ROBUST OBJECT TRACKING BY SIMULTANEOUS GENERATION OF AN OBJECT MODEL

Maria Sagrebin, Daniel Caparròs Lorca, Daniel Stroh and Josef Pauli
Fakultät für Ingenieurwissenschaften
Abteilung für Informatik und Angewandte Kognitionswissenschaft
Universität Duisburg-Essen, Germany

Keywords: Model-based object tracking, Online model generation.

Abstract: Although robust object tracking has a wide variety of applications ranging from video surveillance to recognition from motion, it is not completely solved. Difficulties in tracking objects arise due to abrupt object motion, changing appearance of the object or partial and full object occlusions. To resolve these problems, assumptions are usually made concerning the motion or appearance of an object. However in most applications no models of object motion or appearance are previously available. This paper presents an approach which improves the performance of a tracking algorithm due to simultaneous online model generation of a tracked object. The achieved results testify the stability and the robustness of this approach.

1 INTRODUCTION

Robust object tracking is an important task within the field of computer vision. It has a variety of applications ranging from motion based recognition, automated video surveillance, traffic monitoring, up to vehicle navigation and human-computer interaction. In all situations the main goal is to extract the motion of an object and to localize it an image sequence. Thus in its simplest form, tracking is defined as the problem of estimating the trajectory of an object in an image sequence as it moves in the scene. In high level applications this information could be used to determine the identity of an object or to recognize some unusual movement pattern to give a warning.

Although good solutions are widely desired object tracking is still a challenging problem. It becomes complex due to noise in images, abrupt changes in object motion and partial and full object occlusions. Changing appearance of an object and changes in scene illumination also cause problems a good tracking method has to deal with.

Numerous approaches for object tracking have been proposed. Depending on the environment in which tracking is to be performed they differ in object representations or how motion, appearance or shape is modeled. Jepson et al. (Jepson et al., 2003) proposed an object tracker that tracks an object as a three component mixture, consisting of stable appearance features, transient features and a noise process. An online version of the Expectation-maximization (EM) algorithm is used to learn the parameters of these components. Comaniciu et al. (Comaniciu et al., 2003) used a weighted histogram computed from a circular region to represent the moving object. The mean-shift tracker maximizes the similarity between the histogram of the object and the histogram of the window around the hypothesized object location. In 1994, Shi and Tomasi proposed the KLT tracker (Shi and Tomasi, 1994) which iteratively computes the translation of a region centered on an interest point. Once a new position is obtained the tracker evaluates the quality of the tracked patch by computing the affine transformation between pixel corresponding patches.

Thus usually some of the problems described earlier are resolved by imposing constraints on the motion or appearance of objects. As stated by Yilmaz (Yilmaz et al., 2006) almost all tracking algorithms assume that the object motion is smooth with no abrupt changes. Sometimes prior knowledge about the size and shape of an object is also used to simplify the tracking task. However the higher the number of imposed constraints the smaller is the area of applicability of the developed tracking algorithm.

This paper presents an approach which implies no prior knowledge about the motion or appearance of an object. Instead the appearance of a previously un-

Sagrebin M., Caparròs Lorca D., Stroh D. and Pauli J. (2009).
ROBUST OBJECT TRACKING BY SIMULTANEOUS GENERATION OF AN OBJECT MODEL.
In *Proceedings of the Fourth International Conference on Computer Vision Theory and Applications*, pages 392-397
DOI: 10.5220/0001659803920397

known object is learned online during an initial phase. First all moving objects in the scene are detected by using a background subtraction method. Then distinctive features are extracted from these objects and correspondences of detected object regions across different images are established by using a point tracker. The used point tracker does have several drawbacks. However, during the initial phase it provides enough information to the system to learn the appearance of a tracked object. The generated multiview appearance model is then used to detect an object in subsequent images and thus helps to improve performance of a tracking method.

In the following every step of the described approach is discussed in detail. First the used background subtraction method and the point tracker are presented. The developed algorithm for online generation of an object appearance model is described in section 4. In subsequent sections the results and possible improvement to this approach are discussed.

2 ADAPTIVE BACKGROUND SUBTRACTION

If the amount and the shape of moving objects in the scene is unknown, background subtraction is an appropriate method to detect those objects. Although several background subtraction algorithm have been proposed, all of them are based on the same idea. First the so called background model is built. This model represents the naked scene or the observed environment without any objects of interest. Moving objects are then detected by finding deviations from the background model in the image.

Background subtraction became popular following the work of Wren et al. (Wren et al., 1997). He proposed a method where each pixel of the background is modeled by a Gaussian distribution. The mean and the covariance parameters for each distribution are learned from the pixel value observations in several consecutive images. Although this method shows good results, a single Gaussian distribution is not appropriate to model multiple colors of a pixel. This is necessary in the case of small repetitive movements in the background, shadows or reflections. For example pixel values resulting from specularities on the surface of water, from monitor flicker or from slightly rustling of leaves can not be modeled with just one Gaussian distribution. In 2000, Staufer and Grimson (Stauffer and Grimson, 2000) used instead a mixture of Gaussians to model a pixel color. This background subtraction method is used in the proposed approach to detect moving objects. In the following it is presented in more detail.

Each pixel in the image is modeled by a mixture of K Gaussian distributions. K has a value from 3 to 5. The probability that a certain pixel has a color value of X at time t can be written as

$$p(X_t) = \sum_{i=1}^{K} w_{i,t} \cdot \eta(X_t;\theta_{i,t})$$

where $w_{i,t}$ is a weight parameter of the i-th Gaussian component at time t and $\eta(X_t;\theta_{i,t})$ is the Normal distribution of i-th Gaussian component at time t represented by

$$\eta(X_t;\theta_{i,t}) = \frac{1}{(2\pi)^{\frac{n}{2}} \left|\Sigma_{i,t}\right|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t-\mu_{i,t})^T \Sigma_{i,t}^{-1}(X_t-\mu_{i,t})}$$

where $\mu_{i,t}$ is the mean and $\Sigma_{i,t} = \sigma_{i,t}^2 \mathrm{I}$ is the covariance of the i-th component at time t. It is assumed that the red, green and blue pixel values are independent and have the same variances.

The K Gaussians are then ordered by the value of $w_{i,t}/\sigma_{i,t}$. This value increases both as a distribution gains more evidence and the variance decreases. Thus this ordering causes that the most likely background distributions remain on top.

The first B distributions are then chosen as the background model. B is defined as

$$B = \arg\min_{b} \left(\sum_{i=1}^{b} w_i > T \right)$$

where the threshold T is the minimum portion of the background model. Background subtraction is done by marking a pixel as foreground pixel if its value is more than 2.5 standard deviations away from any of the B distributions.

The first Gaussian distribution that matches the pixel value is updated by the following equations

$$\mu_{i,t} = (1-\rho)\,\mu_{i,t-1} + \rho X_t$$

$$\sigma_{i,t}^2 = (1-\rho)\,\sigma_{i,t-1}^2 + \rho\,(X_t-\mu_{i,t})^T\,(X_t-\mu_{i,t})$$

where

$$\rho = \alpha\eta\,(X_t\,|\mu_i,\sigma_i)$$

and α is the learning rate. The weights of the i-th distribution are adjusted as follows

$$w_{i,t} = (1-\alpha)\,w_{i,t-1} + \alpha(M_{i,t})$$

where $M_{i,t}$is set to one for the distribution which matched and zero for the remaining distributions.

Figure 1 shows some results of the background subtraction method. For this experiment a simple off the shelf web camera was placed in one corner of the room facing the center. The background model of the scene was learned from the first 200 images taken from that camera. Moving objects in the scene were then detected by the deviation of pixel colors from the background model. As shown in the lower image all moving objects have been successfully detected.

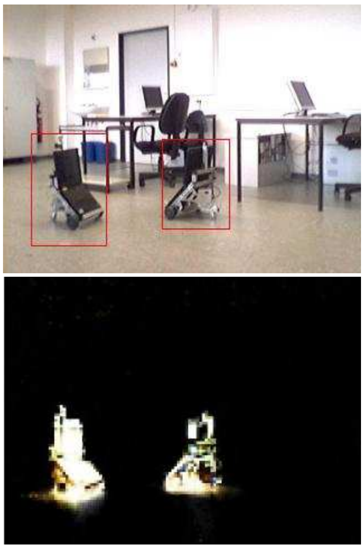

Figure 1: Results of the background subtraction.

Here moving objects are represented by so called foreground pixels. Pixels which belong to the background have a constant black value. Usually for later processing foreground pixels are grouped into regions by a connected components algorithm. This allows to treat an object as a whole. After labeling the components axis parallel rectangles around these regions have been computed. The results are shown in the left image. The centers of these rectangles have been used to specify the positions of the objects in the image.

The positions and rectangles around all detected objects in the scene form the input to the object tracking method. This method is described in more detail in the next section.

3 OBJECT TRACKING

The aim of an object tracker is to establish correspondences between objects detected in several consecutive images. Assuming that an object is not moving fast, one could treat overlapping rectangles $r_{i,t}$ and $r_{j,t+1}$ in two consecutive images t and $t+1$ as surrounding the same object. However as stated above this assumption imposes a constraint on the motion model of an object. In the case of a small fast moving object the rectangles around this object in different images will not overlap and the tracking will fail.

To overcome this problem a different tracking approach was used. The input to this method consists of rectangles around moving objects which have been previously computed in two consecutive images. To establish correspondences between rectangles the following steps have been conducted.

- Extraction of distinctive features in both images.
- Finding correspondences between the extracted features.
- Identifying corresponding rectangles. Two rectangles are considered as being correspondent when features they surround are corresponding to each other.

Figure 2 shows how two robots have been tracked. The upper and lower pictures show images taken from the same camera at timestamps t and $t+1$ respectively. First based on the background subtraction method the robots have been found and then by using the connected components algorithm rectangles around these robots have been computed. Simultaneously, distinctive features have been extracted and tracked over these two images. In the upper image those features are depicted through dots. Computed rectangles were used to select features which represent the moving objects. Corresponding rectangles in two consecutive images have been identified through corresponding features which lie inside those rectangles. The short lines on moving objects in the lower image depict trajectories of successfully tracked features which have been found on the robots. This approach is not restricted to a special kinds of fea-

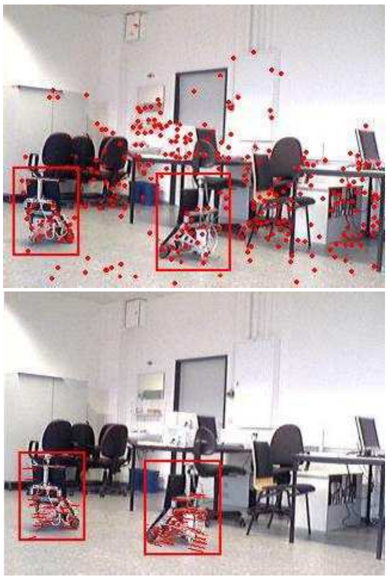

Figure 2: Tracking of moving objects in the scene.

tures. Several interest point detectors can be used. Morevec's interest operator (Moravec, 1979), Harris corner detector (Harris and Stephens, 1988), Kanade-Lucas-Tomasi (KLT) detector (Shi and Tomasi, 1994) and Scale-invariant feature transform (SIFT) detector (Lowe, 2004) are often mentioned in the literature. However in experiments presented here, SIFT detector has been used. According to the survey by Mikolajczyk and Schmid (Mikolajczyk and Schmid, 2003) this detector outperforms most point detectors and is more resilient to image deformations.

SIFT features were introduced by Lowe (Lowe, 2004) in 1999. They are local and based on the appearance of the object at particular interest points. They are invariant to image scale and rotation. In

to these properties, they are also highly distinctive, relatively easy to extract and are easy to match against large databases of local features.

As seen on the left image in Figure 2 features are extracted not only on moving objects but also all over the scene. Previously computed rectangles around the objects which were detected by background subtraction are used to cluster and to select features of interest. These are features which are located on moving objects. However one could also think of clustering the features due to the length and angle of the translation vectors defined by two corresponding features in two images. Features located on moving objects produce longer translational vectors as those located on the background. In such an approach no background subtraction would be necessary and a lot of computational cost would be saved. This method was also implemented but led to very unstable results and imposed too many constraints on the scene and on the object motion. For example to separate features which represent a slowly moving object from those which lie on the background it was necessary to define a threshold. It is easy to see that this approach is not applicable when for example an object stops moving before changing its direction. Later it will be shown that the results of the background subtraction method are also used to generate a model of an object.

Although the proposed tracking method was successfully tested in an indoor environment, it does have several drawbacks. The method tracks objects based on the features which have been extracted and tracked to the next image. However point correspondences can not be found in every situation. It especially becomes complicated in the presence of partial or full occlusions of an object. The same problem arises due to changing appearance of an object. Usually an object does have different views. If it rotates fast only few or no point correspondences at all can be found.

The above problems could be solved if a multiview appearance model of a tracked object would be available. In that case an object could have been detected in every image, hereby improving the performance of the tracking algorithm. Usually such models are created in an offline phase. However, in most applications it is not practicable. Such an approach also restricts the number of objects which can be tracked. Depending on the application, an expert usually defines the amount and the kind of objects to be tracked and trains the system to recognize these objects. Next section describes how a multiview appearance model of a previously unknown object is generated online.

4 GENERATION OF AN OBJECT MODEL

Usually objects appear different from different views. If such an object performs a rotational movement the system has to know the different views of that object so that it can track it. A suitable model for that purpose is a so called multiview appearance model. A multiview appearance model encodes the different views of an object so that it can be recognized in different positions.

Several works have been done in the field of object tracking using multiview appearance models. For example Black and Jepson (Black and Jepson, 1998) proposed a subspace based approach, where a subspace representation of the appearance of an object was built using Principal Component Analysis. In 2001, Avidan (Avidan, 2001) used a Support Vector Machine classifier for tracking. The tracker was previously trained based on positive examples consisting of images of an object to be tracked and negative examples consisting of things which were not to be tracked. However, such models are usually created in an offline phase, which is not practicable in many applications.

In the approach presented here a multiview appearance model of a previously unknown object is generated online. The input to the model generation module consists of all moving objects detected by background subtraction and all correspondences established through the object tracker described in section 3. Figure 3 shows graphically the procedure which is followed by the model generation module.

After background subtraction was performed and moving objects were detected in the image, image regions representing those objects were used to build models of them. In the database a model of an object was stored as a collection of different views of that object. The database was created and updated us-

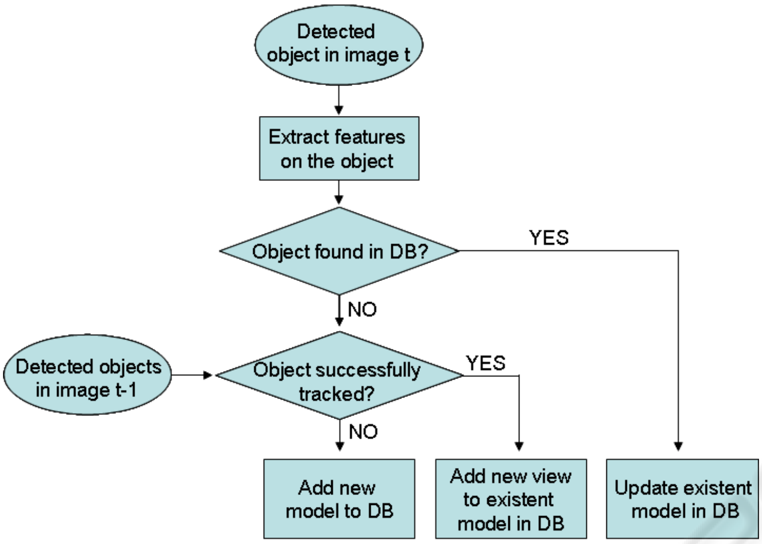
Detected object in image t
Extract features on the object
Object found in DB?
YES
NO
Detected objects in image t-1
Object successfully tracked?
YES
NO
Add new model to DB
Add new view to existent model in DB
Update existent model in DB

Figure 3: Graphical representation of a workflow of the model generation module.

ing the vision system of the Evolution Robotics ERSP platform. Each view of an object was represented by a number of SIFT features, which have been extracted from the corresponding image of that object. Recognition was performed by extracting SIFT features in a new image and by matching them, based on the similarity of feature texture, to the features stored in the database. The potential object matches were then refined by the computation of an affine transform between the new image and the corresponding view stored in the database. Next all different cases shown in figure 3 are described in more detail.

Case 1. In the very beginning when the system starts working and the database is empty, there is no matching of any extracted object with the database. And since it is the first time that this object is detected it is not tracked either. Thus a new model consisting of one view only is added to the database.

Case 2. If the same object could be tracked to the next image but the new view could not be matched with the database then this new view is added to the model which matched the view of the object in the previous image. This situation can occur when an object is rotating and thus changing its appearance. In this case only a few features can be tracked. However these correspondences are usually not sufficient to match the new view to the one already existing.

Case 3. The detected object can be matched with the database. In this case two situations can arise. In the first case the object is matched to one model in the database only. If the matching score is low than a new view is added to that model. In the second case the object is matched to more than one model in the database. In this situation all views of these models are fused together. It is suggested that these views describe the same object and therefore belong to the same model.

The second situation in the third case arises when different models in the database represent the same object. Different models of the same object could have been created when an object performs an abrupt motion and neither the new view is matched with the database nor the tracker is able to track the object. However after a period of time, if a new view can be matched to these different models in the database, these models will be fused together. The duration of this period depends on the motion of the object. Figure 4 shows the multiview model of a robot which was created with the proposed approach.

Different views of the robot have been stored in the database. In this figure points depict the positions of SIFT features which have been extracted from these images. The generated model is specific to camera position and orientation. No views are stored in the database which can not be obtained from that particular camera. This results in a very accurate and concise representation of an object. The obtained database is optimally fitted into the application environment.

The described algorithm for model generation can also work if no point tracker is available. In that case the query for a successfully tracked object in figure 3 would always result in ’NO’. However experiments have shown that an adequate point tracker reduces the time needed to generate an object model.

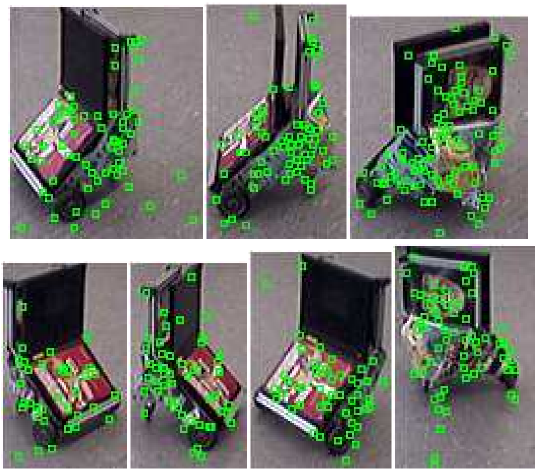

Figure 4: Online generated model of the robot.

With object models being available, objects were tracked more robustly. The trajectory of an object was estimated via results of the feature tracker presented in section 3 in combination with the object detection mechanism which located objects based on their model stored in the database. Even in the case of temporary partial or full occlusions of the tracked object tracking was stable and could have been continued. After the object reappeared, it was detected by the system based on the previously stored model in the database.

5 CONCLUSIONS

In this paper a new object tracking approach was presented which autonomously improves its performance by simultaneous generation of object models. The different multiview appearance models are created online. The presented approach requires no previous training nor manual initialization. With such characteristics it is very good suitable for automated surveillance. Since the method automatically creates a model of moving objects, it can also be used to retrieve information when or how often a particular object was moving in the scene.

Although the presented approach was successfully tested in an indoor environment, it sometimes suffers from one problem which will be eliminated in the future. Background subtraction does not work perfectly. Due to noise or abrupt illumination changes artifacts can arise. The resulting foreground image does not contain only moving objects, but also some clusters of pixels which actually belong to the background. Since the system has no previous knowledge about objects to track it treats these clusters as moving objects and starts to generate an appearance model. The idea to overcome this problem is to develop a module which monitors trajectories and appearances of objects. Clusters of falsely classified pixels which actually belong to the background do not move and their appearance does not change. Based on that information wrongly created models can be deleted from the database.

REFERENCES

Avidan, S. (2001). Support vector tracking. *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 184–191.

Black, M. and Jepson, A. (1998). Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *Int. J. Comput. Vision 26*, 1:63–84.

Comaniciu, D., Ramesh, V., and Meer, P. (2003). Kernel-based object tracking. *IEEE Trans.Patt.Analy.Mach.Intell*, 25:564–575.

Harris, C. and Stephens, M. (1988). A combined corner and edge detector. *In 4th Alvey Vision Conference*, pages 147–151.

Jepson, A., Fleet, D., and ElMaraghi, T. (2003). Robust online appearance models for visual tracking. *IEEE Trans.Patt.Analy.Mach.Intell*, 25:1296–1311.

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

Mikolajczyk, K. and Schmid, C. (2003). A performance evaluation of local descriptors. *In European Conference on Computer Vision and Pattern Recognition*, pages 1615–1630.

Moravec, H. (1979). Visual mapping by a robot rover. *In Proceedings of the International Joint Conference on Artificial Intelligence*, pages 598–600.

Shi, J. and Tomasi, C. (1994). Good features to track. *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600.

Stauffer, C. and Grimson, W. (2000). Learning patterns of activity using real time tracking. *IEEE Trans.Patt.Analy.Mach.Intell. 22*, 8:747–767.

Wren, C., Azarbayejani, A., and Pentland, A. (1997). Pfinder: Real-time tracking of the human body. *IEEE Trans.Patt.Analy.Mach.Intell. 19*, 7:780–785.

Yilmaz, A., Javed, O., and Mubarak, S. (2006). Object tracking: A survey. *ACM Computing Surveys*, 38.