# Object Tracking using Color Interest Points

P. Gabriel, J.-B. Hayet, J. Piater, and J. Verly
Department of Electrical Engineering and Computer Science,
University of Liege, BELGIUM

## Abstract

*This paper presents a new approach for tracking objects in complex situations such as people in a crowd or players on a soccer field. Each object in the image is represented by several interest points (IPs). These IPs are obtained using a color version of the Harris IP detector. Each IP is characterized by the local appearance (chromatic first-order local jet) of the object and by geometric parameters. We track objects by matching IPs from image to image based on the Mahalanobis distance. The approach is robust to occlusion. Performance is illustrated by some examples.*

## 1. Introduction

We are interested in the tracking of multiple objects in color video sequences of complicated scenes, i.e., where objects can have different sizes, be rigid (e.g., cars) or non-rigid (e.g., people) or occlude each other. Most current tracking systems do not deal well with such different conditions. Most are either based on region extraction using a background model [3, 13] or on contour extraction using particle filtering [5, 7].

In our approach, an object is characterized by a set of interest points (IPs) obtained with a color Harris detector [9]. Each IP is characterized by its local appearance (a vector of local characteristics). The use of a set of IPs allows us to track an object through partial occlusion as long as one or more points remain visible. In addition, to increase robustness, we exploit potential geometric relationships between the IPs.

IPs with local descriptors have been used successfully for point matching in stereo reconstruction [10], image indexing [8], and object recognition [11]. However, to the best of our knowledge, this is the first attempt at tracking objects using color IPs characterized both by local descriptors and by a geometric model. Recently, Gouet and Lameyre [2] presented a tracker that uses IPs and snakes, but in grayscale images, without geometric model, and only for a single object in the scene.

This paper is organized as follows: Section 2 describes the color version of the Harris IP detector. In Section 3, we present the proposed tracking system, which combines an appearance model and a geometric model of each object. In Section 4, we show that our approach can track objects through occlusions. Finally, in Section 5, we conclude and propose some future research directions.

## 2 Color Harris detector

An IP is a point in an image where significant changes occur. Example IPs are corners, junctions, black dots on white background, and locations with significant texture changes. Several IP detectors have been developed over the last two decades. Schmid and Mohr [12] compare the performance of several of them.

The most popular IP detector is the Harris detector [4]. While this detector only applies to grayscale images, Montesinos *et al.* [9] generalized it to color images. The IPs produced by their detector are defined as the positive local extrema of the intermediate grayscale image

$$R(x,y) = \det(M(x,y)) - k(\mathrm{trace}(M(x,y)))^2, \quad (1)$$

where $k$ is typically set to 0.04 (as suggested by Harris and Stephens [4]) and $M(x,y)$ is the $2 \times 2$ matrix

$$M(x,y) = \begin{pmatrix} M_{11}(x,y) & M_{12}(x,y) \\ M_{12}(x,y) & M_{22}(x,y) \end{pmatrix}$$

constructed at each point of the image in terms of the three intermediate grayscale images $M_{11}(x,y)$, $M_{12}(x,y)$, and $M_{22}(x,y)$ defined as

$$\begin{aligned} M_{11}(x,y) &= G_{\sigma_i} \otimes (r_x^2 + g_x^2 + b_x^2) \\ M_{12}(x,y) &= G_{\sigma_i} \otimes (r_x r_y + g_x g_y + b_x b_y) \\ M_{22}(x,y) &= G_{\sigma_i} \otimes (r_y^2 + g_y^2 + b_y^2), \end{aligned}$$

where $G_{\sigma_i}$ is an isotropic 2D Gaussian with variance $\sigma_i^2$, $\otimes$ denotes the 2D convolution operation, and $c_x$ and $c_y$ represent the first-order Gaussian derivatives of the channels $c$ of the original image $I(x,y)$ with $c \in \{r,g,b\}$. These derivatives are implemented using a 1D Gaussian with variance $\sigma_d^2$.

According to the comparisons made by Gouet and Boujemaa [1], the above detector appears to be the most stable among the popular color IP detectors with regard to illumination changes, noise, rotation, and viewpoint changes.

0-7803-9385-6/05/$20.00 ©2005 IEEE

# 3 Tracking system

The goal of the proposed system is to track objects from frame to frame in a color video sequence. In the present form of our system, we assume for the first frame ($n = 0$) of the sequence that the objects to be tracked are well separated and that we manually define a rectangular region of interest (ROI) around each of them. Then, the system operates fully autonomously starting with the second frame ($n = 1$). In each frame $n$, we apply the color Harris detector to each ROI. For $n = 0$, we ignore the IPs detected outside the object. To track objects, we search for the correspondences between the IPs in the current frame and those in the previous frame based on a combination of an appearance model and a geometric model.

## 3.1 Combined appearance and geometric model

The appearance model of an object consists of the ensemble of IPs (in the corresponding ROI) characterized by a vector containing some local attributes. The most commonly used point descriptors are: the *local jet* which is a vector of coefficients characterizing the local graylevel surface by a Taylor expansion [6], the *differential invariants* which combine the elements of the local jet to achieve invariance to image rotation [11], and the *Scale Invariant Feature Transform* (SIFT) which represents a neighborhood by a large feature vector invariant to rotation and scaling, as well as robust to small changes in translation, illumination, and affine or projection transformations [8].

Since our application involves raw video sequences, we do not generally have to deal with significant changes from frame to frame. Thus, we do not have a need for differential invariants or SIFT. Therefore, the local jet is particularly adequate for our purpose. In fact, Gouet and Boujemaa [1] showed that, in the case of video sequences, point matching with the local jet gives better results than with differential invariants. Since we deal with color images, we use the local jet for each color channel. Since high-order derivatives are very sensitive to image noise, we use only the local jet up to order 1. It is made up of the channels $r$, $g$, and $b$ and the Gaussian derivatives $r_x$, $g_x$, $b_x$, $r_y$, $g_y$, and $b_y$.

In addition to the local jet, we use the local *cornerness* $R$ given by Eq. 1. Our tests indicate that $R$ helps in tracking IPs from frame to frame.

The drawback of using only an appearance model is that an IP could be matched with a look-alike even though these points are located far away from each other. E.g., for a soccer player, the IPs located above the right and left socks are very similar in appearance and could be confused.

Thus, to track an object robustly, we augment its appearance model with a geometric model. This model describes the coordinates $(x_c, y_c)$ of the IPs with respect to the object center, defined as the center of gravity of the IPs characterizing the object (Fig. 1).

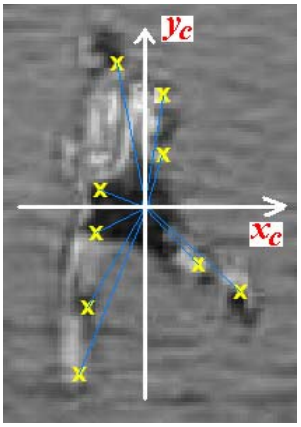


Figure 1: An object is characterized by a set of IPs, each characterized by a set of local-appearance attributes, and by its location relative to the object center, i.e., the center of gravity of the IPs.

Below, we denote each IP in any particular ROI in frame $n$, by $j$ with $j \in \{1, \ldots, P(n)\}$. The key element in our approach to tracking is the combined appearance and geometric feature vector defined for each $j$, in frame $n$, by

$$V_j(n) = (r, g, b, r_x, g_x, b_x, r_y, g_y, b_y, R, x_c, y_c)\,, \quad (2)$$

where all vector elements were introduced above. An object $O$ in frame $n$ is thus modelled by

$$V^O(n) = \{V_j(n) \mid j = 1, \ldots, P(n)\}. \quad (3)$$

## 3.2 Tracking algorithm

Consider a particular ROI (surrounding an object of interest) being tracked from frame to frame, say from $n-1$ to $n$. The basic problem is to find, for each IP $i \in \{1, \ldots, P(n-1)\}$, the best matching IP $j \in \{1, \ldots, P(n)\}$. To find $j$, we compare all $V_j(n)$'s to $V_i(n-1)$, as illustrated in Fig. 2.

A natural choice for the goodness of match is the Mahalanobis distance between $V_i(n-1)$ and $V_j(n)$,

$$d_M^2(i,j) = (V_i(n-1) - V_j(n))^T C^{-1} (V_i(n-1) - V_j(n)), \quad (4)$$

where $C$ is the covariance matrix measured first in a training sequence and then updated for the correctly matched IPs during the tracking.

IP $j$ is declared to match IP $i$ if it minimizes $d_M(i,j)$, $1 \le j \le P(n)$, and if this minimum is inferior to a threshold $\theta$. Clearly, some IPs $i$ and $j$ may remain unmatched.

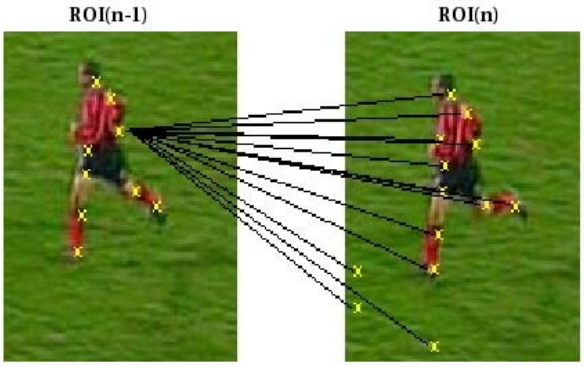


Figure 2: To track the IPs (belonging to a particular object) from frame $n-1$ to frame $n$, we consider each IP in frame $n-1$ and we attempt to find a match among all IPs in frame $n$.

In fact, we use a Kalman filter to predict the position of the center of a ROI in frame $n$. Ideally, this ROI would be centered over the corresponding object. However, this is rarely the case. Then, the new position of the object center is computed as explained in the next section. Finally, we search for the matching IPs.

In addition to the IPs in ROI(n) that have been successfully matched, we also include in $V^O(n)$ any IP in ROI(n) that falls within the bounding rectangle of those successfully matched IPs. This is to take into account the newly appearing IPs that presumably belong to the corresponding object.

Algorithm 1 below describes our approach to tracking a given object from frame to frame.

**Algorithm 1** TRACKING (for each object of interest)

1: Initialization : (a) Manually define ROI of object $O$ in frame $n = 0$. (b) Apply color Harris detector to ROI, manually select IPs belonging to $O$. (c) Compute center of $O$ as center of gravity of its IPs, and determine $V^O(0)$.
2: **for** frame $n = 1$ to $N$ **do**
3: Set $V^O(n) = \emptyset$.
4: Estimate position of ROI(n) using Kalman filter.
5: Apply color Harris detector and fill vectors $V_j(n)$ with appearance features.
6: Compute new position of object center using Algorithm 2 below.
7: Augment vectors $V_j(n)$ with geometric features.
8: **for** $i = 1$ to $P(n-1)$ **do**
9: **for** $j = 1$ to $P(n)$ **do**
10: Compute $d_M(i,j)$ using Eq. 4.
11: **end for**
12: **if** $\min(d_M(i,j)) < \theta$ **then**
13: Match $j$ to $i$ and add $V_j(n)$ to $V^O(n)$.
14: **end if**
15: **end for**
16: Include in $V^O(n)$ any other IPs that falls within bounding rectangle of successfully matched $j$'s.
17: **end for**

## 3.3 Centering algorithm

To compute the coordinates $(x_c, y_c)$ used in $V_j(n)$ in Eq. 2, we need the location of the object center (OC) in the ROI in frame $n$. When applying the color Harris detector to a given ROI, we are likely to find IPs in the background, such as those of ROI(n) in Fig. 2. In this case, the center of gravity of the IPs does not correspond to the OC. Moreover, as explained earlier, the estimation of the Kalman filter is not precise, i.e., the OC is not exactly in the middle of the corresponding ROI. The following algorithm will give us a more precise location of the OC.

If we assume that the IPs do not move much with respect to the OC from frame $n-1$ to frame $n$, we can determine the presumed OC in frame $n$ as follows. We successively consider the best $B$ matches $j \rightarrow i$ using only the appearance features, i.e., the matches obtained using the Mahalanobis distance with $V_i(n-1)$ and $V_j(n)$ limited to their first 10 features. For each of these matches, we compute the displacement vector $(\Delta x, \Delta y)$ of point $j$ with respect to point $i$. A rough estimate of the OC in frame $n$ is then obtained by adding this vector to the OC in frame $n-1$,

$$\begin{aligned} x_{oc}(n) &= x_{oc}(n-1) + \Delta x \\ y_{oc}(n) &= y_{oc}(n-1) + \Delta y. \end{aligned} \quad (5)$$

We then compute a more precise estimate of the OC in frame $n$ by using a trimmed mean over the $B$ above coordinates with $50\%$ outlier rejection. This is illustrated by point $\overline{C}$ in Fig. 3. Then, we suppress any value that is far from this mean. In the particular case of Fig. 3, we see that the rough estimate $C_3$ of the OC is far from $\overline{C}$. Therefore, we remove $C_3$ and any similar outliers, and we recompute the average with the remaining points yielding C as shown in Fig. 3.

Algorithm 2 describes the process of robustly computing the new object center. This process is very useful when an object is not in the middle of the corresponding ROI or when it is occluded.

This process is very useful when an object is not in the middle of the corresponding ROI or when it is occluded. Figures 4 and 5 illustrate the operation of the proposed tracker. The objects are correctly tracked even when the ROIs are not centered on the objects (as in frame 16 of Fig. 4) or when the object is occluded (as in frames 28, 36, and 46 of Fig. 5).

**Algorithm 2** CENTERING

1: Determine the best $B$ matches $j \rightarrow i$ based only on appearance features.
2: Compute the $B$ corresponding centers $C_b$ using Eq. 5.
3: Compute the mean value $\overline{C}$ of the $B$ centers using a trimmed mean.
4: **for** $b = 1$ to $B$ **do**
5: **if** $C_b$ is far from $\overline{C}$ **then**
6: Suppress the corresponding match $j \rightarrow i$.
7: **end if**
8: **end for**
9: Compute the mean value $C$ of the remaining centers.

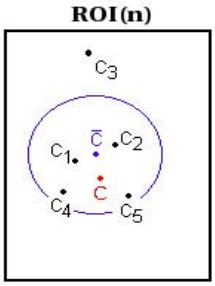


Figure 3: Illustration of the process for estimating the object center (OC) in frame $n$ using Algorithm 2. $C_3$ is an example of an outlier removed at Step 6 of Algorithm 2. Final object center is $C$.

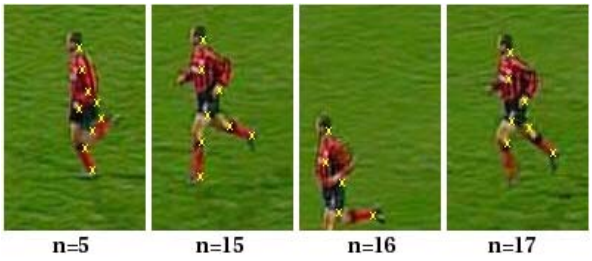


Figure 4: Example showing robustness of Algorithm 1 when the ROI is not centered over the tracked object. (ROI is full image shown.)

# 4 Mutual occlusions

In Section 3, we focused on tracking isolated objects. Tracking objects in crowded scenes necessarily leads to the problem of tracked objects occluding each other.

## 4.1 Disputed points

To deal with occlusion, it is useful to define some predicate indicating whether an occlusion is present or not. Here, we say that an occlusion occurs when two or more ROIs intersect.

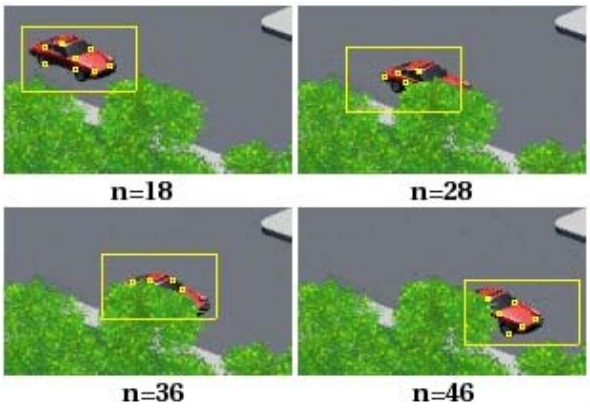


Figure 5: Example showing robustness of Algorithm 1 when the tracked object is occluded. (ROI is defined by the rectangle.)

When we detect an occlusion, we first apply Algorithm 1 to each ROI. Second, we collect the IPs that are in the intersection of the ROIs, as illustrated in Fig. 6 in the case of two ROIs. We refer to these points as the *disputed points*. Finally, we assign each disputed point to one of the objects involved in the occlusion based on the complete feature vector.

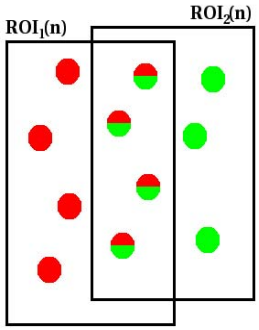


Figure 6: The points in $\mathrm{ROI}_1(n)$ (respectively, in $\mathrm{ROI}_2(n)$) are the IPs obtained using Algorithm 1. The points in the intersection of $\mathrm{ROI}_1(n)$ and $\mathrm{ROI}_2(n)$ are defined as the disputed points.

A disputed point $k$ will belong to an object $O$ if it has the smallest minimum for the distance $d_M^O(i,k)$, where $i = 1, \ldots, P(n-1)$ and $P(n-1)$ is the number of interest points that describe object $O$ in frame $n-1$. Note that we use all IPs deemed to belong to $O$ in the previous frame.

### 4.2 Recovering totally occluded objects

Let $\eta^O(n)$ be the number of matches $i \rightarrow j$ found for object $O$ in frame $n$. We say that $O$ is totally occluded in frame $n$ when $\eta^O(n)$ is less then 3. To recover $O$ in the next frame, we use a reference appearance model for $O$.

When an object is totally occluded, the position of its center is set to the position of the Kalman filter prediction of the corresponding ROI. Then, we apply the color Harris detector to the ROI, but characterize each IP with only the first 10 features of Eq. 2. Finally, we match this description of the object to a reference model $V_r^O(n)$.

The reference model $V_r^O(n)$ corresponds to the mean appearance of the object $O$ in frame $n$. It is updated from frame to frame whenever the object is correctly detected (i.e., with $\eta^O(n) \geq 3$). The updating uses a weighted average of all the IPs vectors, symbolically written as

$$V_r^O(n) = \alpha V_r^O(n-1) + (1-\alpha)\text{mean}(V^O(n)), \quad (6)$$

where $\alpha \in [0,1]$ and the $V_j(n)$'s of $V^O(n)$ contain only the appearance features.

Algorithm 3 describes our approach to tracking through occlusions.

**Algorithm 3** TRACKING THROUGH OCCLUSIONS

**for** each object $O$ **do**
  **if** $\eta^O(n-1) < 3$ **then**
    Mark $O$ as being totally occluded.
    Apply color Harris detector to ROI.
    Search for IPs that resemble $V_r^O(n-1)$.
  **end if**
**end for**
Apply Algorithm 1 to each ROI.
**if** two or more ROIs intersect **then**
  Find the $K$ disputed points.
  **for** $k = 1$ to $K$ **do**
    **for** each object $O$ in occlusion **do**
      Search for $\min_i(d_M^O(i,k))$,
      where $i = 1, \ldots, \eta^O(n-1)$.
    **end for**
    Assign $k$ to object $O$ having the smallest minimum.
  **end for**
**end if**
**for** each object $O$ **do**
  Compute $\eta^O(n)$.
  **if** $\eta^O(n) \geq 3$ **then**
    Update $V_r^O(n)$.
  **end if**
**end for**

## 5 Experimental results

As an experiment we tested the proposed algorithm in two different conditions: first with a short soccer sequence of 50 frames taken with a moving camera and second with an indoor scene of 200 frames taken with a fixed camera.

In both cases, we apply Algorithm 3 for two people crossing each other. Figures 8 and 9 show some results where an occlusion is involved. We can see that, in both cases, the two persons near the center of the frames are correctly tracked through the occlusion.

One of the limitations of our approach is that when an object is detected as lost, we assume that it will continue its trajectory in the same direction and at the same speed. However, this assumption is often violated in practice. Figure 7 illustrates a situation where the tracking falls.

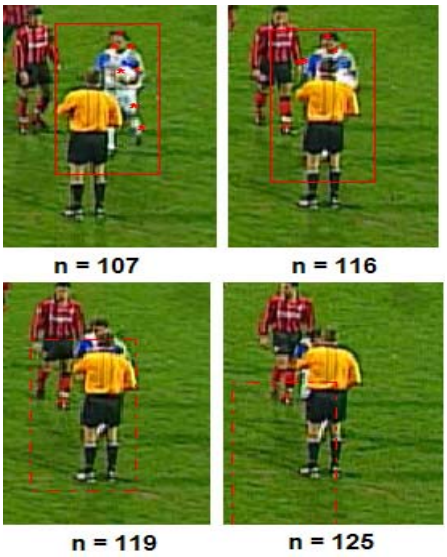


Figure 7: Example showing the result of tracking one object through an occlusion using Algorithm 3. The tracker loses the object.

## 6 Conclusions and future work

We have proposed a new approach for object tracking using color interest points (IPs). In this approach, an object is defined by a set of IPs detected with the color version of the Harris detector. Each point is first characterized by its local appearance, i.e., the color local jet up to order 1 plus the cornerness. Each point is further characterized by its position relative to the estimated center of the object. The use of a set of IPs allows us to track an object through partial occlusion as long as several points remain visible. The implemented system has been successfully tested on different

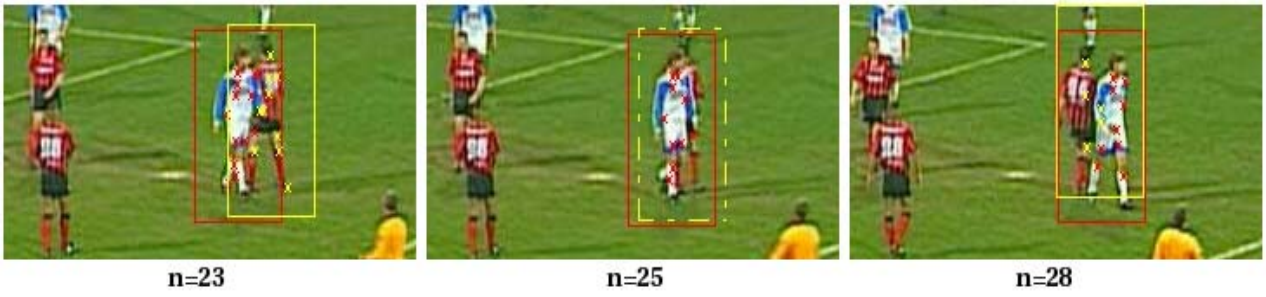


Figure 8: First example describing the result of tracking two objects through occlusion using Algorithm 3. The dashed line on an ROI boundary indicates that the corresponding object $O$ is totally occluded, which means that $O$ is described by less then 3 IPs. Both players near the center of the frames are correctly tracked through the occlusion.

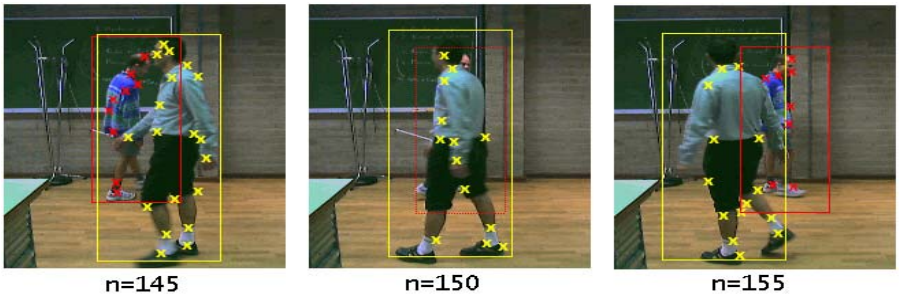


Figure 9: Second example describing the result of tracking two objects through occlusion using Algorithm 3.

scenes, thereby showing its generality.

In the current version of the system, we assume that the objects of interest in the *first frame* of the sequence are well separated and we manually place the ROIs around the objects as well as remove the IPs falling on the background. In future versions, we will, among other things, (a) fully automate the processing of first frame, (d) deal with objects entering or leaving the field of view, and (c) make the geometric model orientation independent.

# References


[1] V. Gouet and N. Boujemaa. About optimal use of color points of interest for content-based image retrieval. *Internal Report, INRIA Rocquencourt*, 2002.

[2] V. Gouet and B. Lameyre. SAP: A robust approach to track objects in video streams with snakes and points. *British Machine Vision Conference*, 2004.

[3] I. Haritaoglu, D. Harwood, and L. S. Davis. W4 - a real time system for detection and tracking people. *European Conference on Computer Vision*, pages 877–892, 1998.

[4] J. Harris and M. Stephens. A combined corner and edge detector. *4th ALVEY Vision Conference*, pages 147–151, 1988.

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

[6] J.J. Koenderink and A.J. Van Doorn. Representation of local geometry in the visual system. *Biological Cybernectics*, 55(6), 1987.

[7] P. Li, T. Zhang, and A.E.C. Pece. Visual contour tracking based on particle filters. *Image and Vision Computing*, 21(1):111–123, 2003.

[8] D. Lowe. Object recognition from local scale-invariant features. *International Conference on Computer Vision*, 1999.

[9] P. Montesinos, V. Gouet, and R. Deriche. Differential invariants for color images. *International Conference on Pattern Recognition*, 1998.

[10] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or 'how do i organize my holiday snaps?'. *European Conference on Computer Vision*, pages 414–431, 2002.

[11] C. Schmid and R. Mohr. Local greyvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, 1997.

[12] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.

[13] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. *Computer Vision and Pattern Recognition*, pages 246–252, 1999.