at the areas that are not affected by the clutter (e.g., the parts of the object that are not occluded or the parts of the object that are far from the occluder). To this end, the corners voting for the maxima can be classified, and a mechanism can be applied to them to decide which corners are suitable for updating.

6. EXPERIMENTAL RESULTS

This approach has been tested with ten different sequences containing occlusions where a pedestrian is occluded either by a car or another pedestrian. Nine of the sequences have been chosen from pets2001 and pets2006. Since the Frames #535–610 of pets2001 (Figure 11) have been used in the paper to clarify different stages of the algorithm, the other analyses of the method, both qualitative and quantitative, are shown using the same sequence.

Figure 11 shows a sequence from pets2001. In the sequence, the girl starts walking toward the right, while a car appears from the right side of the scene. The occlusion lasts for almost 25 frames (Frames #560–585), and half of the body of the girl is occluded in the scene. The result for this sequence has been provided in three modes: (a) occlusion detection and stop learning, (b) partial learning during occlusion along with the addition of unassociated corners to the model, and (c) partial learning during occlusion, but without the addition of unassociated corners to the model. The left column shows the results of the tracking algorithm where the tracker detects occlusions using mode analysis of the voting space histogram and decides to stop learning during the occlusion. In this method, the tracker does not classify the corners, because it does not update the model at all during the occlusion. However, in safe situations, since all maxima are members of the pool of winners, the tracker does not need to use the classifier for updating. Despite this fact, to have the data related to the classifier, in order to be able to compare it with other policies (Figure 10), the tracker classifies the corners based on the maxima and keeps the data but does not use them. As it is shown in the first column of Figure 11, the tracker fails to track the object at frame 580. The reason can be that after being occluded for almost 20 frames (from Frame #560–580), not only has the model not been updated for a long time, but half of the object has also been occluded. These cause the observations, which not all of them belong to the object, to vote based on an old model, and hence a smaller number of them may vote for a proper position. In Figure 11, the red box is the bounding box, the blue box shows the area in which some corners are added to the model, and the yellow one shows the area containing the associated corners (the model corners that have an association).

The second column shows the results when the tracker uses partial learning along with the addition of unassociated corners to the model. Again, the tracker fails at Frame #580. This time the reason is because the system is overwhelmed by the unwanted unassociated corners (see Frame #575). A large area of the blue box has been covered by the occluder, and hence many of the unassociated corners may belong to the occluder, but they enter the model. Having another look at Figure 10 may provide some clarification.

The third column shows the results when the tracker uses partial learning but does not add the unassociated corners to the model. This time the tracker succeeds. A large area of the yellow box in Frame #575 again has been covered by the occluder. This means that the corners that are updated with associated ones lay in this area. However, this does not mean that all of the corners in the box enter the model, since some of them may not have an association and hence they will not be added to the model. The yellow box has been graphed by surrounding the associated corners, and it may happen that other unassociated corners exist in this area. Moreover, incorrectly updating a low number of corners does not affect the entire voting and tracking mechanisms (e.g., Figure 10) unless their number is significant. Frame #580 gives a better visual result of classification where the yellow

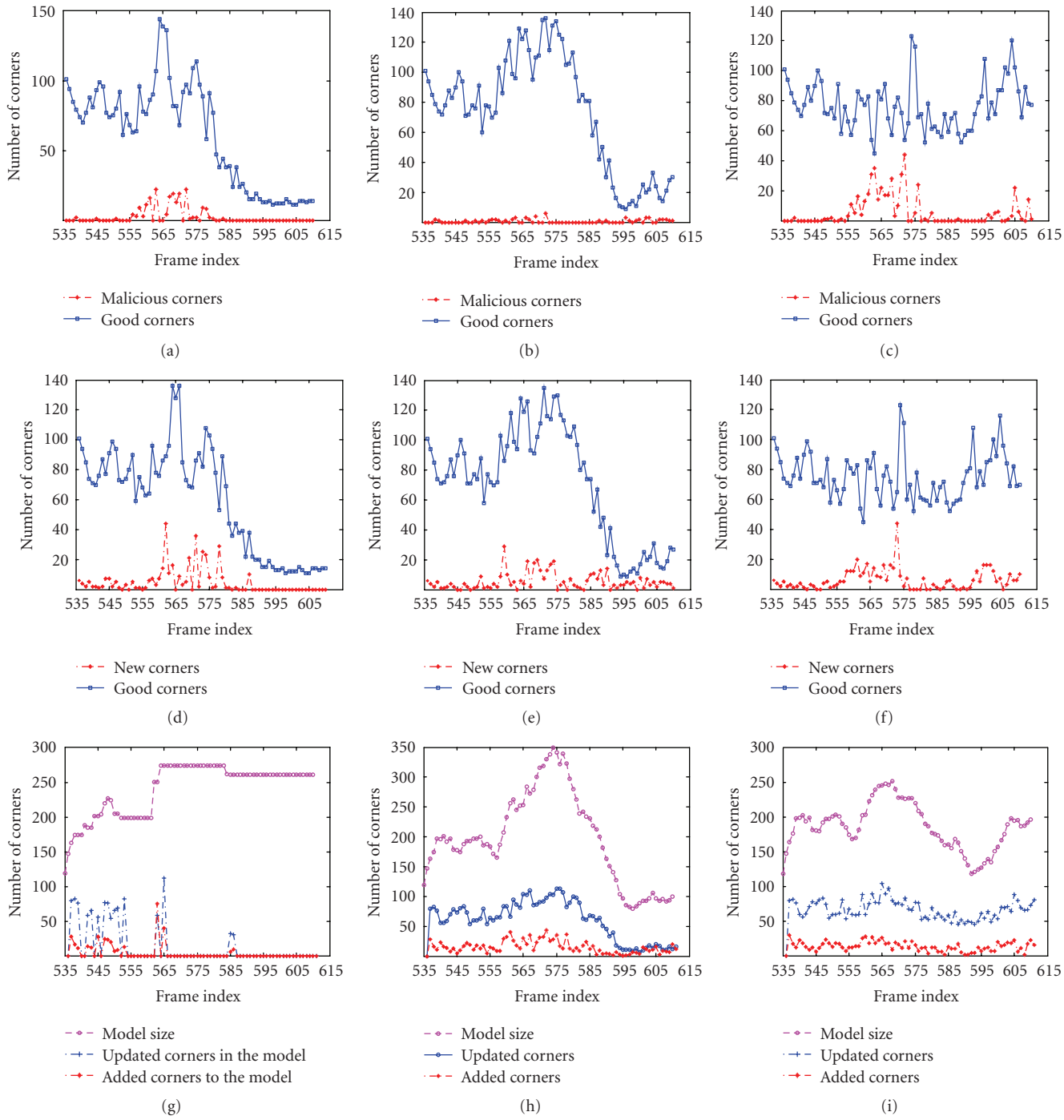


Figure 12: The data statistics regarding the object tracked in Figure 11. Each column corresponds to the same column in Figure 11. First row: number of corners in each class after the first step of classification. Second row: number of corners in each class after the second step of classification. Third row: the model size, the number of associated (updated) corners in the model, and the number of added corners to the model, in each frame.

box has covered only the upper part of the target and not the occluded parts. The data statistics related to this sequence for all three modes have been provided in Figures 12 and 13.

Figure 12 shows the data statistics related to Figure 11. The first, the second, and the third columns of Figure 12 refer to the first, second, and third columns in Figure 11, respectively. All graphs show the data for all image frames used in tracking; they started with Frame #535 and ended with Frame #610. The first row of Figure 12 shows the number of corners in each class after the first step of classification. The blue graph, labeled as good corners in the graph, shows the total number of the corners in two

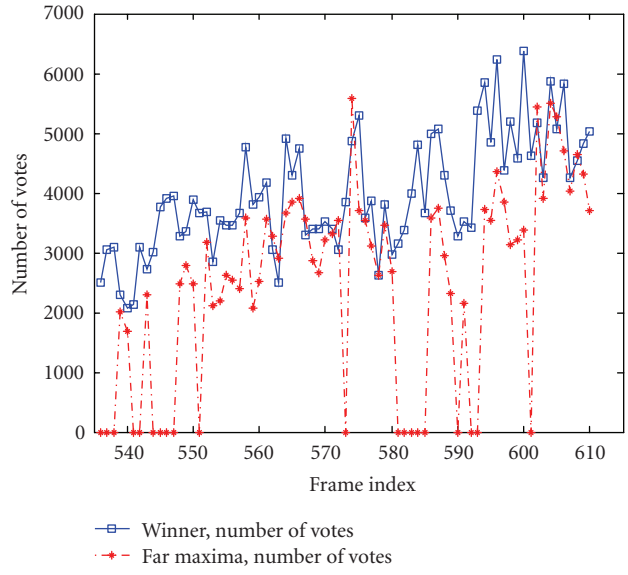


Figure 13: Number of votes for the winner maximum (blue) and for the far maxima (red).

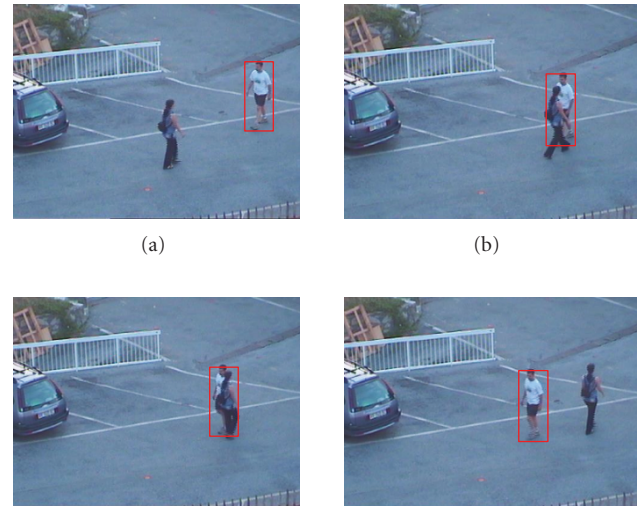


Figure 14: (a) Reference frame (Frame #200), (b) Frame #220, (c) Frame #224, and (d) Frame #235.

classes: good corners class plus mixed-good corners class. The motivation is that despite the fact that they are in two different classes, they are treated the same in the learning phase using the current approach. The red graph in the first row indicates the number of malicious corners. In the first two columns, the number of good corners decreases when the object is lost. It is because when the tracker loses the object in Figure 11, in the left column, it stops on the road and in the second column it follows the car while a large area of it has been covered by the road where there are not many corners. Having a look at the graph related to the malicious corners shows that in the first mode of tracking where the tracker stops learning there are a high number of malicious corners. In the second mode of tracking, where the unassociated corners are added to the model, the classification is not precise and there are not a high number of corners in the malicious set. This is seen in Figure 9. This also shows that an unsuitable learning method may cancel the effect of the classifier and leads to a failure in tracking. The third row indicates that not only has the number of the good corners not been decreased, but a higher number of malicious corners have been detected and classified. Watching the graph together with the results in the third column of Figure 11 indicates good news on the classifier and the learning policy. The increment in the number of malicious corners at the end of the graph is again due to the clutter; this time the clutter is caused by the background nearby the road and the grass. The higher the number of malicious corners in a frame, the more cluttered in the scene.

The second row shows the second step of classification. In this row the neutral corners are shown in the red color labeled as "new", and the total number of good and mixed-good corners in blue color labeled as "good corners". The number of good corners in this row (second step of classification) is less than or equal to the number of good corners in the first row (first step of classification). As mentioned before in Section 4.2 and Figure 9, the reason is that in the second step of classification only those corners are chosen that are inside the union of bounding boxes, and hence, some corners presented in the first step of the classification are discarded. The behavior of the good corners in this row is the same as the first row. As the malicious corners are removed in the second step of classification, there is no graph in this step related to them. The last graph of the second row shows that in the middle of occlusion event (about Frame #575), when the larger area of the bounding box is covered by the occluder (occlusion is heavier), the number of the new corners has also increased.

The third row shows the data related to the behavior of the GHT table (model) in different frames. The pink color shows the model size (the number of corners in the model). For the first method (first column), the pink graph shows that the model size has been fixed in two situations: when the tracker stops learning and hence no corner is added to or removed from the model, and when the tracker loses the object and is fixed on the road. This can be because the persistency values of the model corners in the last frames are still greater than the threshold. The second graph shows that as the object is occluded and then lost, and the model is updated successively, the model corners start to be removed from the model. The difference from the previous graph is that in the previous graph the model has been fixed for a long time and after that the corners have not had enough time to be removed from the model. The last graph has an average behavior following the behavior of the good corners (last column, rows 2 and 3), because it is updated using the good corners. The blue and red lines indicate the number of the corners that have been updated (they had associations) and have been added to the model, respectively. When the object

is lost (columns 1 and 2), the number of the updated corners and added corners has been decreased (this can be increased or behave normally depending on the direction and position of the bounding box after losing the object). Consider that in the third column, no corner is actually added to the model, but the graph has been plotted based on the number of the unassociated corners in each frame after learning the model, just to see the behavior of the tracker.

Figure 13 shows the number of votes for the winner maximum in blue for all frames of the sequence appearing in Figure 11 using the third method (learning without adding the unassociated corners to the model). This graph corresponds to the third column of Figures 11 and 12. The red line shows the number of votes to the far maxima. When the red one is zero, there have been no far maxima. As it can be seen, sometimes the number of votes for the far maxima is higher than the number of votes for the winner. This shows the added value of formula (52) by using both votes and distance to find the winner maximum.

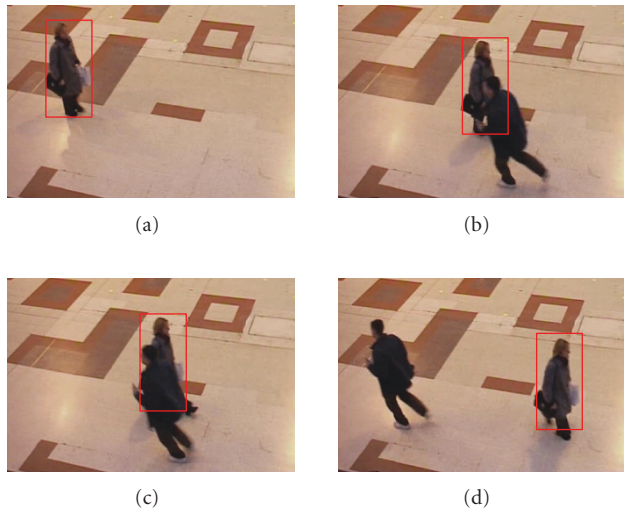
Figure 14 shows some results obtained by the proposed method. The sequence starts at frame #200. After about 18 frames an occlusion occurs. Consider two images (b) and (c) at Figure 14. Despite the fact that a large area of the object has been occluded, the method has successfully tracked it.

Figure 15 shows another sequence of pets2006. The boy is passing from the right side to the left side. Just after 16 frames an occlusion occurred but the algorithm was easily able to track the object. Figure 16 was also successfully tracked.

Figure 17 is somehow challenging. This is because this figure starts at Frame #102 and just after eight frames an occlusion starts. Although the occlusion lasts for just about nine frames, the important issue is that it started quite soon. If an occlusion starts quite soon after the model is formed, there may be a high possibility that the observations strongly affect the model, before giving the model enough time to improve. However, one can see that Figure 17 shows a success in tracking.

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

Figure 15: (a) Reference frame (Frame #1072), (b) Frame #1088, (c) Frame #1091, and (d) Frame #1098.



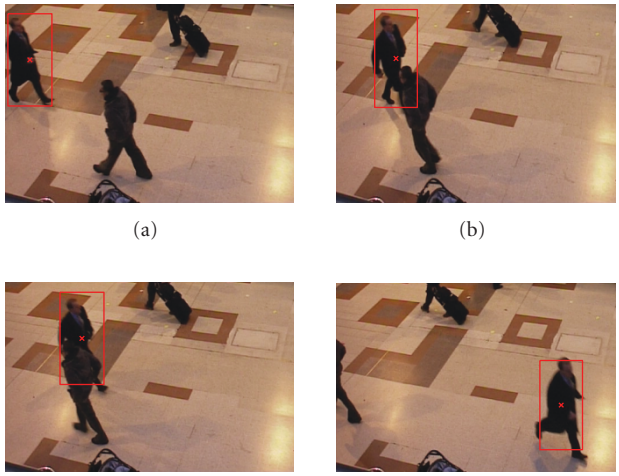
(a) (b) (c) (d)

Figure 16: (a) Reference frame (Frame #2246), (b) Frame #2269, (c) Frame #2271, and (d) Frame #2287.

7. CONCLUSION AND FUTURE WORKS

A Bayesian framework for a new continuous shape model learning method, able to track nonrigid objects in the presence of occlusions, has been introduced. In this method, the new position of the target was estimated by analyzing a multimodal histogram in the voting space. Since the model is updated at any frame, it could be affected by clutter and occlusion events. To avoid this problem, the histogram is divided into four regions and several maxima are chosen. After validating them, based on the number of their votes and their displacement vectors with respect to the previous object position, the most probable reference point is chosen as the position of the object. Then, a two-step classification policy is applied on the corners. The first step automatically classifies all the corners who have voted to the maxima into three classes: good, mixed-good, and malicious corners. Considering several bounding boxes centered at each maximum in the pool of winners and extracting the corners inside all of them, a further step of classification is performed using the output data of the first step of classification, and the corners are classified into four groups: good, mixed-good, malicious, and neutral. Using these classes, two different policies for partial learning were introduced. Both of them removed the malicious corners and continued learning using the good and mixed-good classes. The unassociated corners may or may not be added to the model. Therefore, the model could be learned continuously, even during occlusion, leading to a more precise estimation of shape and motion.

Yet, the learning mechanism can be improved, because two classes, good and mixed-good, have been treated in the same way. Moreover, all the corners in the neutral class were considered the same. A further improvement should consider the good corners for updating, and the mixed-good



(a) (b)

(c) (d)

Figure 17: (a) Reference frame (Frame #102), (b) Frame #110, (c) Frame #113, and (d) Frame #142.

corners as labeled elements for updating such that they can be identified in the next frame using the labels. In this way, other information (e.g., motion), about these corners can be extracted, leading to a better decision. In this case, each corner in the mixed-good class can be later classified based on its motion as a mixed-good corner tending to the object or to the occluder. The neutral corners can also be treated in such a way. Having motion information can improve the method greatly. The corners raised because of clutter in the background may have a zero motion, or a special predictable motion such as a small motion of a tree caused by wind in the scene. Moreover, the moving objects coming from other directions may be separated after monitoring the corners motions in two or three frames. For the objects coming in the same direction as the target, the velocity #may help.

ACKNOWLEDGMENTS

The authors wish to thank the anonymous reviewers for their helpful comments and suggestions, which helped to improve the paper.

REFERENCES

[1] F. Oberti, S. Calcagno, M. Zara, and C. S. Regazzoni, "Robust tracking of humans and vehicles in cluttered scenes with occlusions," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '02)*, vol. 3, pp. 629–632, Rochester, NY, USA, September 2002.

[2] S. M. Smith and J. M. Brady, "SUSAN—a new approach to low level image processing," *International Journal of Computer Vision*, vol. 23, no. 1, pp. 45–78, 1997.

[3] L. Marcenaro, L. Marchesotti, and C. S. Regazzoni, "Self-organizing shape description for tracking and classifying multiple interacting objects," *Image and Vision Computing*, vol. 24, no. 11, pp. 1179–1191, 2006.

[4] P. Gabriel, J.-B. Hayet, J. Piater, and J. Verly, "Object tracking using color interest points," in *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS '05)*, pp. 159–164, Como, Italy, September 2005.

[5] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the 4th Alvey Vision Conference*, pp. 147–151, Manchester, UK, August-September 1988.

[6] D. Wei and J. Piater, "Tracking by cluster analysis of feature points using a mixture particle filter," in *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS '05)*, pp. 165–170, Como, Italy, September 2005.

[7] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, pp. 674–679, Vancouver, BC, Canada, August 1981.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society B*, vol. 39, no. 1, pp. 1–38, 1977.

[9] Y. Rosenberg and M. Werman, "Representing local motion as a probability distribution matrix and object tracking," in *Proceedings of the DARPA Image Understanding Workshop*, pp. 153–158, New Orleans, La, USA, May 1997.

[10] L. Wiskott, "Segmentation from motion: combining Gabor- and Mallat-wavelets to overcome the aperture and correspondence problems," *Pattern Recognition*, vol. 32, no. 10, pp. 1751–1766, 1999.

[11] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.

[12] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1981.

[13] M. Asadi, A. Dore, A. Beoldo, and C. S. Regazzoni, "Tracking by using dynamic shape model learning in the presence of occlusion," in *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS '07)*, pp. 230–235, London, UK, September 2007.

[14] M. Asadi, A. Beoldo, and C. S. Regazzoni, "A nonlinear-shift approach to object tracking based on shape information," in *Proceedings of the 14th International Conference on Image Analysis and Processing (ICIAP '07)*, pp. 311–316, Modena, Italy, September 2007.