Joint Random Sample Consensus and Multiple Motion Models for Robust Video Tracking

Petter Strandmark[1,2] and Irene Y.H. Gu[1]

[1] Dept. of Signals and Systems, Chalmers Univ. of Technology, Sweden
{irenegu,petters}@chalmers.se
[2] Centre for Mathematical Sciences, Lund University, Sweden
petter@maths.lth.se

Abstract. We present a novel method for tracking multiple objects in video captured by a non-stationary camera. For low quality video, RANSAC estimation fails when the number of good matches shrinks below the minimum required to estimate the motion model. This paper extends RANSAC in the following ways: (a) Allowing multiple models of different complexity to be chosen at random; (b) Introducing a conditional probability to measure the suitability of each transformation candidate, given the object locations in previous frames; (c) Determining the best suitable transformation by the number of consensus points, the probability and the model complexity. Our experimental results have shown that the proposed estimation method better handles video of low quality and that it is able to track deformable objects with pose changes, occlusions, motion blur and overlap. We also show that using multiple models of increasing complexity is more effective than just using RANSAC with the complex model only.

1 Introduction

Multiple object tracking in video has been intensively studied in recent years, largely driven by an increasing number of applications ranging from video surveillance, security and traffic control, behavioral studies, to database movie retrievals and many more. Despite the enormous research efforts, many challenges and open issues still remain, especially for multiple non-rigid moving objects in complex and dynamic backgrounds with non-stationary cameras. Despite that human eyes may easily track objects with changing poses, shape, appearances, illuminations and occlusions, robust machine tracking remains a challenging issue.

Blob-tracking is one of the most commonly used approaches, where a bounding box is used for a target object region of interest [6]. Another family of approaches is through exploiting local point features of objects and finding correspondences between points in different image frames. Scale-Invariant Feature Transform (SIFT) [7] is a common local feature extraction and matching method that can be used for tracking. Speeded-Up Robust Features (SURF) [1], has been proposed for speeding up the SIFT through the use of integral images. Both methods provide high-dimensional (e.g. 128) feature descriptors that are invariant to object rotation and scaling, and affine changes in image intensities.

A.-B. Salberg, J.Y. Hardeberg, and R. Jenssen (Eds.): SCIA 2009, LNCS 5575, pp. 450–459, 2009.

Typically, not all correspondences are correct. Often, a number of erroneous matches far away from the correct position are returned. To alleviate this problem, RANSAC [3] is used to estimate the inter-frame transformations [2,4,5,8,10,11]. It estimates a transformation by choosing a random sample of point correspondences, fitting a motion model and counting the number of agreeing points. The transformation candidate with the highest number of agreeing points is chosen (consensus). However, the number of good matches obtained by SIFT or SURF may often momentarily be very low. This is caused by motion blur and compression artifacts for video of low quality, or by object deformations, pose changes or occlusion. If the number of good matches shrinks below the minimum required number needed to estimate the prior transformation model, RANSAC will fail. A key observation is that it is difficult to predict whether a sufficient number of good matches is available for transformation estimation, since the ratio of good matches to the number of outliers is unknown.

There are other methods for removing outliers from a set of matches. [12] recently proposed a method with no prior motion model. However, just like RANSAC the methods assumes that several correct matches are available, which is not always the case for the fast-moving video sequences considered in this work.

Motivated by the above, we propose a robust estimation method by allowing multiple models of different complexity to be considered when estimating the inter-frame transformation. The idea is that when many good matches are available, a complex model should be employed. Conversely, when few good matches are available, a simple model should be used. To determine which model to choose, a probabilistic method is introduced that evaluates each transformation candidate using a prior from previous frames.

2 Tracking System Description

To give a big picture, Fig. 1 shows a block diagram of the proposed method. For a given image $\mathbf{I}_t(n, m)$ at the current frame t, a set of candidate feature points $\mathbf{F}_t^c$ are extracted from the entire image area (block 1). These features are then matched against the feature set of the tracked object $\mathbf{F}_{t-1}^{\text{obj}}$, resulting in a matched feature subset $\mathbf{F}_t' \subset \mathbf{F}_t^c$ (block 2). The best transformation is estimated by evaluating different candidates with respect to the number of consensus points and an estimated probability (block 3). The feature subset $\mathbf{F}_t'$ is then updated by

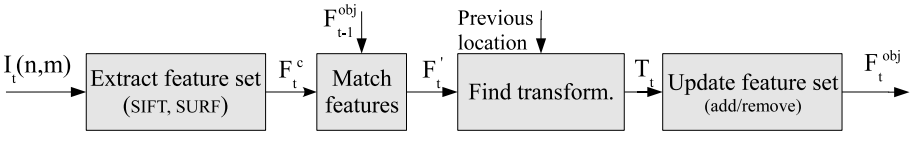

Fig. 1. Block diagram for the proposed tracking method

allowing adding new features within the new object location (block 4). Within object intersections or overlaps updating is not performed. This yields the final feature set for the tracked object $\mathbf{F}_t^{\text{obj}}$ in the current frame t. Block 3 and 4 are described in section 3 and 4, respectively.

3 Random Model and Sample Consensus

To make the motion estimation method robust when the number of good matches becomes very low, our proposed method, RAMOSAC, chooses both the model used for estimation and the sample of point correspondences randomly. The main novelties are: **(a)** Using four types of transformations (see section 3.1), we allow the model itself to be chosen at random from a set of models of different complexity. **(b)** A probability is defined to measure the suitability of each transformation candidate, given the object locations in previous frames. **(c)** The best suitable transformation is determined by the maximum score, defined as the combination of the number of consensus points, the probability of the given candidate transformation, and the complexity of the model. It is worth mentioning that while RANSAC uses only the number of consensus points as the measure of a model, our method differs by using a combination of the number of consensus points and a conditional probability to choose a suitable transformation. Briefly, the proposed RAMOSAC operates in an iterative fashion similar to RANSAC in the following manner:

1. Choose a model at random;
2. Choose a random subset of feature points;
3. Estimate the model using this subset;
4. Evaluate the resulting transformation based on number of agreeing points and the probability given the previous movement.
5. Repeat 1–4 several times and choose the candidate T with the highest score.

Alternatively, each of the possible motion models could be evaluated a fixed number of times. However, because the algorithm is typically iterated until the next frame arrives, the total number of iterations is not known. Choosing a model at random every iteration ensures that no motion model is unduly favored over another. Detailed description of RAMOSAC will be given in the remaining of this section.

3.1 Multiple Transformation Models

Several transformations are included in the object motion model set. The basic idea behind is to use a range of models with an increasing complexity, depending on the (unknown) number of correct matches available. A set of transformation models $\mathcal{M} = \{\mathcal{M}_a, \mathcal{M}_s, \mathcal{M}_t, \mathcal{M}_p\}$ is formed which consists of 4 candidates:

1. Pure translation $\mathcal{M}_t$, with 2 unknown parameters;
2. Similarity transformation $\mathcal{M}_s$, with 4 unknown parameters: rotation, scaling and translation;

3. Affine transformation $\mathcal{M}_a$, with 6 unknown parameters;
4. Projective transformation (described by a 3×3 matrix) $\mathcal{M}_p$, with 8 unknown parameters (since the matrix is indifferent to scale).

The minimum required number of correspondence points for estimating the parameters for the models $\mathcal{M}_t$, $\mathcal{M}_s$, $\mathcal{M}_a$ and $\mathcal{M}_p$ are $n_{\min}$=1, 2, 3 and 4, respectively. If the number of correspondence points available is larger than the minimum required number, least-squares (LS) estimation should be used to solve the over-determined set of equations.

One can see that a range of complexity is involved in these four types of transformations: The simplest motion model is translation, which can be described by a single point correspondence, or by the mean displacement if more points are available. If more matched correspondence points are available, a more detailed motion model can be considered: with a minimum of 2 matched correspondences, the motion can be descried in terms of scaling, rotation and translation by $\mathcal{M}_s$. With 3 matched correspondences, affine motion can be described by adding more parameters such as skew and separate scales in two directions using $\mathcal{M}_a$. With 4 matched correspondences, projective motion can be described by the transformation $\mathcal{M}_p$, which completely describes the image transformation of a surface moving freely in 3 dimensions.

3.2 Probability for Choosing a Transformation

To assess whether a candidate transformation T estimated from a model $\mathcal{M} \in \{\mathcal{M}_t, \mathcal{M}_s, \mathcal{M}_a, \mathcal{M}_p\}$ is suitable for describing the motion of the tracked object, a distance measure and a conditional probability are defined by using the position of the object from the previous frame $t-1$. We assume that the object movement follows the same distribution in two consecutive image frames. Let the normalized boundary of the tracked object be $\boldsymbol{\gamma} : [0,1] \mapsto \mathbf{R}^2$, and the normalized boundary of the tracked object under a candidate transformation be $T(\boldsymbol{\gamma})$. A distance measure is defined as the movement of the boundary under the transformation T:

$$\mathrm{dist}(T|\boldsymbol{\gamma}) = \int_0^1 ||\boldsymbol{\gamma}(t) - T(\boldsymbol{\gamma}(t))||dt. \tag{1}$$

When the boundary can be described by a polygon $\boldsymbol{p}_t = \{p_t^k\}_{k=1}^n$, only the distances moved by the points are considered:

$$\mathrm{dist}(T|\boldsymbol{p}_{t-1}) = \sum_{k=1}^{n} ||p_{t-1}^k - T(p_{t-1}^k)||. \tag{2}$$

A distribution that have been empirically proven to approximate the inter-frame movement is the exponential distribution (density function $\lambda e^{\lambda x}$). The parameter λ is estimated from the movements measured in previous frames. The probability of a candidate transformation T is the probability of a movement with greater

or equal magnitude. Given the previous object boundary and the decay rate λ this probability is:

$$\mathrm{P}(T|\lambda, p_{t-1}) = e^{-\lambda \operatorname{dist}(T|\boldsymbol{p}_{t-1})} \tag{3}$$

This way, transformations resulting in big movements are penalized, while transformations resulting in small movements are favored. In addition to the number of consensus points, this is the criterion used to select the correct transformation.

3.3 Criterion for Selecting a Transformation Model

A score is defined for choosing the best transformation and is computed for every transformation candidate T, which are estimated using a random model and a random choice of point correspondences:

$$\operatorname{score}(T) = \#(C) + \log_{10} \mathrm{P}(T|\lambda, \boldsymbol{p}^{t-1}) + \varepsilon n_{\min}, \tag{4}$$

where $\#(C)$ is the number of consensus points, and $n_{\min}$ is the minimum number of points needed to estimate the model correctly. The last term $\varepsilon n_{\min}$ is introduced to slightly favor a more complicated model. Otherwise, if the movement is small, both a simple and a complex model might have the same number of consensus points and approximately the same probability, resulting in the selection of a simple model. This would ignore the increased accuracy of the advanced model, and could lead to unnecessary error accumulation over time. Adding the last term hence enable, if all other terms are equal, the choice of a more advanced model. $\varepsilon = 0.1$ was used in our experiments.

The score is computed for every candidate transformation. The transformation T having the highest score is then chosen as the correct transformation model for the current video frame, after LS re-estimation over the consensus set. It is worth noting that the score in the RANSAC is $\operatorname{score}(T) = \#(C)$ with only one model. Table 1 summarizes the proposed algorithm.

4 Updating Point Feature Set

It is essential that a good feature set of the tracked object $\mathbf{F}_t^{\mathrm{obj}}$ is maintained and updated. A simple method is proposed here for updating the feature set of the tracked object, through dynamically adding and pruning feature points. To achieve this, a score S_t is assigned to each object feature point. All feature points are then sorted according to their score values. Only the top M feature points are used for matching the object. The score for each feature point is then updated based on the matching result and motion estimation:

$$S_t = \begin{cases} S_{t-1} + 2 & \text{matched, consensus point} \\ S_{t-1} - 1 & \text{matched, outlier} \\ S_{t-1} & \text{not matched} \end{cases} \tag{5}$$

Table 1. The RAMOSAC algorithm in pseudo-code

Input: Models $\mathcal{M}_i, i = 1, \ldots, m$, Point correspondences $(\boldsymbol{x}_k^{(t-1)}, \boldsymbol{x}_k'^{(t)})$,
$\boldsymbol{x}_k^{(t-1)} \in \mathbf{F}_{t-1}^{\text{obj}}$, $\boldsymbol{x}_k'^{(t)} \in \mathbf{F}_t'$, λ, $\boldsymbol{p}_{t-1}$
Parameters: $i_{\max} = 30$, $d_{\text{thresh}} = 3$

$s_{\text{best}} \leftarrow -\infty$
for $i \leftarrow 1 \ldots i_{\max}$ **do**
  Randomly pick $\mathcal{M}$ from $\mathcal{M}_1 \ldots \mathcal{M}_m$
  $n_{\min} \leftarrow$ number of points to estimate $\mathcal{M}$
  Randomly choose a subset of n_{min} index points
  Using $\mathcal{M}$, estimate T from this subset
  $C \leftarrow \{\}$
  foreach $(\boldsymbol{x}_k, \boldsymbol{x}_k')$ **do**
    if $||\boldsymbol{x}_k' - T(\boldsymbol{x}_k)||^2 < d_{thresh}$ **then** Add k to C
  end
  $s \leftarrow \#(C) + \log_{10} \mathrm{P}(T|\lambda, \boldsymbol{p}_{t-1}) + \varepsilon n_{\min}$
  if $s > s_{\text{best}}$ **then**
    $\mathcal{M}_{\text{best}} \leftarrow \mathcal{M}$
    $C_{\text{best}} \leftarrow C$
    $s_{\text{best}} \leftarrow s$
  end
end
Using $\mathcal{M}_{\text{best}}$, estimate T from C_{best}
return T

Initially, the score of a feature point is set to be the median of the feature points currently used for matching. In that way, all new feature points will be tested in the next frame without interfering with the important feature points that have the highest scores. For low-quality video with significant motion blur, this simple method was proven successful. It allows the inclusion of new features while maintaining stable feature points.

Pruning of feature points: In practice, only a small portion of the candidate points with high score are kept in the memory. The remaining feature points are pruned for maintaining a manageable size of feature list. Since these pruned feature points have low scores, they are unlikely to be used as the key feature points for tracking the target objects. Figure 2 shows the final score distribution of the 3568 features collected throughout the test video "Picasso", with $M = 100$.

Updating of feature points when two objects intersect or overlap: When multiple objects intersect or overlap, feature points located in the intersection need special care in order to be assigned to the correct object. This is solved by examining the matches within the intersection. The object having consensus points within the intersection area is considered the foreground object and any new features within that area are assigned to it. No other special treatment is required for tracking multiple objects. Figure 5 shows an example of tracking results with two moving objects (walking persons) using the proposed method.

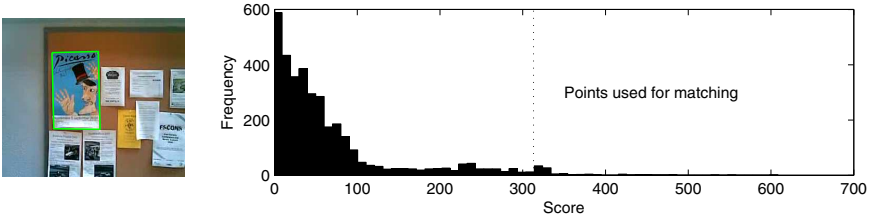

Fig. 2. Final score distribution for the "Picasso" video. The $M = 100$ highest scoring features were used for matching.

Fig. 3. RANSAC (red) compared to proposed method RAMOSAC (green) for frames #68–#70, #75–#77 of the "'Car" sequence. See also Fig. 6 for comparison. For some frames in this sequence, there is a single correct match with several outliers, making RANSAC estimation impossible.

Fig. 4. Tracking results from the proposed method RAMOSAC for the video "David" [9], showing matched points (green), outliers (red) and newly added points (yellow)

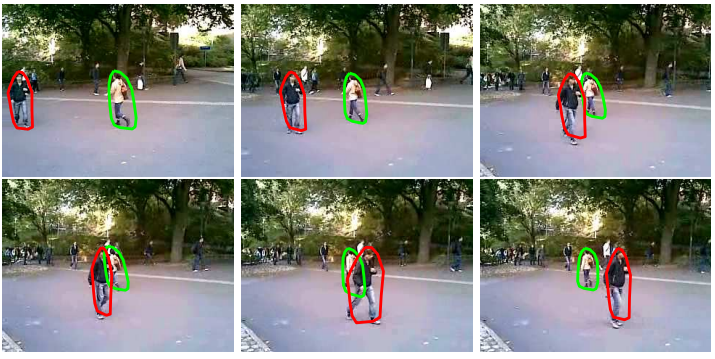

Fig. 5. Tracking two overlapping pedestrians (marked by red and green) using the proposed method

5 Experiments and Results

The proposed method RAMOSAC have been tested for a range of scenarios, including tracking rigid objects, deformable objects, objects with pose changes and multiple overlapping objects. The video used for our tests were recorded by using a cell phone camera with a resolution of 320×200 pixels. Three examples are included: In Fig. 3 we show an example of tracking a rigid license plate in video with a very high amount of motion blur, resulting in a low number of good matches. Results from the proposed method and from RANSAC are included for comparison. In the 2nd example, shown in the first row of Fig. 4, a face (with pose changes) was captured with a non-stationary camera. The 3rd example, shown in the 2nd row of Fig. 5, simultaneously tracks two walking persons (containing overlap). By observing the results from these videos in our tests, and from the results shown in these figures, one can see that the proposed method is robust for tracking moving objects with a range of complex scenarios.

The algorithm (implemented in MATLAB) runs in real-time on a modern desktop computer for 320×200 video if the faster SURF features are used. It should be noted that over 90% of the processing time is nevertheless spent calculating features. Therefore, any additional processing required by our algorithm is not an issue. Also, both the extraction of features and the estimation of the transformation is amenable to parallelization over multiple CPU cores.

All video files used in this paper are available for download at `http://www.maths.lth.se/matematiklth/personal/petter/video.php`

5.1 Performance Evaluation

To evaluate the performance, and compare the proposed RAMOSAC estimation with RANSAC estimation, the "ground truth" rectangle for each frame of the "Car" sequence (see Fig. 3) was manually marked. The Euclidean distance between the four corners of the tracked object (i.e. car license plate) and the ground truth

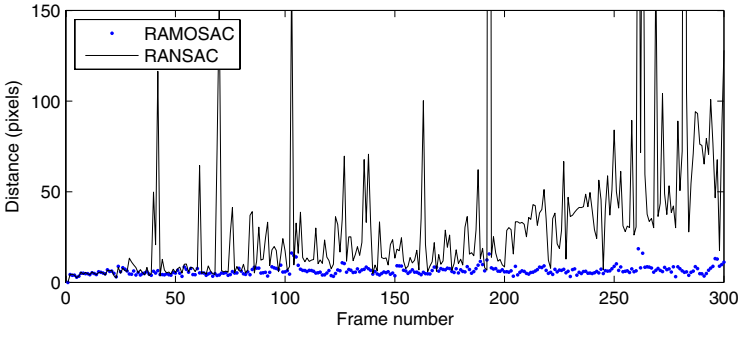

Fig. 6. Euclidean distance between the four corners of the tracked license plate and the ground truth license plate vs. frame numbers, for the "Car" video. Dotted blue line: the proposed RAMOSAC. Solid line: RANSAC.

was then calculated over all frames. Figure 6 shows the distance as a function of image frame for the "Car" sequence. In this comparison, RANSAC always used an affine transformation, whereas RAMOSAC chose from translation, similarity and an affine transformation. The increased robustness obtained from allowing models of lower complexity during difficult passages is clearly seen in Fig. 6.

6 Conclusion

Motion estimation based on RANSAC and (e.g.) an affine motion model requires that at least three correct point correspondences are available. This is not always the case. If less than the minimum number of correct correspondences are available, the resulting motion estimation will always be erroneous.

The proposed method, based on using multiple motion transformation models and finding the maximum number of consensus feature points, as well as a dynamic updating procedure for maintaining feature sets of tracked objects, has been tested for tracking moving objects in videos. Experiments have been conducted on tracking moving objects over a range of video scenarios, including rigid or deformable objects with pose changes, occlusions and two objects with intersect and overlap. Results have shown that the proposed method is capable of, and relatively robust in handling such scenarios.

The method has shown especially effective for tracking in low quality videos (e.g. captured by mobile phone, or videos with large motion blur) where motion estimation using RANSAC runs into some problems. We have shown that using multiple models of increasing complexity is more effective than RANSAC with the complex model only.

Acknowledgments

This project was sponsored by the Signal Processing Group at Chalmers University of Technology and in part by the European Research Council (GlobalVision

grant no. 209480), the Swedish Research Council (grant no. 2007-6476) and the Swedish Foundation for Strategic Research (SSF) through the programme Future Research Leaders.

References

1. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. Computer Vision and Image Understanding (CVIU) 110(3), 346–359 (2008)
2. Clarke, J.C., Zisserman, A.: Detection and tracking of independent motion. Image and Vision Computing 14, 565–572 (1996)
3. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24(6), 381–395 (1981)
4. Gee, A.H., Cipolla, R., Gee, A., Cipolla, R.: Fast visual tracking by temporal consensus. Image and Vision Computing 14, 105–114 (1996)
5. Grabner, M., Grabner, H., Bischof, H.: Learning features for tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2007, June 2007, pp. 1–8 (2007)
6. Li, L., Huang, W., Gu, I.Y.-H., Luo, R., Tian, Q.: An efficient sequential approach to tracking multiple objects through crowds for real-time intelligent cctv systems. IEEE Trans. on Systems, Man, and Cybernetics 38(5), 1254–1269 (2008)
7. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 20, 91–110 (2004)
8. Malik, S., Roth, G., McDonald, C.: Robust corner tracking for real-time augmented reality. In: VI 2002, p. 399 (2002)
9. Ross, D., Lim, J., Lin, R.-S., Yang, M.-H.: Incremental learning for robust visual tracking. International Journal of Computer Vision 77(1), 125–141 (2008)
10. Simon, G., Fitzgibbon, A.W., Zisserman, A.: Markerless tracking using planar structures in the scene. In: IEEE and ACM International Symposium on Augmented Reality (ISAR 2000). Proceedings (2000)
11. Skrypnyk, I., Lowe, D.G.: Scene modelling, recognition and tracking with invariant image features. In: ISMAR 2004, Washington, DC, USA, pp. 110–119. IEEE Comp. Society, Los Alamitos (2004)
12. Li, X.-R., Li, X.-M., Li, H.-L., Cao, M.-Y.: Rejecting outliers based on correspondence manifold. Acta Automatica Sinica (2008)