Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

Particle Filtering Using Multiple Cross-Correlations for Tracking Occluded Objects in Cluttered Scenes

Arie Nakhmani, and Allen Tannenbaum

Abstract— **This paper is concerned with the tracking of partially or entirely occluded objects in a video sequence. We propose certain modifications to the template matching approach, which seem to fit the type of tracking data being considered in the present note. Specifically, we will use a nonstandard particle filtering method via the following two steps: The first step employs the normalized cross-correlation function as the likelihood. The second step is to resample, and to fuse the results of multiple cross-correlations of different patches of the given object, in order to refine the likelihood for the particle filter. Experimental results show that the method is reliable for noisy measurements, and provides accurate results in cases of occlusion or heavy shadows.**

I. Introduction

TARGET tracking in video sequences is a very important problem in surveillance and security with applications ranging from homeland security to missile defense.

Our goal in this work is to track the target object defined by its template (for example, a snapshot from the first frame in a video sequence), and to construct the target's trajectory based only on this template. We are not interested in the present work in extracting the form or features of the target, but only in tracking a point within the given object, preferably close to its center of mass.

The proposed algorithm works in a variety of scenarios. The method deals naturally with clutter and noise in scenes, low contrast targets, and partial occlusions. The incorporation of a modified particle filter provides a smooth target trajectory, and ameliorates the problem of target occlusion.

We should note that the problem formulation is not new, and a large literature is available on this topic. We mention here only a few of the most relevant works for the approach taken in this paper. A broad survey on general tracking methods can be found in the book by Blackman and Popoli [1], and in the book by Bar-Shalom et al. [2]. A deep analysis of particle filters is provided in [3], where rigorous theory and applications of different kinds of particle filters are presented. Also, a powerful application of particle filters to image sequences (CONDENSATION algorithm) can be found in the paper by Blake and Isard [4]. More recent extensions incorporating particle filtering in visual tracking are available (e.g., [5], [6]).

The development of cross-correlation based trackers began in the sixties. Many image comparison algorithms were invented since then, among them normalized cross-correlation coefficient with fast computation algorithm, e.g. [7], statistical correlation [8], median correlation [9], correlation with multiple patches [10], and other template matching schemes. The idea of a particle filter that utilizes two measurement steps (including the correlation measurement) is presented in work on the direction of arrival of audio signals [11]. In our algorithm, we use a related idea for 2-D image signals. Also, by analogy with multiresolution analysis, we define and we use multiple cross-correlation coefficients for better tracking.

In general, the video tracking systems can be divided to four basic classes:

Template matching (e.g., [7]-[10]);
Feature / model based matching (e.g., [12]);
Contour matching (e.g., [13]);
Motion and structure based trackers (e.g., [14]).

A number of algorithms suppose a stationary camera, visible target contours, key features such as target color, or need a preliminary learning process or adaptation. Our algorithm neither assumes stationary camera, nor finds the contour, nor assumes color information. The algorithm provides the robust way of template matching without adaptation for targets without prominent features.

The remainder of this paper is organized as follows. Section 2 explains the template-matching problem. We briefly discuss the solution with the normalized cross-correlation coefficient function (NCC), and we define the concept of half-wave rectified multiple normalized cross-correlation (MNCC). We point out important differences between NCC and MNCC. In Section 3, we discuss the general problem of tracking with particle filters, and present the particle filter that using two measurement steps that are based on NCC and MNCC. In Section 4, we test our algorithm on four video sequences, which clearly illustrate its features. Finally, in Section 5, we summarize our research, and present the conclusions. We also discuss some key problems that still have to be solved, and propose the future directions for the research.

Manuscript received August 8, 2008.

A. Nakhmani is with the Department of Electrical Engineering, Technion - Israel Institute of Technology, Haifa 32000, Israel (phone: 972-(4)-8293285; e-mail: anry@tx.technion.ac.il).

A. Tannenbaum is with the Department of Electrical Engineering, Technion - Israel Institute of Technology, Haifa 32000, Israel, and Schools of Electrical and Computer and Biomedical Engineering, Georgia Institute of Technology Atlanta, Georgia 30332-0250 (e-mail: tannenba@ece.gatech.edu).

978-1-4244-3124-3/08/$25.00 ©2008 IEEE

II. Template Matching by Multiple Normalized Cross-Correlations

Let $I(m,n)$ denote the intensity value of the image (or the search region), and $P(i,j)$ denote the intensity value of the template patch. We assume that the size of I is $M_x \times M_y$, and the size of P is $N_x \times N_y$ (see Fig.1). Clearly, we assume that the size of I is greater than the size of P. It is known that the noisy version of the patch is placed somewhere in the image I. Our goal is to determine the most probable position of the patch in image I. The standard approach to this problem is to compute the coordinates of the maximum normalized cross-correlation coefficient (NCC) between the image and the template. These coordinates represent the position of the best match. The normalized cross-correlation coefficient is defined for any pixel (m,n) by:

$$NCC_P(m,n) = \frac{\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}(I(i+m-1,j+n-1)-\overline{I}(m,n))(P(i,j)-\overline{P})}{\sqrt{\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}(I(i+m-1,j+n-1)-\overline{I}(m,n))^2\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}(P(i,j)-\overline{P})^2}} \quad (1)$$

where the mean intensities $\overline{I}$ and $\overline{P}$ are defined by:

$$\overline{P} = \frac{1}{N_x N_y}\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}P(i,j), \quad (2)$$

$$\overline{I}(m,n) = \frac{1}{N_x N_y}\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}I(i+m-1,j+n-1), \quad (3)$$

$$\begin{aligned} m &= 1,2,\ldots,M_x - N_x + 1, \\ n &= 1,2,\ldots,M_y - N_y + 1. \end{aligned} \quad (4)$$

The values of $NCC(m,n)$ are between -1 and 1 (1 for perfect match, and 0 for "no correlation").

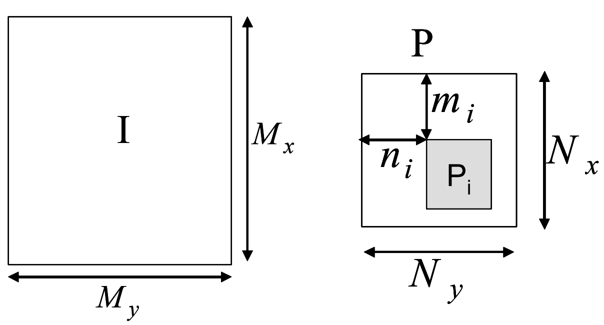

Figure 1: The dimensions of image I, the dimensions of template patch P, and the position of sub-part P_i in the template.

We are interested only in non-negative values of NCC; therefore we define the half-wave rectified cross-correlation as follows:

$$RNCC_P(m,n) = \begin{cases} NCC(m,n), & if\ NCC(m,n) \ge 0, \\ 0, & otherwise. \end{cases} \quad (5)$$

The position of the maximal value of RNCC can be used for tracking the desired object in image sequence. The presented technique is used in many practical applications, and has shown a robustness to noise and intensity variations [7]. The problem is that this technique may fail in the case of a partial occlusion of the desired object in the image I, or in case that the object is partially deformed (not rigid). In addition, the peak of normalized cross-correlation is blunt, and not always appropriate for accurate tracking (see Fig. 2a).

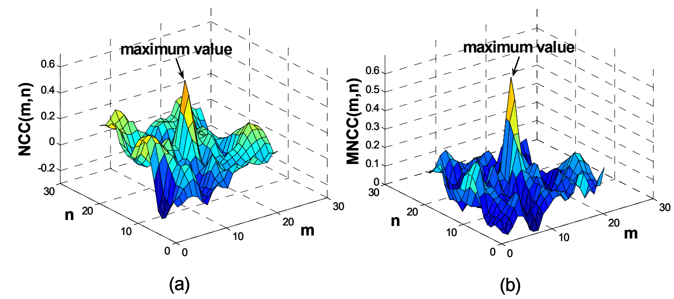

Figure 2: a. Example of normalized cross-correlation map
b. Example of multiple normalized cross-correlations map for the same image and template.

To overcome these problems we propose to use multiple normalized cross-correlation coefficients. The idea is to divide the template patch into K rectangular sub-regions (not necessarily disjoint), and to compute the RNCC map for every sub-region. Then, all the maps are fused to a single map, according to the offsets of the sub-regions in the template patch, namely:

$$MNCC_P(m,n) = \frac{1}{K}\sum_{i=1}^{K} RNCC_{P_i}(m+m_i, n+n_i) \quad (6)$$

where
P_i = the sub-region #i;
(m,n) is defined in (4);
(m_i, n_i) denotes the offset of i-th sub-region according to the top-left corner of the template patch $(0 \le m_i \le N_x - 1,\ 0 \le n_i \le N_y - 1)$.

The position of maximal value of MNCC function is equivalent to the position of the best match. The problem of partial occlusion is avoided, because we are constructing the correlation map by computing cross-correlations of different parts of desired object, and the maximum value is achieved where the most parts are matched. The occluded parts give low RNCC coefficients, therefore their influence on MNCC is weak, and they can be ignored. The peaks of multiple cross-correlations are sharp (see Fig. 2b) that makes this technique less (than NCC) appropriate for general tracking, but more appropriate for fine-tuning after the search region in the image *I* is narrow and well-defined.

In the next section, we will combine the advantages of the RNCC and MNCC techniques in a particle filtering framework with low number of state parameters, which makes it appropriate for real-time applications.

III. Robust Tracking with Particle Filters

A. Particle filtering

In this section, we present a modified algorithm for particle filtering, which we believe is quite useful for visual tracking in our framework. Our approach may be divided into two steps. The first step incorporates the advantages of normalized cross-correlation for template matching. The second step, after resampling, uses MNCC as the likelihood for determining the object's position. This algorithm showed the robust tracking results in the case of translational moving object in clutter and with possible partial and entire occlusions, as we will see in the experimental section below. We refer the reader to reference [3] for the complete background on particle filtering.

In general, the goal of particle filter is to estimate the sequence of hidden state parameters X_k, based only on the observed data Z_k. These estimates follow from the posterior distribution $P(X_k|Z_0,Z_1,\ldots,Z_k)$. It is assumed that the state and the observations are first order Markov processes, and each Z_k depends only on X_k. The particle filter estimates the $P(X_k|Z_0,Z_1,\ldots,Z_k)$ distribution, and it does not require any linearity or Gaussian assumptions on the model. The particle filter will generate a set of N samples that approximate the filtering distribution. For the k-th frame, we denote the state vector by $X_k=(x_1,x_2,\ldots)$. For example, (x_1,x_2) can be the top-left corner coordinates of the desired object in the frame. The state estimate is recursively obtained as follows:

$$\begin{aligned} &P(X_k \mid Z_0, Z_1, \ldots Z_k) \propto \\ &\qquad P(X_{k-1} \mid Z_0, Z_1, \ldots Z_{k-1}) \cdot \pi_k \cdot P(X_k \mid X_{k-1}) \end{aligned} \tag{7}$$

where

$$\pi_k = P(Z_k \mid X_k) \approx \begin{cases} P^{(RNCC)}(Z_k \mid X_k) \propto RNCC, \text{for 1st measure} \\ P^{(MNCC)}(Z_k \mid X_k) \propto MNCC, \text{for 2nd measure} \end{cases} \tag{8}$$

The prediction step that corresponds to the distribution $P(X_k \mid X_{k-1})$ is governed by system state dynamical equations. For example, if state time evolution is assumed to be smoothly changing, and there is no additional information about the object dynamics, then the simplest model given by

$$X_k = X_{k-1} + v_k, \quad v_k \sim N(0,\Sigma) \tag{9}$$

is many times appropriate. The mean of X_k over all the particles is approximately the actual value of X_k.

B. The Algorithm

The state estimation is carried out by updating weighted particles according to (7). In the first part of the algorithm we employ a bootstrap filter which uses the system model and $P^{(NCC)}(Z_k \mid X_k)$ as the likelihood, due to its blunt peaks. For MNCC, the likely result is a function $P^{(MNCC)}(Z_k \mid X_k)$ with many sharp peaks in it. Therefore, we can use this distribution only in the second part to localize the desired object. In this step only the weights of particles are updated according to $P^{(MNCC)}(Z_k \mid X_k)$.

Initialization:

The N particles $X_0^{(n)}, (n=1,\ldots,N)$ are drawn from the uniform distribution.

For every video frame (k-th frame), we perform the following steps.

STEP 1:

Using the particles from previous frame, predict the new state by sampling from:

$$\tilde{X}_k^{(n)} \sim P(X_k \mid X_{k-1}^{(n)}). \tag{10}$$

STEP 2:

Measure and weight the new position in terms of the measured features Z_k:

$$\begin{aligned} \pi_k^{(n)} &= P^{(RNCC)}(Z_k \mid \tilde{X}_k^{(n)}), \\ \tilde{w}_k^{(n)} &\propto w_{k-1}^{(n)} \pi_k^{(n)}. \end{aligned} \tag{11}$$

STEP 3:

Resample the particles $\tilde{X}_k^{(n)}, (n=1,\ldots,N)$ according to the weight $\tilde{w}_k^{(n)}, (n=1,\ldots,N)$. Denote the resampled particles by $X_k^{(n)}$.

STEP 4:

Update the weights of the $X_k^{(n)}$ particles as:

$$w_k^{(n)} \propto P^{(MNCC)}(Z_k \mid X_k^{(n)}) \tag{12}$$

STEP 5:

Compute the state estimate from:

$$\hat{X}_k \approx \frac{1}{N} \sum_{n=1}^{N} X_k^{(n)}, \tag{13}$$

and repeat the steps (1-5) for the next video frame.

The result of this algorithm is the estimated state $\hat{X}$, that includes the information about the position of the desired object in every video frame.

IV. Experiments and Discussion

We tested the proposed algorithm in various situations, including highly cluttered exterior scenes with shadows and occlusions with a high rate of success. A single template was used for every video. To see more clearly the results of tracking, only the relevant parts of the full scene images are shown in Figures 3, 4, 6 and 7.

- The following examples include gray level sequences, recorded at 25 frames per second, with resolution of 240×320 pixels.
- We chose the simplest dynamics model (9) for the first three sequences, and constant velocity model [1 p.203] for the last sequence.
- The target is selected manually in the first video frame.
- The template size is constant.
- For MNCC calculation, the template is divided to 3×2 non-overlapping quadrants.

A. Sequence 1: Woman walking through shadow

In the first sequence, we want to track a woman that walks on the alleyway through the shadow from the tree (see Fig. 3). This scenario represents the difficulty in tracking under outdoor conditions, where the illumination of the target is changing abruptly. In addition, surrounding objects can mislead the tracker, especially when the woman walks through the shadow, and she is hardly visible.

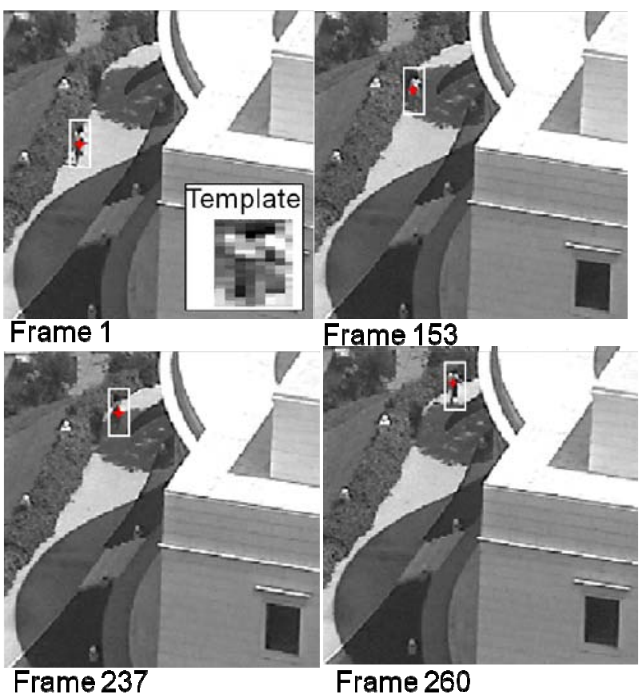

Figure 3: The video sequence where the woman walks through the shadow.

In this sequence, the proposed algorithm robustly tracked the target with 60 particles, while the NCC based tracker lost its track after 100 frames.

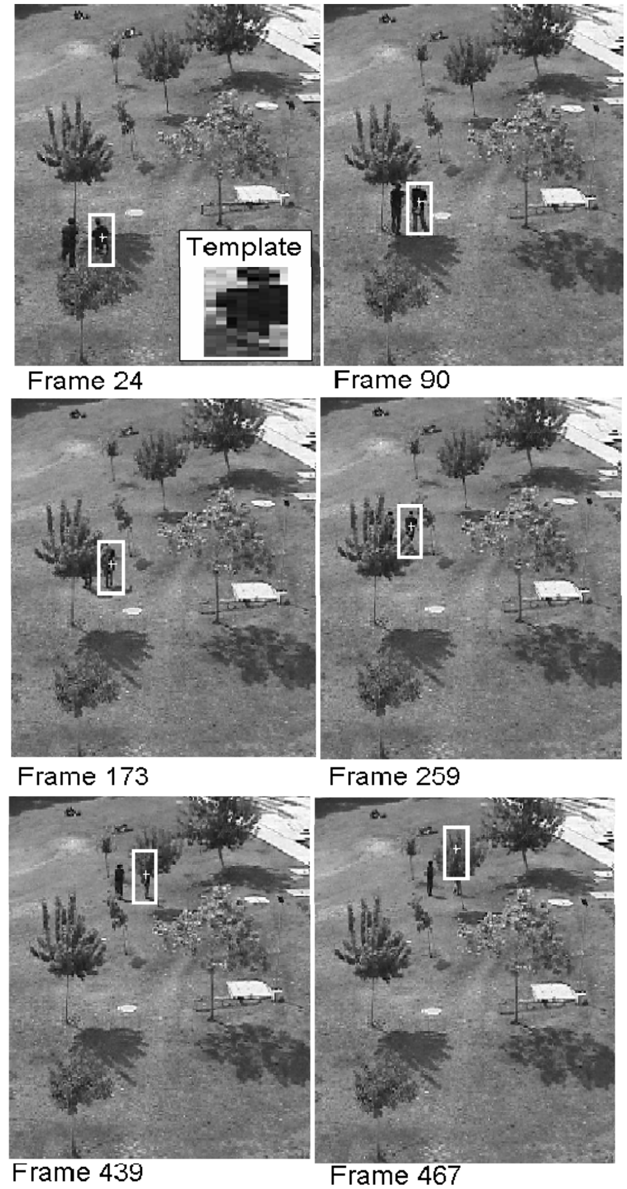

Figure 4: The video sequence with two men walking together. Both of them partially (or entirely) occluded from time to time.

B. Sequence 2: Two people walking together with partial occlusions

The second sequence indicates two people walking together. The aim here is to follow the person from the right side. The camera is unsteady, and the two people look alike. Many moving parts of the picture (e.g., the moving leaves on the trees) can distract a tracker, but our tracker is robust to such changes. Also, the tracker manages to overcome the problem of partial occlusion (see Fig. 4).

NCC based trackers (with and without the standard PF) will follow the target person only until the frame 173, where the partial occlusion occurs. In this case, our multiple correlations based algorithm follows the object without any problem with 50 particles. In the frame 439, we see a partial occlusion of the target. The second unoccluded person would mislead the standard cross-correlation, but the MNCC still provides the right measurement for the particle filter. In frame 467, the target person is completely occluded by the tree, and now only the particle filter is able to compensate for the full occlusion, and to keep the right track (at least for a short time).

To verify the robustness of our algorithm, we want to examine the number of particles that is sufficient for successful tracking. It is clear, that if we are willing use more particles, then one should expect better tracking results and runs that are more successful. On the other hand, this will cause longer computation times. We consider a tracking run a success for a given video sequence, if for every frame of the video, the tracker pointing on the desired object (compared to manually tracked position).

The Fig. 5 shows the number of successful runs (from 100) as function of number of particles. From this graph, we conclude that with our proposed algorithm, 60 particles are sufficient to get very reasonable tracking results. The graph also suggests that although the second sequence is longer, it is easier to track.

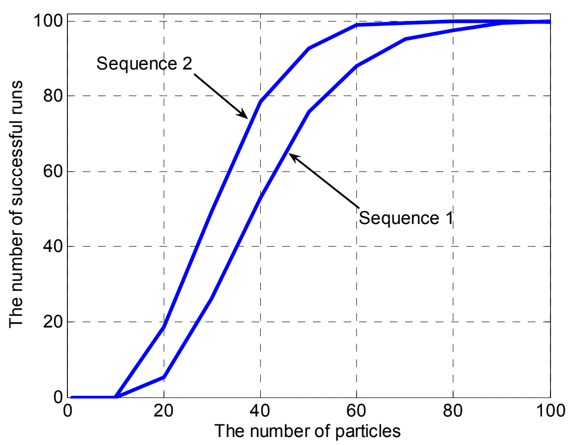

Figure 5: The approximate number of successful random runs (from 100) as function of the number of particles for two tested sequences.

C. Sequence 3: A car driving on the road captured by non-stationary camera

In this sequence, we want to track a white car driving on the road. The results of tracking with 60 particles are shown in Fig. 6. In the frame 59 the car is significantly rotated (relative to its initial template position), and our tracker temporarily loses track. Nevertheless, with a smaller rotation angle, such as in frame 200, our tracker finds the right position of the car. The frame 445 shows three similar cars with some zoom. The bottom car is the tracked one. Our tracker still gives the right result in the presence of closely placed similar cars.

D. Sequence 4: A maneuvering vehicle with full occlusion

In this sequence, the vehicle is maneuvering, and trying to hide, and the camera moving following him. The full occlusion occurs in the frames 81-103.

Despite the changes in scaling and appearance, our tracker

Figure 6: The video sequence with a white car driving on the road, captured by a non-stationary camera. This example includes a moving background, car rotations, and scaling.

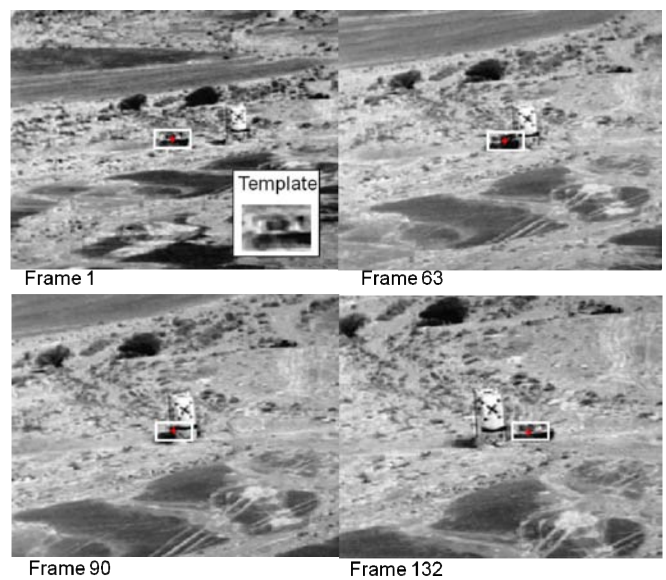

Figure 7: The video sequence with a maneuvering vehicle, captured by a non-stationary camera. This example includes a moving background, vehicle appearance changes, scaling, and entire occlusion.

with 80 particles follows the target even when it is in the hiding-place (see Fig. 7 frame 90).

E. Comparison of correlation-based algorithms to our approach

We tested the correlation-based algorithms [7-10] with each sequence, and most of them failed to track the selected targets. The algorithms [7-9] based on NCC cannot follow the occluded target and often provide non-smooth trajectories. The algorithm [9] significantly increases the robustness of correlation tracking, and can be later incorporated in our approach. The approach of Guo and

Dyer [10] overcomes the problem of partial occlusions, but cannot solve the problem with full occlusion, and the resulting trajectories are not always smooth, in contrast to our approach.

We recall that for all sequences we used simple target dynamics model and a constant template. We assumed that no additional information is given about the target, besides the template. With learned higher order models, and smoothly changing adaptive template we expect to get even better results with the same algorithm.

V. CONCLUSION

In this paper, we presented an algorithm for tracking in video sequences of occluded objects without the need for adaptation and learning mechanisms. With rather small number of particles and low number of states (compared, for example, to the CONDENSATION algorithm), we achieve a robust tracking results with many complicated and cluttered real world video sequences, including sequences with camera motion.

The combination of the particle filter with two types of correlation trackers makes it possible to get smooth target trajectories. The algorithm can cope with translations, and moderate deformations of the tracked object, when the deformations affect only a small portion of pixels in any template patch used by MNCC. The algorithm is appropriate also for small targets with low contrast. The proposed approach is time efficient, and should be suitable for real-time applications. The disadvantage of our methodology is that it is not capable of tracking targets subject to large rotations and a large scaling (zoom). The next step in our research is to add scaling and rotation states to the particle filter definition, and to choose good dynamic models for scaling and rotation. With these modifications, we expect to get scale and rotation invariant tracking. In addition, other types of correlation measures should be tested. Finally, in the future, the algorithm should be extended for multiple target tracking.

VI. REFERENCES

[1] S. Blackman, and R. Popoli. Design and Analysis of Modern Tracking Systems. Artech House Publishers, 1999.

[2] Y. Bar-Shalom, X.-Rong Li, and T. Kirubarajan. Estimation with Applications to Tracking and Navigation. John Wiley & Sons, Inc., 2001.

[3] A. Doucet, N. de Freitas, and N. Gordon. Sequential Monte Carlo methods in practice. Springer, 2001.

[4] A. Blake, and M. Isard. CONDENSATION -- conditional density propagation for visual tracking. International Journal of Computer Vision, Vol. 29(1), pp.5-28, 1998.

[5] Y. Li, H. Ai, T. Yamashita, S. Lao, and M. Kawade, "Tracking in Low Frame Rate Video: A Cascade Particle Filter with Discriminative Observers of Different Lifespans," Proc. of Computer Vision and Pattern Recognition, 2007.

[6] P. Brasnett, L. Mihaylova, D. Bull, and N. Canagarajah, "Sequential Monte Carlo tracking by fusing multiple cues in video sequences," Image and Vision Computing, 25, (8), pp. 1217-1227, 2007.

[7] J.P. Lewis. Fast Template Matching. Vision Interface, Quebec, Canada, pp. 120-123, 1995.

[8] W.K. Pratt. Correlation Techniques of Image Registration. IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-10, pp.353-358, May 1974.

[9] M. Trujillo, and E. Izquierdo. A Robust Correlation Measure for Correspondence Estimation. 3DPVT, pp. 155-162, 2004.

[10] G. Guo, and C.R. Dyer. Patch-based Image Correlation with Rapid Filtering. CVPR, pp. 1-6, 2007.

[11] M. Mizumachi, N. Ikoma, and K. Niyada. DOA Estimation Based on Cross-Correlation by Two-Step Particle Filtering. EUSIPCO, 2006.

[12] M. Krinidis, N. Nikolaidis, and I. Pitas. Feature-Based Tracking Using 3D Physics-Based Deformable Surfaces. EUSIPCO, 2005.

[13] Y. Rathi, N. Viswani, and A. Tannenbaum. A generic framework for tracking using particle filter with dynamic shape prior. IEEE Transactions on Image Processing, Vol. 16, pp. 1370-1382, 2007.

[14] T. Vieville, and Quang-Tuan Luong**.** Computing motion and structure in image sequences without calibration. Pattern Recognition, 1994. Proceedings of the 12th IAPR International Conference on Computer Vision & Image Processing, Vol. 1, pp. 420 - 425, 1994.