A COMPARISON OF DIFFERENT APPROACHES TO NONLINEAR SHIFT ESTIMATION FOR OBJECT TRACKING

M. Asadi, C.S. Regazzoni

Department of Biophysical and Electronic Engineering, University of Genoa, Genoa, Italy

ABSTRACT

This paper presents a corner based voting method for estimating the object shift in video image frames. Information about the corners distribution around a reference point is used to represent the object shape and then to find the most probable target position in the next frame. Tracking is done through using a voting space obtained by matching corners information. A motion vector for the reference point is nonlinearly estimated with three different strategies by using the global information of the matched corners. The results show a comparison between three considered strategies for estimating the object shift.

Index Terms— Object tracking, corner-based tracking

1. INTRODUCTION

During the last two decades, many researches have been done on tracking for the purpose of security surveillance. Considering security surveillance, tracking methods can be classified into two main groups: feature-based and model-based methods. One way to describe the objects shape model is using high curvature points called corners [1]. In literature, corners information for tracking has been used for two motivations: using corners to detect the object and using them to correct predictions based on filters like Kalman. Oberti et al. used a model similar to the one used here to detect the object inside a predefined search area [2]. Again, the same model has been used in another work as a part of tracking algorithm to recognize the objects in the scene when an overlap happens between their bounding boxes [3]. Gabriel et al. [4] exploit corners information to correct the prediction of the object position done by Kalman filtering. Finally, Wei et al. applies KLT on the object corners to detect the velocities [5]. After clustering the corners based on their spatial location and motion, each cluster is considered as a mixture component modeled by an individual particle filter.

This paper presents a nonlinear shift estimation for a given reference point. The reference point is associated with an object described using the object corners information. To estimate the object shift, a new corners set is formed in any iteration and a voting mechanism is applied on the set. Therefore, corners of the set vote in a voting space based on the last object model. Then, the corners in the current iteration are associated with the ones in the model. Using the voting space and associated corners information, the new position for the reference point is estimated with three different strategies: a) averaging associated corners motion vectors, b) averaging associated corners motion vectors after removing outliers by using rank order filtering, and c) finding the position with the highest number of votes as the most probable reference point. These three estimation strategies are described and compared in experimental results. Finally, after all iterations, the model is updated.

Since the reference point is estimated through using corners global information, and the corners represent the object structure, one iteration is enough for estimation. The other iteration is just to consider all object corners to have a more exhaustive list of the associated corners for updating the model.

The main contributions to the paper are that the proposed method -despite the existing corner-based tracking methods- neither does search in an area to detect the object -but just estimate its location-, nor does it use other tools such as Kalman filtering or KLT. Moreover, three different strategies for shift estimation are analyzed.

The rest of the paper is organized as follow: section 2 explains the proposed method in details. The experimental results are shown and discussed in section 3. Finally, conclusion and future works appear in section 4.

2. THE PROPOSED ALGORITHM

Starting from the reference image frame, a user may select a target -with a bounding box- to be tracked in sequence. The center of the bounding box is considered as the reference point that must be tracked. The SUSAN corner extractor [1] extracts corners inside the bounding box and the target model will be formed through using these corners. In the next frame, a *target candidate* is assumed in the same position and of the same size as the previous bounding box, assuming that the movement of the object in two successive frames is such that they have overlapped each other. Extracting the corners of this region, a matching step is done among the newly extracted corners *(candidate corners)* and the ones in the model *(model corners),* and the candidate corners vote in the voting space. After regularization, three different strategies are applied to estimate the amount of the object shift. Finally, the model will be updated. Different steps of the algorithm are explained in detail in the following sections.

2.1. Target shape model

As mentioned above, a target is represented by a set of corners $\{C_i\}$ $1 \le i \le n$ inside the bounding box. Each given corner is considered as a vector of some informative elements:

$$C_i = [\omega_i, dx_i, dy_i, x_i, y_i, p_i, a_i] \qquad 1 \le i \le n \qquad (1)$$

where ω_i is the gradient angle of the corner i, x_i and y_i are its coordinates, dx_i and dy_i are its relative coordinates of the corner with respect to the reference point (x_{ref}, y_{ref}):

$$d_{x_i} = x_{ref} - x_i \quad , \quad d_{y_i} = y_{ref} - y_i \qquad (2)$$

p_i is the persistency value and a_i is the appearance time of the corner, and they are set to 1 and 0, respectively, and are used in the updating step. ω_i is defined as follows:

$$\omega_i = arctg\left(\nabla I(x_i, y_i)\right) \quad , \quad \omega_i \in [-\pi, \pi] \qquad (3)$$

$$\nabla I(x_i, y_i) = \left[\frac{\partial I(x_i, y_i)}{\partial x}, \frac{\partial I(x_i, y_i)}{\partial y}\right] \qquad (4)$$

where $I(x_i, y_i)$ is the luminance value of the i^{th} corner, ∇I is the gradient vector consisting of two elements; i.e. the derivation (∂I) of the image pixels with respect to both axes, x, y. Finally, $arctg(.)$ is the inverse tangent function.

2.2. Target candidate

Having an area inside the bounding box, a target candidate is represented by the set of the area corners $\{C_j(t)\}$ $1 \le j \le m$:

$$C_j(t) = [\omega_j(t), x_j(t), y_j(t)] \qquad 1 \le j \le m \qquad (5)$$

where $\omega_j(t)$, $x_j(t)$, $y_j(t)$ are the gradient angle and coordinates of the j^{th} corner at time t.

2.3. Voting space structure

The voting space used here is a two dimensional space with Cartesian coordinates corresponding with the image plane. Each position value of the space is set to zero, initially, but it increases by one for every vote to that position. Every corner based on the voting mechanism may vote to any position.

2.4. Voting mechanism

Having the model in frame $t-1$ and a target candidate in frame t represented by (1) and (5) respectively, each candidate corner is compared with all model corners based on their gradient angles (6) and if their difference is less than a given threshold, it will vote for a possible reference point based on this model corner (7).

$$\left|\omega_j(t) - \omega_i(t-1)\right| \le T_{angle} \qquad (6)$$

$$\begin{cases} voted_x(t) = x_j(t) + dx_i(t-1) \\ voted_y(t) = y_j(t) + dy_i(t-1) \end{cases} \qquad (7)$$

Furthermore, the identities of these two corners are associated with the voted position as a pair (figure 1). Therefore, after the voting phase, a list of pairs associated with any voted position in the voting space will be available.

2.5. Regularization

Since the object has small changes in two successive frames, and as a result the relative coordinates of corresponding object corners -with respect to the reference point- are not fixed, the corners in the current frame do not vote to a fixed point as the reference point. Instead, they vote to an area around it. Therefore, it is necessary to do regularization: sweep the voting space with a window mask (3x3 or 5x5) and substitute the values of the voting space positioned on the center of the mask with the summation of the values of the voting space that overlap the mask. Then, the shift of the object must be estimated and finally, the model is updated. The estimation strategies are discussed in a separated section.

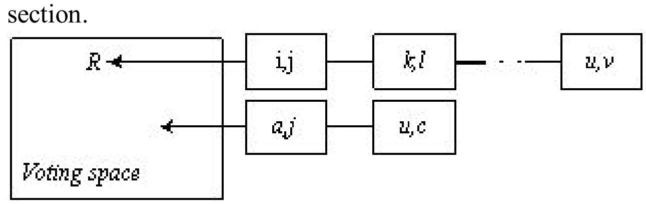

Figure 1: voting space along with two lists of the pairs associated with two voted positions in the voting space.

2.6. Updating strategy

After tracking the target, the model is updated using the union of those lists of the pairs that have been associated with the point in the voting space with maximum number of votes and its surrounding area; and hence the changes in the number of the corners, their gradient angles and relative coordinates are applied in the model, making the model ready to use in the next frame. Every element of a pair is called a *partner* for the other element. To update the model, every corner in the union list of the pairs must have a unique partner. For any corner in the list that has more than one partner, the partner closer to the element is chosen based on the Euclidean distances between all of them and that element. The Euclidean distance is based on their relative coordinates from the reference point. The algorithm uses a persistency value P_i for each corner and it is set to 1 the first time that a corner appears. It will increase by 1 if the corner has a unique match, otherwise, it will decrease by 1.

Once corner i has a unique match k, its values are updated:

$$\begin{cases} dx_i(t-1)=dx_k(t) & x_i(t-1)=x_k(t) \\ dy_i(t-1)=dy_k(t) & y_i(t-1)=y_k(t) \\ \omega_i(t-1)=\omega_k(t) & P_i(t-1)=P_i(t-1)+1 \end{cases} \quad (8)$$

Appearance value a_i is the index of the frame in which corner i appeares for the first time. a_i and P_i are used to discard the corners that are not persistent: after appearing for the first time, the corner is given an opportunity equal to some numbers of frames (living time) to increase its persistency and after that, once its persistency goes below a threshold, it will be discarded from the model. After updating the model using the list, all remaining corners in the bounding box in the current frame are added to the model with persistency values equal to one.

The updating strategy discussed above, lets the persistent corners of the object remain in the model, the newly found corners add to the model, and the corners inside the bounding box but belong to the background are removed from the model.

3. ESTIMATING THE OBJECT SHIFT

Three different strategies to estimate object shift are proposed: absolute maximum, averaging the corners motion vectors, and averaging the corners motion vectors after removing the outliers.

3.1. Absolute maximum

Since most object corners vote for the reference point based on the relation among them, and vote to the wrong positions randomly, it is reasonable to consider the point with the maximum number of votes (after regularization) as the point that is most likely the reference point. Considering this point as the newly found reference point in the current frame, the shift of the object equals to the difference between two reference points in two successive frames.
However, there may be some corners in the pairs list (figure 1) that have matched with wrong partners (corners), but have voted to the correct position. This is a rare situation but may happen. “Rare situation” means that in every frame there may be a few number of such pairs comparing to the whole number of the pairs. The advantage of this strategy is that since the aforementioned problem is a rare situation, it does not affect the position with the maximum number of votes. Furthermore, the model updated using these associated corners, is robust enough to be used in the next frames.

3.2. Averaging the corners motion vectors

This is the simplest but noisiest strategy. Having a list of the pairs associated with the point with maximum number of votes (after taking the union with the lists associated with the surrounding points), the motion vectors of the pairs can be computed: let us assume R in figure 1 to be the point with a maximum number of votes. Its associated list indicates that corner j in current frame has voted to R based on the relative coordinates of the model corner i (formula 7) and hence one can say that corner j in current frame is the same as corner i in the previous frame. Therefore, its motion vector can be calculated (figure 2, left, blue dotted lines). The simplest strategy is to say that the motion of the whole object equals to the average motion of all corners in the list (figure 2, left, red dashed line). However, since the voting is based on the gradient angle, there may be some wrong associations as mentioned in 3.1: a corner is matched with a wrong corner but votes for the correct position, randomly. These wrong associations can affect the average result. To overcome this problem, the third strategy is defined to use rank order filtering to remove the outliers, before averaging.

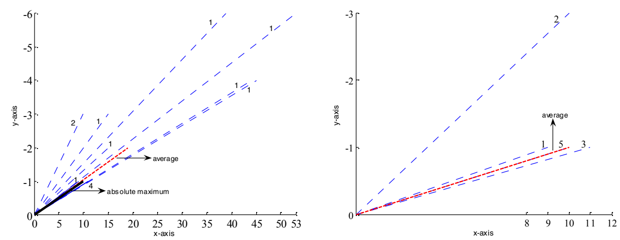

Figure 2: motion vectors (blue dotted lines) of pairs of corners for frame #217 and their average motion vectors (red dashed lines), left: without removing the outliers, right: after removing the outliers, the solid black line at the left is the motion vector obtained by the maximum number of votes

3.3. Averaging after removing the outliers

The aim of this strategy is to remove the pairs that strongly affect the average value (outliers). To do this, motion vectors of the pairs along with their modulus are computed in the same way as the previous strategy and are ranked in ascending order. Then, a percentage of the corners with smaller modulus are considered. This is because the noisy ones are the pairs with higher modulus. Empirically, it was realized that about 60-70% of the pairs are reliable. Yet, this is not general, as this percentage was set statically and that it does not change in different frames. Removing the outliers for the same image frame, a better estimation is achieved (figure 2, right, red dashed line). For instance, figure 2 shows two graphs of motion vectors of the pairs in the list associated with the most probable reference point in frame #217 of figure 3, before and after removing the outliers. It can be seen from figure 2 (left) that the second strategy gives an average shift for the object equal to the vector (19,-2), while the third strategy (figure 2, right) gives an average shift for the object equal to the vector (10,-1). The object shift in this frame -with respect to the previous frame- using the first strategy equals to (10,-1) (figure 2, left, black solid line). In this figure, the numbers near the vectors show the number of corners with the same motion. This again says that as the object does not change much between two successive frames, the corresponding corners motion vectors must be close to each other and that some of them even have the same motion. This fact may be considered to improve the removal of the outliers in the future. The other implicit result is that estimating the motion vector using the absolute maximum strategy as the correct estimation along with removing outliers to filter the wrongly associated corners, can help handle object scaling and rotation, and building 3D object models in the future.

4. EXPERIMENTAL RESULTS

To justify the above discussion and to compare different strategies, results of applying the proposed method using all strategies on an image sequence has been shown in figure 3. Tracking the car starts when the car is completely in the scene. The first row compares the averaging strategies before (left) and after (right) removing the outliers. Without removing the outliers, the tracking will be affected very soon (after 6 frames), while after removing the outliers the method is more robust (up to frame #222). The second row of figure 3 compares the results of averaging after removing the outliers with the ones of absolute maximum. It is seen that at frame #222, the bounding box has been distracted strongly by the outliers.

To show the robustness of the absolute maximum strategy, figure 4 shows the car in different situations in different image frames. Since the model is updated regularly, it is adapted step by step in each frame to overcome the changes. Considering the position of the estimated reference point on the car (that is in the middle part of the car between the doors and the windows of the car) and considering that it may oscillate about the same position by some pixels, the absolute maximum strategy is regarded as a robust strategy.

5. CONCLUSION AND FUTURE WORKS

In this paper a nonlinear shift estimator was introduced that uses the shape corner information as the target model. In this method, finding the new position of the object has been done using a voting mechanism followed by three different strategies. It was shown that the absolute maximum strategy is the most robust strategy. The goodness of the method is that due to the nonlinearity, few iterations are needed to find the proper position. In addition, the method –despite many other existing methods- takes advantage of using only corners information for tracking. Furthermore, the method does not search to detect the object, but it is just attracted toward the proper position and hence can be considered as a tracking-before-detection method.

Handling object scaling and rotation in order to handle the bounding box size to reach to a better visual results and a smaller yet more precise model is considered as future scopes. In addition, keeping a history of the uniquely matched corners with high persistency values, the method has the capability of individual tracking of every corner trajectory that is useful in 3D object modeling.

a)

b)

c)

Figure 3: a) reference frame: frame #211, b)left: frame #217 with averaging, right: frame #217 with averaging after removing outliers, and c) left: frame #222 with averaging after removing outliers, right: frame #222 with absolute maximum strategy

a) b)

c) d)

Figure 4: the tracked car along with the found reference point (red star) using the maximum absolute strategy. The reference point is in the same position of the car: a)frame #230, b)frame #238, c)frame #243, d)frame #246

6. REFERENCES

[1] S.M. Smith, and J.M. Brady, "SUSAN - A New Approach to Low Level Image Processing", *International Journal of Computer Vision,* Kluwer Academic Publishers, MA, USA, Volume 23, Issue 1, May 1997, pp. 45-78.

[2] F. Oberti, S. Calcagno, M. Zara, and C. S. Regazzoni, Robust tracking of humans and vehicles in cluttered scenes with occlusions, IEEE ICIP, 3:629–632, USA, September 2002.

[3] L. Marcenaro, L. Marchesotti, and C.S. Regazzoni, Self-organizing shape description for tracking and classifying multiple interacting objects, Elsevier Image and Vision Computing, 24, 2006, 1179-1191.

[4] P. Gabriel, J.B. Hayet, J. Piater, and J. Verly, Object tracking using color interest points, IEEE AVSS, 159-164, Italy, September 2005.

[5] D. Wei, and J. Piater, Tracking by cluster analysis of feature points using a mixture particle filter, IEEE AVSS, 165–170, Italy, September 2005.