Robust Feature Descriptors for Efficient Vision-Based Tracking

Gerardo Carrera[1], Jesus Savage[1], and Walterio Mayol-Cuevas[2]

[1] Universidad Nacional Autonoma de Mexico(UNAM), Department of Electrical Engineering, Bio-Robotics Laboratory
Mexico City, Mexico
[2] University of Bristol, Computer Science
Bristol, U.K BS8 1UB

Abstract. This paper presents a robust implementation of an object tracker able to tolerate partial occlusions, rotation and scale for a variety of different objects. The objects are represented by collections of interest points which are described in a multi-resolution framework, giving a representation of those points at different scales. Inspired by [1], a stack of descriptors is built only the first time that the interest points are detected and extracted from the region of interest. This provides efficiency of representation and results in faster tracking due to the fact that it can be done off-line. An Unscented Kalman Filter (UKF) using a constant velocity model estimates the position and the scale of the object, with the uncertainty in the position and the scale obtained by the UKF, the search of the object can be constrained only in a specific region in both the image and in scale.

The use of this approach shows an improvement in real-time tracking and in the ability to recover from full occlusions.

Keywords: Object tracking, Harris detector, Speeded-Up Robust Features (SURF), Unscented Kalman Filter.

1 Introduction

Object tracking is at the core of many interesting computer vision systems. It is also challenging, due to the large space of object poses, perspective, illumination and scale changes and clutter. If an object tracker is capable to successfully solve these problems and at the same time keep the computational complexity of the tracker as low as possible, then it could facilitate several applications such as: security and surveillance, traffic management, augmented reality, mobile robotics, etc.

Recent advances in object detection (e.g. [2] [3]), demonstrate the capabilities of vision algorithms to deal with large occlusion and viewpoint changes. Usually, they relay on the detection of key or interesting points to be used as features, and on the building of descriptors around those points of interest. This robustness to scale and occlusion usually translates in the expensive computation needed by those algorithms.

L. Rueda, D. Mery, and J. Kittler (Eds.): CIARP 2007, LNCS 4756, pp. 251–260, 2007.

There are numerous approaches to detect interest points [4] [2] [3], most of them differ on the information that the points represent, this yields to a very important issue which is distinctiveness, which means how well the points can be matched in different images. This and the repeatability of the points detected between different images of the same scene under different changes in viewing conditions, are of major concern.

In [5] it was demonstrated that the Harris detector performs well compared to other keypoint detection algorithms in terms of repeatability, but it is well known that this detector is not scale invariant, so to overcome this deficiency, the multi-resolution framework proposed by [1] is used here, where for each point detected, a SIFT-like descriptor is created at several fixed scales. This is done only once and it is computed off-line which overcomes the computational cost of computing the scale space in each frame as done in say SIFT [2] or SURF [6].

To predict scale and object position and their associated uncertainties, an estimator is used. In this case, this is an unscented Kalman filter (UKF) although other estimators are equally applicable.

The goal of this paper is to develop a fast, accurate and efficient tracker that benefits from the repeatability of the computationally expensive detectors created for object recognition and from the well established methods of estimation.

The first part of the algorithm consists of representing the target object with low-level information using interest points extracted from the ROI (see Fig. 1). These interest points are the representation of the object, the use of these points has some advantages, they are locally extracted which gives to the object a robust level of invariance to occlusion as well as to noise and illumination changes. Because the object is often moving and changing its position, it may appear different in each frame so it is important to obtain invariance to some image transformations. This paper it is focused on rotation and scale invariance.

The remainder of this article is organised into 6 parts. Section 2 describes related work, section 3 gives a brief description of the algorithm. Section 4 shows the process of object representation which consists in the detection and the description of the interest points, and describes the object tracking framework using the UKF. Section 5 shows the experiments and the results obtained and finally section 6 provides the conclusion.

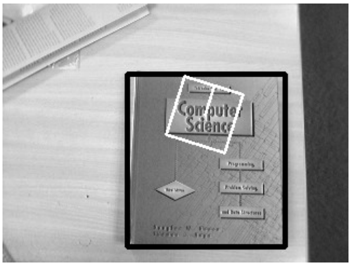

Fig. 1. Region of Interest (black rectangle) showing a single interest point (middle point inside the white rectangle)

2 Related Work

In a typical visual tracker, two components can be distinguished: 1) target or object representation and localisation, and 2) filtering and data association [7]. From the object representation point of view, amongst the wide variety of approaches adopted some employ a reduced amount of information extracted from the object, such as color [8] [9], intensity [10], interest points [11] or spatialized color histograms [12]. Some integrate different representations such as in [13]. There are also approaches that use a well described model of the object, this basis is useful when the goal is to track say solid models. These approaches are based mostly on the contour, edges or on a more detailed representation of an image curve using a parameterisation like B-splines [14]. This level of description can be complicated at best to achieve. There are however examples where this approach works well e.g. [15] [16].

In the second component of a visual tracker, the principal idea is to estimate the next state of the object, using a sequence of noisy measurements made on the system. To do this, an estimator or filter can be used. There are different filters used in tracking problems, under certain circumstances, it is assumed that an optimal solution is given by the Kalman Filter (KF) when the problem is linear, however, in a typical tracking problem there are different factors that make the problem highly not linear.

The Extended Kalman Filter (EKF) is probably the most widely used estimation algorithm for nonlinear systems, unfortunately it exhibits potential drawbacks and serious limitations. First, linearization is only reliable if the error propagation can be approximated by a linear function and can be applied only if the Jacobian matrix exists [17]. Second, the derivations of the Jacobian matrices can be complex, causing implementation difficulties. A most recent alternative is the Unscented Kalman Filter (UKF) [18], which handles the problems caused by linearization providing a mechanism for transforming the mean and covariance information and avoiding the calculus of Jacobian matrices. This estimation algorithm will be described later with more detail. Another more general class of filters are the particle filters which are based on Monte Carlo integration methods. In these filters the current state is represented by a set of randomly generated samples which are used to approximate the filtering distribution, [19] [20] [21]. An issue with particle filters is the need to evaluate and keep a relatively large number of particles and the complication of deciding which hypothesis to use to indicate the location of the object. This work uses the UKF as a good compromise but as mentioned before other estimators and filters can be incorporated.

3 Tracking Algorithm Overview

The first part of the algorithm consists in the object representation, its definition does not assume a fixed form, however, the region of the image to be tracked is delimitated by a rectangle defined by two opposite corners, this generates a

ROI where the interest points are going to be extracted. The interest points are extracted using the Harris detector and the points are not extracted in different scales. To solve this problem, the scale invariance is incorporated in the descriptor building a stack of descriptors at several scales, one for each interesting point, this idea has been proven to work effectively using SIFT descriptors for a visual SLAM system [1].

Once the interest points are extracted and matched against those detected in the first frame, the object center is calculated. It can be obtained in two different ways: the first one is by taking into account the relative coordinates of the points calculated in the first frame to the object center and the scale, the second one is calculating the homography between tracked and original template and getting its center. In the second phase of the algorithm, a UKF estimates frame by frame the center and the scale of the object. This is very important for the performance of this tracker because the interest points are only extracted from a region of the image constrained by the predicted scale covariance and the object center. The predicted scale covariance and the scale predicted are also used to find the region in the stack of descriptors where the system is going to look for possible matches, which makes the matching step computationally easier. It is expected that if the camera looses the object, the uncertainty in the scale and the position of the object will grow until the search region covers the whole image and the complete stack of descriptors in scale (see Fig. 2). If the object is "re-localised" before that moment, an efficient use of the known information would have taken place.

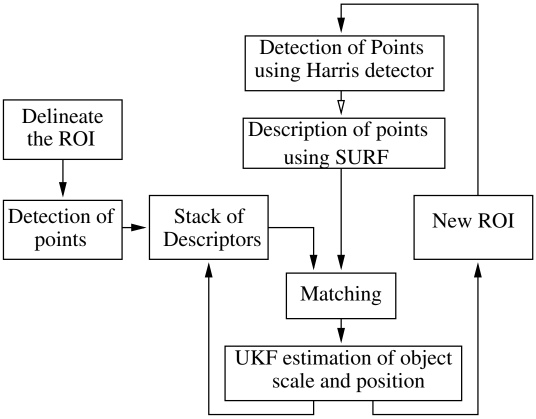
Detection of Points using Harris detector
Description of points using SURF
Delineate the ROI
Detection of points
Stack of Descriptors
New ROI
Matching
UKF estimation of object scale and position

Fig. 2. Block diagram of the object tracking system

One possible problem of this approach is having too many points representing the object, making the matching step more difficult. This is solved in the next stages of the algorithm during the object tracking procedure where the interest points descriptors are built only at a fixed scale and are matched against only those descriptors between a pre-defined range of scales. The interest points description is based in SURF (Speeded-Up Robust Features) [6], which has been proven to approximate or even outperform previously proposed schemes and the fact that it can be computed faster makes it more reliable for our system. The SURF descriptor will be described in the next section.

4 Object Representation

To extract the interest points, the Harris detector is used [4]. This detector is well known for detecting not only corners but also locations in the image where the signal changes two-dimensionally, this is achieved by using the autocorrelation function defined by (1).

$$c(x,y) = [\Delta x, \Delta y]\ M\ [\Delta x, \Delta y]^T \tag{1}$$

where Δx and Δy are shifts of small windows centered on (x,y). The matrix M denotes the intensity structure of the local neighbourhood, this 2×2 matrix is computed from image derivatives:

$$M = \begin{bmatrix} \sum_W \frac{\partial^2 I}{\partial x^2}(i,j) & \sum_W \frac{\partial^2 I}{\partial x \partial y}(i,j) \\ \sum_W \frac{\partial^2 I}{\partial x \partial y}(i,j) & \sum_W \frac{\partial^2 I}{\partial y^2}(i,j) \end{bmatrix} \tag{2}$$

where (i,j) are the index of the values in the window W over the image I. The location of the feature point is obtained by doing maximum suppression over a 3×3 region using the next function:

$$cornerness = det[M] - \alpha[trace(M)]^2 \tag{3}$$

After the interest points are extracted from the ROI, the description of the interest points is achieved using the fast descriptor coined SURF [6], which makes use of integral images [22] and Haar-wavelets responses. A point $p = (x,y)$ in an integral image $Integral(x,y)$ represents the sum of all pixels in the input image $I(x',y')$ of a rectangular region formed by the point p and the origin.

$$Integral(x,y) = \sum_{x'=0}^{x} \sum_{y'=0}^{y} I(x',y'). \tag{4}$$

The SURF algorithm makes use of the integral image to detect the interest points as well as to describe them.

After computing the integral image, the invariance to rotation is achieved by calculating the Haar-wavelets responses in x and y direction. Because of the use of the integral image, only six operations are needed to compute the response in x or y direction at any scale.

The responses are represented as vectors. To get the dominant orientation, first it is necessary to get the orientation in a sliding window covering an angle of $\pi/3$, by summing all the vectors that are within the window. The longest vector leads the dominant orientation of the feature point.

To compute the descriptor, a square region is defined centered around the feature point and oriented along the dominant orientation. This region is split into 4 × 4 square sub-regions. In each sub-region, regularly spaced sample points are taken and over these point, the Haar wavelets responses in x and y directions are calculated,

for simplicity they are called d_x and d_y.are also weighted with a Gaussian($\sigma = 3.3s$) centered at the feature point, this is done to achieve robustness towards geometric deformations and localisation errors.

The responses over each sub-region are summed obtaining a vector over each region. These vectors and the sum of the absolute value of the responses over each sub-region give us the total entry of the descriptor. So each sub-region will contribute to the descriptor with 4 values. The structure of the descriptor is then $D = (\sum d_{x_i}, \sum d_{y_i}, \sum |d_{x_i}|, \sum |d_{y_i}|, ...)$ where $i = 1, ..., 16$. The descriptor is turned into a unit vector to achieve invariance to contrast [6].

4.1 Multi-resolution Descriptors

Instead of using a scale-space representation to achieve scale invariance, a list of descriptors for each interest point is built at initialisation. Multiple descriptors are constructed at the first frame at different resolutions and they are saved in a stack list, this scheme is useful in two different ways: first, with this approach the scale invariance is achieved and second, an efficient use of computational resources is made. In the subsequent frames the descriptors are computed only in a fixed resolution so this list is used to seek to match those descriptors to those computed at a fixed resolution. In this article the terms resolution and scale are considered equivalent. Not only the size of the region where the descriptor is extracted is scale dependent but also the length between samples, which means that the number of samples is fixed so it is only increased or decreased the size of the window and the length of the sampling interval according to the resolution where the descriptor is going to be computed [1].

In the first frame as it is known the spatial position of the object, it is also known the object center is defined. To be able to calculate the center in subsequent frames, it can be done in two different ways: in the first one it is avoided the calculation of a transforming mapping. In the first frame it is saved for each interest point, the position (x, y) relative to the object center P_c. In the next frames it is used this measure and the scale of the object to get the object center, as it can be seen in the next equation:

$$P_c(x, y) = \frac{\sum_{i=0}^{N} (M_i(x, y)/Scale_i) + p_i(x, y)}{N} \tag{5}$$

where N is the number of points that matches, $M_i(x, y)$ is the measurement of the point relative to the center, $Scale_i$ is the scale of the point and $p_i(x, y)$ is the spatial position of each point.

The second way of doing it, is computing the homography. This is defined as an invertible mapping where a plane can be projected trough a point onto another plane [23]. In this work it is used an affine mapping for the homography calculation, which includes scales, rotations, translations, and shears. For a robust estimation, the RANSAC algorithm is used to generate a better homography. With this method the object center can be computed in each frame. The idea is that when it calculates a bad homography, the system can still use (5) to get the object center.

4.2 Filtering and Data Association

The tracking process is achieved by predicting the object center position and the scale in the next frame. The Unscented Transform(UT) is used to compute the first two statistical moments for the position and the scale, the means μ_p and μ_s, and variances σ_p^2 and σ_s^2. Using μ_s and σ_s^2 it can be searched in an interval defined by $I = \mu_s \pm 3\sigma_s^2$, in the stack of descriptors, where the matching scale should be found with high probability. With these statistical measures and the size of the ROI obtained in the first frame, the region of the image where the object is located is constrained by, $Width_{newROI} = \frac{Width_{ROI}}{\mu_s} + k\sigma_s^2$ and $Height_{newROI} = \frac{Height_{ROI}}{\mu_s} + k\sigma_s^2$, where k is a constant chosen experimentally (see Fig. 3).

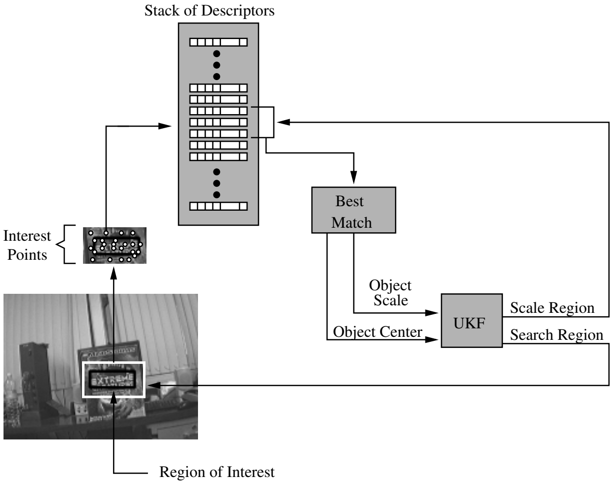

Fig. 3. Schematic view of the object tracking algorithm

Unscented Kalman Filter. The UKF is a variant of the Kalman Filter (KF) for non-linear systems that address the EKF deficiencies. It is based upon the Unscented Transformation (UT) which is a method for calculating the statistics of a random variable which undergoes a nonlinear transformation [18]. A set of sample points are chosen deterministically in order to compute the mean and covariance of the random variable, when these points are propagated through the non-linear system, then information can be extracted about the posterior mean and covariance with an accuracy up to the 2nd order in the Taylor series expansion. The basic idea then is propagate the mean an covariance information through nonlinear transformations. The UKF is an extension of the UT regarding the recursive nature of the KF. More details about the UKF can be found in [17].

5 Experiments and Results

The system was tested on a 2.2 GHz Pentium 4 PC. The implementation is on Linux in C++ using the openCV library. A firewire camera with FOV of 42° which feeds video at 30 fps at a resolution of 320x240 is used.

Figure 4.I shows a comparative graphic of the tracking system developed in this work vs the naive SURF algorithm, using a video sequence considering changes in scale, rotation, partial occlusions and the total lost of the object. The tracking system shows a fast recovery after temporary total lost of the object. In graphic a), it can be seen that the frame rate is much better through the entire sequence giving a mean of $\mu_{System} = 11.1\ fps$ compared with $\mu_{SURF} = 6.04\ fps$. In graphic b), the number of points detected and matched against the points detected in the first frame are shown.

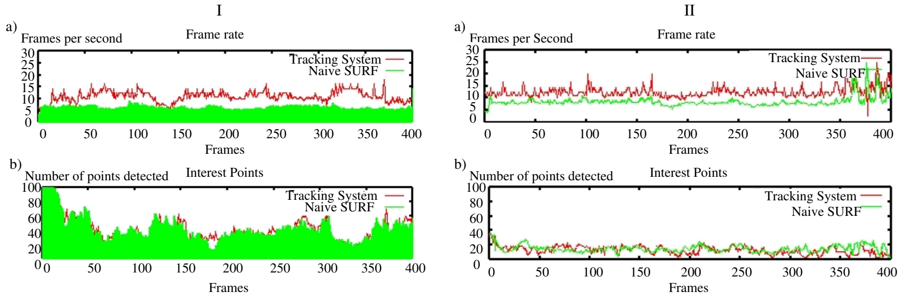

Fig. 4. Graphic showing the performance of the tracking system compared with SURF for two different objects: I) and II)

Figure. 4.II shows a sequence with a different object, where the number of points detected in the first frame is less for the two approaches, 40 for the proposed tracking system and 46 for SURF, compared with those detected from the object I. It can be seen also from this sequence that the frame rate is better in the whole video giving a mean of $\mu_{System} = 12.24\ fps$ compared with $\mu_{SURF} = 8.63\ fps$.

From the graphics b) in Fig. 4 , it can be seen that the amount of points correctly matched for the two objects are roughly the same. Figure 5 shows sample frames of the tracked object over sequence 1. Video sequences of the performance of the method can be seen at [24] and [25].

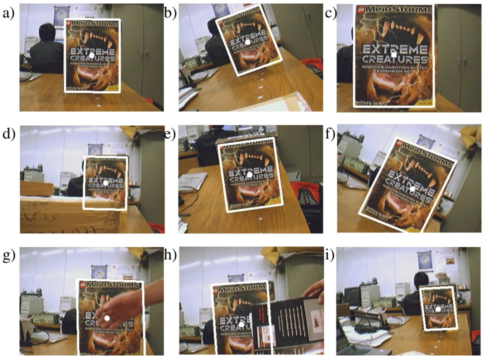

Fig. 5. Sample images of the test sequence using the proposed tracking system, each image represent the next frames: a)8, b)110, c)139, d)187, e)260, f)380, g)406, h)487, i)337

6 Conclusion

This paper presents a robust implementation of an object tracker using a vision system that takes in consideration partial occlusions, rotation and scale for a variety of different objects. The approach does not assume the form of the object and the results showed that it can track successfully and efficiently identified objects.

By utilising the proposed framework, an efficient implementation of an object tracker is achieved. It is notorious that the use of an estimator (in this case a UKF) of the scale and position of the object, improve the velocity of the algorithm and makes it stable against erratic motion and fast recovery against total lost. The use of the Harris points detector combined with SURF descriptors has proved to give a robust way for an object representation. The scheme of constructing multiple descriptors in the first frame gives to the system the scale invariance and it adds a better performance due to the fact that it is done only once and it avoids the use of scale-space in each frame.

References

[1] Chekhlov, D., Pupilli, M., Mayol-Cuevas, W., Calway, A.: Real-time and robust monocular slam using predictive multi-resolution descriptors. In: 2nd International Symposium on Visual Computing (November 2006)

[2] Lowe, D.: Distinctive image features from scale-invariant keypoints. Int. J. Computer Vision 2(60), 91–110 (2004)

[3] Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. IJCV 1(60), 63–86 (2004)

[4] Harris, C.J., Stephens, M.: A combined corner and edge detector. In: Proc. 4th Alvey Vision Conferences, pp. 147–151 (1988)

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

[6] Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. In: Proceedings of the ninth European Conference on Computer Vision (May 2006)

[7] Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Transactions on Pattern Annalysis and Machine Intelligence 25, 564–577 (2003)

[8] Rasmussen, C., Toyama, K., Hager, G.D.: Traking objects by color alone. Technical report, Yale University (June 1996)

[9] Perez, P., Hue, C., Vermaak, J., Gagnet, M.: Color-based probabilistic tracking. In: ECCV, pp. 661–675 (2002)

[10] Pahlavan, K., Eklundh, J.O.: A head-eye system- analysis and design. CVGIP: Image Understanding 56, 41–56 (1992)

[11] Tissainayagam, P., Suter, D.: Object tracking in image sequences using point features. Pattern Recognition 38, 105–113 (2005)

[12] Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: CVPR, pp. 142–149 (2000)

[13] Serby, D., Meier, E.K., Gool, L.V.: Probabilistic object tracking using multiple features. In: ICPR 2004, pp. 184–187 (2004)

[14] Cipolla, R., Yamamoto, M.: Stereoscopic tracking of bodies in motion. Image and Vision Computing 8(1), 85–90 (1990)

[15] Blake, A., Curwen, R., Zisserman, A.: A framework for spatiotemporal control in the tracking of visual contours. Int. J. Computer Vision 11(2), 127–145 (1993)

[16] Blake, A., Isard, M.: Active Contours. Springer, London (1998)

[17] Julier, J., Uhlmann, K.: Unscented filtering and nonlinear estimation. Proceedings of the IEEE 93, 401–422 (2004)

[18] Julier, S.J., Uhlmann, J.K.: A new extension of the kalman filter to nonlinear systems. In: AeroSense: The 11th Int. Symp. on Aerospace/Defence Sensing, Simulation and Controls (1997)

[19] Doucet, A., Godsill, S., Andrieu, C.: On sequential monte carlo sampling methods for bayesian filtering. Statistics and Computing 10(3), 197–208 (2000)

[20] Perez, P., Vermaak, J., Blake, A.: Data fusion for visual tracking with particles. Proceedings of the IEEE 92, 495–513 (2004)

[21] Liu, J., Chen, R.: Sequential monte carlo methods for dynamic systems. American Statistical Association 93, 1032–1044 (1998)

[22] Viola, P., Jones, M.: Rapid object detection using boosted cascade of simple features. Computer Vision and Pattern Recognition 1, 511–518 (2001)

[23] Heckbert Paul, S.: Fundamentals of texture mapping and image warping. Master's thesis, University of California, Berkeley. Dept. of Electrical Engineering and Computer Science (June 1989)

[24] Video of the object I. http://www.youtube.com/watch?v=bpfkVq-w53E

[25] Video of the object II. http://www.youtube.com/watch?v=eR0IMpRK0c8