Published in final edited form as:

SIAM J Imaging Sci. ; 4(1): 220–242. doi:10.1137/090779280.

Particle Filtering with Region-based Matching for Tracking of Partially Occluded and Scaled Targets*

Arie Nakhmani[†] and Allen Tannenbaum[‡]

[†]Electrical Engineering Department, Technion, Haifa, 32000, Israel (anry@tx.technion.ac.il).

[‡]Electrical Engineering Department, Technion, Haifa, 32000, Israel, and Schools of Electrical and Computer and Biomedical Engineering, Georgia Institute of Technology, Atlanta, GA 30332 (allen@ee.technion.ac.il).

Abstract

Visual tracking of arbitrary targets in clutter is important for a wide range of military and civilian applications. We propose a general framework for the tracking of scaled and partially occluded targets, which do not necessarily have prominent features. The algorithm proposed in the present paper utilizes a modified normalized cross-correlation as the likelihood for a particle filter. The algorithm divides the template, selected by the user in the first video frame, into numerous patches. The matching process of these patches by particle filtering allows one to handle the target's occlusions and scaling. Experimental results with fixed rectangular templates show that the method is reliable for videos with nonstationary, noisy, and cluttered background, and provides accurate trajectories in cases of target translation, scaling, and occlusion.

Keywords

visual tracking; particle filtering; normalized cross-correlation; occlusions; scale changes

1. Introduction

Visual tracking is a fast growing research area within the field of computer vision. Some form of tracking is an inseparable part of many dynamic segmentation and object recognition algorithms. Moreover, it is used in a variety of applications ranging from security and surveillance to guided surgery and space exploration.

1.1. Problem description

In this paper, we consider the tracking of arbitrary targets in cluttered real-world video sequences. The target is selected in the first frame of the video (a bounding box defined by top-left and bottom-right points). Then the template, extracted from this selection, is used to construct the two-dimensional trajectory of the target's centroid along the video and to find the target's bounding box for each frame. The reference template is the only available information on the target. Some of the properties of the videos considered are as follows:

*This work was supported in part by grants from NSF, AFOSR, and ARO, by a grant from NIH (NAC P41 RR-13218) through Brigham and Women's Hospital, as well as by the Ollendorff Minerva Center for Vision and Image Sciences. This work is part of the National Alliance for Medical Image Computing (NAMIC), funded by the National Institutes of Health through the NIH Roadmap for Medical Research, grant U54 EB005149. Information on the National Centers for Biomedical Computing can be obtained from <http://nihroadmap.nih.gov/bioinformatics>. <http://www.siam.org/journals/siims/4-1/77928.html>

- Since most videos are recorded in gray-level format, color information is not used by our algorithm.
- The template, extracted from the first frame, includes sufficient information for further tracking; that is, the entire target is visible in the first frame.
- The target is undergoing translation, scaling, and occlusions. In principle, in-plane rotations can be handled by adding the state of rotation angle and estimating the rotation by the particle filter. This would demand more particles and longer computational time and is not considered in the present work.
- The target has no major out-of-plane rotations (self-turning).
- The target does not undergo considerable nonrigid transformations, and the reference template adequately represents the target for a majority of video frames. This means that the correlation between the appropriately scaled template and the target in a sufficient number of video frames should be higher than the correlation to the background. This property is limiting, but it is necessary for any correlation-based tracker. Many top-view videos satisfy this property and are the good examples of videos considered here.
- The camera is not necessarily stationary, but the camera motion is smooth.
- The target motion is smooth; i.e., there are no abrupt changes between the consecutive video frames.

Although the problem has a simple formulation, visual tracking even with the above assumptions remains a challenging task. The main reasons for this are clutter, noisy measurements, target deformations, and occlusions. Robust target tracking in the presence of noise and uncertainty is the goal of any realistic visual tracker. Accordingly, in this paper, we treat the problem of tracking arbitrary scaled targets in the presence of noise and occlusions.

The basic idea of the proposed solution is to use a time-proven technique for template matching, namely, normalized cross-correlation, with the appropriately scaled template, while the scale is estimated by a particle filter. Moreover, for better target localization and for solving the problem of occlusions, we propose dividing the reference template into multiple patches and combining the modified normalized cross-correlations obtained from all those patches into a likelihood map, which is used by a particle filter for estimation of the target's location. The decisions on the location of patches (inside the template) and their size are made by a second embedded particle filter, which chooses the patches with low probability of being occluded.

The main novelty of our approach is in the concept of avoiding an explicit occlusion handling procedure and solving the problem approximately through particle filtering. Also, we propose a novel scale-invariant template matching scheme, which is more robust to occlusions than popular normalized cross-correlation.

In the next subsection we provide the necessary background and briefly describe the connection of our research to prior work.

1.2. Prior related work

Visual tracking is a broad research area, with a wide variety of visual tracking techniques proposed in the literature. Thus we do not attempt to give a comprehensive literature survey, but instead concentrate on several background works and papers most relevant to the

development of the present research. We indicate how our approach extends prior techniques.

A good introduction to some of the various methods used in visual tracking can be found in the survey paper by Yilmaz, Javed, and Shah [45]. A general introduction for the application of estimation theory to the system's state tracking can be found in Bar-Shalom and Fortmann [5] and Blackman and Popoli [7]. These books describe dynamical system modeling (including the constant velocity model [5, pp. 82–84] that we use in our work) and earlier statistical techniques of state estimation. The use of particle filters in state estimation began with the pioneering contribution of Gordon, Salmond, and Smith [13]. The theory of Monte Carlo methods and particularly mathematical aspects and applications of particle filtering are explored in Doucet, de Freitas, and Gordon [12] and Ristic, Arulampalam, and Gordon [35]. Isard and Blake proposed the CONDENSATION algorithm [16], which is the first application of particle filtering used in conjunction with deformable contours for the visual tracking problem. A number of recent papers indicate the success of the probabilistic particle filtering framework for visual tracking (e.g., [25, 27, 41]). In our work, we use a modified particle filter [3] based on the CONDENSATION algorithm. Our algorithm proposes a natural way to overcome the problem of scaling and occlusion by introducing particle scaling and modifying template matching.

Most visual tracking algorithms can be divided roughly into two groups: parametric and nonparametric. The former suppose that the class of allowable target transformations is known in advance. Such a class can consist of translations, scalings, rotations, or any other template transform with known structure. The latter group does not suppose a rigid transformation but uses some properties of the target (e.g., centroid tracker [7]). The nonparametric algorithms often provide the target's segmentation along with the tracking. Recently Rath, Vaswani, and Tannenbaum [34] also proposed the fusion of parametric and nonparametric approaches by model-based segmentation and tracking. Our algorithm belongs to a parametric group of trackers.

For the parametric trackers, the measurement that is made on each video frame can also be divided into two groups: feature-based and template-based measurements. The former class tries to match the distinguishable target features in each frame and to compute the appropriate transformation. For template-based measurements, the selected template is compared to the possible location of a target in each frame. The choice of template-based measurements in our work is dictated by the absence of prominent features in the considered targets. We have chosen a correlation-based similarity measure, but in future research other measures can be explored within the same framework, e.g., the normalized sum of squared distances proposed by Kolmogorov et al. [19] or Image Euclidean Distance (IMED) proposed by Wang, Zhang, and Feng [42]. The modified correlation is not a consistently constructed likelihood (as proposed in Sullivan et al. [38]), but a heuristic representative for the likelihood. In further research, it may be possible to use the exemplar-based probabilistic paradigm proposed by Toyama and Blake [39] as a more rigorous way to transform a correlation-based measure into a genuine likelihood, by using the noise model learned from the data.

The techniques for region matching by cross-correlation began to develop in the late 1960s [2], with some improvements and modifications that are still underway today (e.g., [6, 40, 44, 46]). These techniques are based on a similarity measure between the image and the reference template. Nineteen popular image similarity measures, including the normalized cross-correlation (NCC) that we have used in our work, were explored and compared by Aschwandt and Guggenbuhl [4]. Ooi and Rao [32] tested the robustness of NCC to illumination, scale, and viewpoint changes. The results in [4] show that NCC is an

appropriate similarity measure for the targets considered in the present work. Lewis [21] developed a fast computation method for NCC and suggested that NCC is a viable choice for many applications. Trujillo and Izquierdo [40] suggested replacing the arithmetic mean in the NCC formulation with the more statistically robust median, scale, and viewpoint changes. The results in [4] show that NCC is an appropriate and provided examples where this replacement can indeed improve the matching. We can incorporate the change proposed in [40] into our framework in order to improve tracking. Also, the correlation with scaled templates has been studied in several works [6, 8, 46].

The idea of dividing the template into patches is not new. Different template partitioning schemes, which consider the problem of image matching under partial occlusion, and methods for combining the votes of each patch were proposed by Guo and Dyer [15], Adam, Rivlin, and Shimshoni [1], Mitani and Saji [28], and Jurie and Dhome [17]. Some of these works use color information (e.g., [28]) or histograms (e.g., [1]), and others need an optimization or learning process (e.g., [17]) before the tracking can be performed. Nguyen and Smeulders [31] proposed an original tracking algorithm which overcomes partial occlusions by smoothing appearance features by Kalman filters, one for each pixel. Another interesting algorithm for occlusion detection was proposed by Wang, Han, and Yan [43], which is based on explicitly segmenting the locally distributed scores in each sliding window. In contrast our algorithm uses an implicit occlusion detection method, without offline learning.

The incorporation of traditional template matching into the particle filtering framework was proposed by Zhou, Chellappa, and Moghaddam [47], Mei, Zhou, and Porikli [27], and Shao, Zhou, and Chellappa [37]. These methods use highly adaptive templates and provide nice examples of tracking scaled targets (unlike our method, where the template is constant, or close to constant). Mei, Zhou, and Porikli [27] proposed incorporating robust template matching based on IMED [42] and an incremental subspace update for advanced online foreground modeling. Zhou, Chellappa, and Moghaddam [47] also proposed using an adaptive motion model with an adaptive number of particles. Although adaptation can sometimes significantly improve the tracking process, there are well-known difficulties associated with it. Template drift, where the target is not centered on the template due to erroneous subpixel updating, and partial occlusion inclusion in the template updating are examples of such difficulties (e.g., see [26]).

Some papers have described algorithms for computing the optimal particle number and allocation (e.g., [33]), but we use a fixed number of particles in order to simplify our algorithm and to reduce the computational load. Although many sophisticated methods for template update have been proposed recently (e.g., [14, 26]), we employ the simple updating scheme used by Wong [44]. In this scheme, a first order α tracker is used (exponential smoothing of the template), where the parameter α is proportional to the NCC between the the previous template and the subregion at the measured location of the target. To avoid problems of drift, we have used a constant template ($\alpha = 1$) or a very slowly updated template ($\alpha > 0.95$). Another important approach to appearance changes of the target was proposed by Lim et al. [22] and Ross et al. [36]. They presented the concept of incremental learning to reflect the changes in a target and to aid tracking. In contrast our tracker uses a constant reference template or a very slowly varying template. Furthermore, Kawamoto [18] recently considered the tracking of affine target motions with a particle filter. This method demands thousands of particles, and the problem of occlusion is not discussed.

In this work, we suppose that the only parameters that need be estimated are the location, velocity, and scaling of the target, and thus we can use an adapted template matching scheme (based on modified NCC) in our framework with a very small set of estimated

parameters i.e., a low state order). As a consequence, only a small number of particles is needed for robust tracking. This way we avoid the problem known as “the curse of dimensionality” [11]. To solve the problem of degeneracy of particles, where all but one particle have a very small weight, we use the measure for effective sample size proposed by Kong, Liu, and Wong [20] and resampling. There is an additional problem that can be considered based on the effective sample size. If a full occlusion occurs, most particles get similar weights, and thus the effective sample size is large. In this case, there is no reason to resample them and to move to other regions.

We should also note that our tracker can be improved by the incorporation of the meanshift algorithm [9, 10] in the prediction and measurement stage. Moreover, it is possible to use the exclusion principle and partitioned sampling, as proposed by MacCormick and Blake [23] and MacCormick and Isard [24], to extend our tracker to the case of multiple similar looking targets. However, this would increase the computational burden and thus is not discussed in the present paper. Some preliminary versions of certain parts of the work described here have appeared previously in [29, 30].

We now summarize the remainder of this paper. In section 2, we recall for the convenience of the reader some of the definitions proposed in [29]. We propose a measurement that considers partial occlusions and may be used as the likelihood for a particle filter. In section 3, we propose a general framework for visual tracking based on particle filtering. We describe the stages of prediction, measurement, estimation, and update of the reference template. In section 4, we summarize the overall algorithm and discuss practical issues of initialization and the choice of parameters. In section 5, we present some experimental results for four challenging real-world scenarios which elucidate the features of our tracking methodology. We compare our results to the results produced by the CONDENSATION and mean-shift algorithms, and by the recent algorithm proposed by Mei, Zhou, and Porikli [27]. Finally in section 6, we summarize the advantages and disadvantages of our approach and propose several directions for future research.

2. Multiple scaled normalized cross-correlations

A major part of any visual tracker is measurement: if the measurements are not adequate, it is impossible to track the target. Additional tracker components (prediction, estimation, and template update) are intended to eliminate problems associated with the measurements (false alarms and missing detections). The measurement computes the similarity between the reference template and the candidate for the location of the desired target. We begin this section with a simple and popular similarity measure, *zero mean normalized cross-correlation (NCC)*, and propose certain modifications that try to overcome the problems of weak localization, partial occlusions, and scaling.

2.1. Normalized cross-correlation

Let $I(m, n)$ denote the intensity value of the image (or the search region), and let $T(i, j)$ denote the intensity value of the reference template. The size of I is $M_x \times M_y$, and the size of T is $N_x \times N_y$ (see Figure 1).

To determine the most probable location of the deformed reference template in the image I , one must compute the coordinates of the maximum NCC coefficient between the image and the template. These coordinates represent the position of the best match. The NCC coefficient is defined for any pixel (m, n) by

$$NCC_T(m, n) = \frac{COV(I, T)}{\sigma_I \sigma_T}, \quad (2.1)$$

where

$$\begin{aligned} COV(I, T) &= \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \left(I(i+m-1, j+n-1) - \bar{I}(m, n) \right) \left(T(i, j) - \bar{T} \right), \\ \sigma_I &= \sqrt{\sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \left(I(i+m-1, j+n-1) - \bar{I}(m, n) \right)^2}, \\ \sigma_T &= \sqrt{\sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \left(T(i, j) - \bar{T} \right)^2}, \\ \bar{I}(m, n) &= \frac{1}{N_x N_y} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} I(i+m-1, j+n-1), \\ \bar{T} &= \frac{1}{N_x N_y} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} T(i, j), \\ m &= 1, 2, \dots, M_x - N_x + 1, \\ n &= 1, 2, \dots, M_y - N_y + 1. \end{aligned} \quad (2.2)$$

Remarks—

- The values of $NCC(m, n)$ are between -1 and 1 (1 for perfect match, and 0 for “no correlation”).
- Clearly, we assume that the size of I is not smaller than the size of T .

The presented technique is used in many practical applications and has shown robustness to noise and intensity variations [4]. Nevertheless, the reference template cannot be compared to a target of different size, and so NCC is inappropriate for tracking scaled targets. The straightforward solution to this problem is to find the location where NCC is maximal, where the correlation is measured between the image I and an appropriately scaled reference template.

2.2. Scaled normalized cross-correlation

Suppose that target scaling $s > 0$ is known. The NCC approach with the scaled template $\tilde{T}$ is called *scaled normalized cross-correlation (SNCC)* and is defined as follows:

$$SNCC_{\tilde{T}}(m, n, s) = \frac{COV(I, \tilde{T})}{\sigma_I \sigma_{\tilde{T}}}, \quad (2.3)$$

where

$$\begin{aligned}
COV(I, \tilde{T}) &= \sum_{i=1}^{\lfloor sN_x \rfloor} \sum_{j=1}^{\lfloor sN_y \rfloor} \left(I(i+m-1, j+n-1) - \bar{I}(m, n) \right) \left(\tilde{T}(i, j) - \bar{T} \right), \\
[\alpha] &\text{ is closest to } \alpha \text{ integer,} \\
\tilde{T}(i, j) &= T \left(\left\lceil \frac{\frac{N_x-1}{\lfloor sN_x \rfloor} (i-1)}{\frac{N_y-1}{\lfloor sN_y \rfloor} (j-1)} \right\rceil + 1 \right), \\
\sigma_I &= \sqrt{\sum_{i=1}^{\lfloor sN_x \rfloor} \sum_{j=1}^{\lfloor sN_y \rfloor} \left(I(i+m-1, j+n-1) - \bar{I}(m, n) \right)^2}, \\
\sigma_{\tilde{T}} &= \sqrt{\sum_{i=1}^{\lfloor sN_x \rfloor} \sum_{j=1}^{\lfloor sN_y \rfloor} \left(\tilde{T}(i, j) - \bar{T} \right)^2}, \\
\bar{I}(m, n) &= \frac{1}{\lfloor sN_x \rfloor \lfloor sN_y \rfloor} \sum_{i=1}^{\lfloor sN_x \rfloor} \sum_{j=1}^{\lfloor sN_y \rfloor} I(i+m-1, j+n-1), \\
\bar{T} &= \frac{1}{\lfloor sN_x \rfloor \lfloor sN_y \rfloor} \sum_{i=1}^{\lfloor sN_x \rfloor} \sum_{j=1}^{\lfloor sN_y \rfloor} \tilde{T}(i, j), \\
m &= 1, 2, \dots, M_x - \lfloor sN_x \rfloor + 1, \\
n &= 1, 2, \dots, M_y - \lfloor sN_y \rfloor + 1.
\end{aligned} \tag{2.4}$$

We are interested only in nonnegative correlation; therefore we define the half-wave rectified scaled normalized cross-correlation (RSNCC) as follows:

$$RSNCC_{\tilde{T}}(m, n, s) = \begin{cases} SNCC_{\tilde{T}}(m, n, s) & \text{if } SNCC_{\tilde{T}}(m, n, s) \geq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{2.5}$$

The maximum of RSNCC should appear where the best match is found between the scaled template and the image I . This maximum can be used for tracking.

Until now we have assumed that the scaling s is known, but that may not be realistic in general. Theoretically, it is possible to check the entire range of template scalings, but this solution is computationally infeasible. In agreement with our earlier assumption that the target and camera are moving smoothly, we propose simplifying the problem of scale detection by assuming that the scale does not change abruptly and that it can be modeled as a simple Markov process, i.e., for the frame k :

$$s_{k+1} = s_k + v_k; \quad s_0 = 1, \tag{2.6}$$

where v_k (the uncertainty in scale) is modeled as a random noise process described by its prior probability distribution. Assuming that most of the time the scale does not change very much ($v_k \approx 0$), the minimal scale is 0, and the maximal scale is s_{max} , we propose using the truncated normal distribution for v_k :

$$v_k \sim N(0, \sigma^2); \quad v_k \in (-s_k, s_{max} - s_k). \tag{2.7}$$

Remark—If some prior knowledge about changes in scale is available, this knowledge can be incorporated into the model by modifying the distribution of v_k . For example, if we suppose that most of the time the scale will not change, then we should choose the truncated normal distribution added to delta distribution at zero.

2.3. Multiple scaled normalized cross-correlation (MSNCC)

The definition of RSNCC fits well into the particle filtering framework and makes the estimation of scale tractable. The problem is that this technique may fail in the case of occlusion of the desired target in the image I or if the target is partially deformed. In addition, the peak of SNCC is blunt and not always appropriate for accurate tracking (see Figure 2).

To overcome these problems, we propose using MSNCC coefficients. The idea is to divide the scaled template into M rectangular subregions (patches) P_i ($i=1, \dots, M$), not necessarily disjoint, and to compute the RSNCC map for each subregion. Then, all the maps are fused to a single one, according to the offsets of the subregions in the template, namely,

$$MSNCC(m, n, s) = \frac{1}{M} \sum_{i=1}^M RSNCC_{P_i}(m+m_i, n+n_i, s), \quad (2.8)$$

where

$$\begin{aligned} P_i &= \text{the subregion \#}i, \\ m &= 1, 2, \dots, M_x - \lfloor sN_x \rfloor + 1, \\ n &= 1, 2, \dots, M_y - \lfloor sN_y \rfloor + 1. \end{aligned} \quad (2.9)$$

Here (m_i, n_i) denotes the offset of the i th patch according to the top-left corner of the template ($0 \leq m_i \leq \lfloor sN_x \rfloor - 1$, $0 \leq n_i \leq \lfloor sN_y \rfloor - 1$) (see Figure 1 with scaled T).

The position of the maximal value of the MSNCC function is equivalent to the position of the best match. This result is obtained by “major voting” of the individual RSNCC maps. Figure 2 illustrates the difference between the RSNCC and MSNCC. This example is constructed with the same image and template. The peaks of MSNCC are sharper (compared to RSNCC), which makes this technique more appropriate for exact localization. Sullivan et al. [38] discussed the problem of incorporation of likelihoods that are too sharply formed in the particle filtering framework. In our case, the MSNCC never degenerates into a very sharp peak, and thus the problem is less significant.

2.4. Numerical example

In this subsection, we explore a simple numerical example which shows properties of MSNCC and its advantages over NCC.

Suppose that the target is given by

$$T = \begin{pmatrix} 10 & 20 & 50 & 50 \\ 10 & 20 & 70 & 70 \\ 20 & 30 & 100 & 100 \\ 20 & 30 & 80 & 80 \end{pmatrix}. \quad (2.10)$$

Let us begin with ideal conditions, when the tested candidate for target is equal to T , and thus $I = T$. In this case,

$$NCC = \begin{pmatrix} .27 & .38 & .33 & .32 & .02 & -.28 & -.23 \\ .52 & .75 & .64 & .69 & .13 & -.42 & -.33 \\ .51 & .8 & .7 & .81 & .09 & -.59 & -.44 \\ .57 & .9 & .78 & 1 & 0 & -.82 & -.57 \\ .46 & .69 & .51 & .54 & -.11 & -.73 & -.53 \\ .15 & .23 & .12 & .05 & -.27 & -.61 & -.46 \\ .02 & .03 & -.03 & -.09 & -.24 & -.4 & -.32 \end{pmatrix}. \quad (2.11)$$

Note that the best match (1 with the coordinates (4, 4)) is obtained for the lower-right target's point; i.e., two images are matched when the lower-right point of the tested target is located at (4, 4), as expected.

To compute MSNCC, we divide the template into four equal and nonoverlapping 2×2 quadrants. The appropriate MSNCC is given by

$$MSNCC = \begin{pmatrix} 0 & 0 & .14 & .08 & .09 & 0 & 0 \\ 0 & 0 & .25 & .25 & .24 & 0 & 0 \\ .14 & .23 & .58 & .5 & .47 & 0 & 0 \\ 0 & 0 & .6 & 1 & .55 & 0 & 0 \\ .22 & .42 & .66 & .72 & .52 & 0 & 0 \\ 0 & 0 & .25 & .25 & .24 & 0 & 0 \\ 0 & 0 & .14 & .05 & .1 & 0 & 0 \end{pmatrix}. \quad (2.12)$$

As one can see, the same maximum is attained at the same place (1 with the coordinates (4, 4)). Computing the variances of NCC and MSNCC around the middle point (4, 4), we obtain

$$Var(NCC) = \sum_{i=1}^7 \sum_{j=1}^7 ((i-4)^2 + (j-4)^2) |NCC(i, j)| = 138.5, \quad (2.13)$$

and for MSNCC

$$Var(MSNCC) = \sum_{i=1}^7 \sum_{j=1}^7 ((i-4)^2 + (j-4)^2) |MSNCC(i, j)| = 26.5, \quad (2.14)$$

which is much smaller and thus better localized.

We claim that the change in the measurements should be proportional to the target's shift. For example, if the tested candidate target has been shifted to the right by 50% of its size, then the similarity measure between the reference template and the shifted candidate template should be reduced to one-half of its maximum value. It is easy to see that MSNCC is much closer than NCC to the expected function (at least for small translations). Moreover, MSNCC may better handle occlusions, as shown in the next example.

Indeed, suppose that the lower half of image I is saturated by glare (say from the sun). This kind of corruption may be regarded as an occlusion. The corrupted image is given by

$$I = \begin{pmatrix} 10 & 20 & 50 & 50 \\ 10 & 20 & 70 & 70 \\ 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 \end{pmatrix}. \quad (2.15)$$

The NCC for this image is

$$NCC = \begin{pmatrix} .27 & .38 & .34 & .32 & .02 & -.28 & -.23 \\ .52 & .75 & .64 & .69 & .13 & -.42 & -.33 \\ .3 & .46 & .38 & .26 & -.05 & -.44 & -.35 \\ .53 & .83 & .63 & .41 & -.08 & -.6 & -.45 \\ .46 & .71 & .45 & .17 & -.2 & -.62 & -.46 \\ .15 & .23 & -.02 & -.33 & -.44 & -.61 & -.46 \\ .02 & .03 & -.12 & -.28 & -.33 & -.4 & -.32 \end{pmatrix}. \quad (2.16)$$

One can see that the maximum is obtained at the wrong place (0.83 with the coordinates (4, 2)) and that $Var(NCC) = 132.8$.

In contrast, for the same template and corrupted image,

$$MSNCC = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ .29 & .45 & .72 & .65 & .47 & 0 & 0 \\ 0 & 0 & .62 & 1 & .55 & 0 & 0 \\ .15 & .35 & .61 & .85 & .55 & 0 & 0 \\ 0 & 0 & .5 & .5 & .49 & 0 & 0 \\ 0 & 0 & .29 & .1 & .19 & 0 & 0 \end{pmatrix}. \quad (2.17)$$

The two lower quadrants will not contribute to the MSNCC; thus the maximum is obtained at the right location, and the variance is still lower ($Var(MSNCC) = 28.4$) than the variance of NCC.

In conclusion, if an appropriate location for the patches P_i (in MSNCC computation) has been chosen, then MSNCC has a better localization and lower variance than NCC.

2.5. Occlusion handling by particle filtering

In this section, we explore the question of choosing the proper location and size of the patches P_i in the reference template. Basically, we would like to choose subregions whose location is not occluded in the current frame I ; these patches have higher probability in order to have better correlation with I . If we know some prior information on the target, this knowledge can be incorporated into the choice of a prior patch distribution as well. It is not practical to check all locations and sizes of all the patches, and thus we propose solving the problem approximately by particle filtering. The idea is to distribute the patches randomly based on some prior probability distribution and, after measurement, to select the more probable ones. This process can be repeated to obtain a better occlusion estimate, with a price being paid on the computational complexity.

Let us denote the state of the patch for the frame k by $r_k^i = (m_i, n_i, \text{height}_i, \text{width}_i)_{i=1}^M$, where M is the number of patches, (m_i, n_i) is the location of top-left corner of P_i (see (2.9)), and $(\text{height}_i, \text{width}_i)$ is the height and width of P_i , respectively. The hidden state sequence r_k that we estimate is assumed to be Markovian. Suppose that the measurements Z_k are available and supply sufficient information to differentiate between the occluded and unoccluded subregions.

One possible option is to extend the general system's state vector with patch state data; this solution is computationally demanding and requires a significant increase in the number of necessary particles. We have chosen another solution: to run a second particle filter (from now on referred to as the *embedded particle filter*), which finds the correct distribution of the subregions from the prior likelihood and actual correlation measurement (i.e., RSNCC).

More precisely, the embedded particle filter recursively estimates the posterior distribution $p(r_k|Z_1, \dots, Z_k)$ from the likelihood $p(Z_k|r_k)$ and the state transition probability $p(r_k|r_{k-1})$ as follows:

$$p(r_k|Z_1, \dots, Z_k) \propto p(Z_k|r_k) \int p(r_k|r_{k-1}) p(r_{k-1}|Z_1, \dots, Z_{k-1}) dr_{k-1}. \quad (2.18)$$

The recursive approximation to (2.18) is obtained from [35]:

$$p(r_k|Z_1, \dots, Z_k) \approx \sum_{i=1}^M w_k^i \delta(r_k - r_k^i), \quad (2.19)$$

where $\delta(\bullet)$ denotes the Dirac delta function and r_k^i denotes M independent identically distributed samples (particles) drawn from the posterior distribution. The weights w_k^i are given by

$$w_k^i \propto w_{k-1}^i p(Z_k|r_k^i). \quad (2.20)$$

The following constant position model is used for prediction of the patch distributions:

$$r_{k+1} = r_k + \eta_k; \quad \eta_k \sim N(0, \Sigma). \quad (2.21)$$

The noise η_k should be sampled from the normal distribution truncated to the size of the template, i.e., each patch placed inside the template.

The algorithm for patch distribution is summarized below (based on [35]).

In the first frame, the M patches are distributed uniformly within the bounding box defined by the reference template size. All patches have the same importance (weight). Then the particles are diffused according to the motion prediction model. Some of the M patches may fall outside the bounding box, and these are redrawn uniformly again. This step allows one to return patches to regions of temporarily long occlusion and temporarily low correlation. The next step is to obtain the patches from the following frame, and to measure the weight of each particle (the correlation to the reference template). If the correlation with the patch is high, then there is a high probability that the region close to this patch is not occluded or deformed too much. Therefore, by the resampling process, the patches with low correlation are moved to the highly rated locations. At this time, some of the patches may take the same size and location, but they will be diffused around this location by the prediction procedure just before reading the next frame. In this manner, most of the patches are distributed in the regions of low probability of partial occlusion. The bounding box moves according to the dynamics governing the particle filter described in the following section, and the preceding algorithm is repeated for the following frame. Note that all the patches obtained from the algorithm above are used as the a prior distribution in the following video frame, and hence the fact that the regions of occlusion do not suddenly appear or disappear is exploited naturally.

The example of patch distribution in the case of occlusion is shown in subsection 5.2.

To conclude this section, we note that changing the arithmetic means to the more statistically robust medians in (2.4), as suggested in [40], can improve the tracking, especially in image regions with depth discontinuities.

In the next section, we will employ the MSNCC measurement in a particle filtering framework with a small number of state parameters.

3. Particle filtering framework

In this section, we describe the framework for visual tracking based on particle filtering. The tracking begins with the selection of the target in the first frame of the video. This selection can be made manually by the user or via some detection algorithm. The reference template is defined as the gray-levels in the selected bounding box around the target. For each video frame, the search regions are selected by the motion dynamics model (image regions in accord with the predicted states). The reference template is compared to the selected regions by some similarity measure (measurement step), and the posterior likelihood distribution is found by Bayes' rule. Finally, the target's location is computed using the posterior likelihood, and the template is updated to keep up with the target's nonrigid deformations.

State definition—The state of the system consists of all the parameters one wishes to estimate. In our case, the target's location, velocity, and scale are the parameters that the particle filter should estimate. Therefore, all of those parameters are included in the state (for the frame k):

$$X_k = (x, y, v_x, v_y, s), \quad (3.1)$$

where

$$\begin{aligned} (x, y) &= \text{target's location in } I, \\ (v_x, v_y) &= \text{target's velocity,} \\ s &= \text{target's scale.} \end{aligned}$$

The main goal of the particle filter is to estimate the hidden parameters X_k using the measurement Z_k . The detailed explanation on the state space and dynamic models may be found in [5]. In the present work, we use a constant velocity dynamic model [5] for its simplicity and reasonable results, but any other model may be used instead. The constant velocity assumes that the target's velocity is constant up to some tolerance factor. If another model is considered, then the state should be updated accordingly.

State prediction—The state prediction is made (before the measurement) to define the search region where the target is expected to be found. Using the Markovian model assumption, the prediction is described by the transitional probability distribution $p(X_k/X_{k-1})$. This distribution is computed by means of the dynamic motion model:

$$X_k = \begin{pmatrix} 1 & 0 & 1/f & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1/f & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} X_{k-1} + v_{k-1}, \quad (3.2)$$

where $v_{k-1} \sim N(0, \Sigma)$ and $1/f$ is the reciprocal of frame rate.

Remarks:

- Other distributions for v_{k-1} may be employed, based on some prior motion statistics, if available.

- The covariance matrix Σ is diagonal. Its main diagonal values will determine the location and radius of distribution of particles around the previous target's location.

Measurement—The heuristic representative for the likelihood $p(Z_k|X_k)$ is computed from actual measurements taken from the video frame k . We have defined the MSNCC to have values in the interval $[0, 1]$, and it is possible to assign

$$p(Z_k|X_k) \propto \text{MSNCC}, \quad (3.3)$$

where MSNCC is measured between the template T and the current video frame I by means of Algorithm 1.

Resampling—To solve the problem of particle degeneracy, where only a few (or one) of the most prominent particles affect the distribution approximation, we use the notion of *effective particle number* proposed by Kong, Liu, and Wong [20]:

$$\widehat{N}_{eff} = \frac{1}{\sum_{i=1}^N (w_k^i)^2}. \quad (3.4)$$

If $\widehat{N}_{eff}$ is lower than some predefined threshold, then the particles should be resampled according to [35, p. 42].

State estimation—The state is estimated based on the previous frame posterior estimation, the transitional probability distribution, and the likelihood:

$$\begin{aligned} & p(X_k|Z_1, \dots, Z_k) \\ & \propto p(Z_k|X_k) \int p(X_k|X_{k-1}) p(X_{k-1}|Z_1, \dots, Z_{k-1}) dx_{k-1}. \end{aligned} \quad (3.5)$$

Template update—For most of the videos of interest in the present work, a fixed reference template, acquired in the first frame of the video, is sufficient for robust tracking. Nevertheless, most real-world targets exhibit deformations that make such a fixed template inappropriate for robust tracking. On the one hand, constantly changing the template can cope with deformations of the target. On the other hand, when the template is updated too fast, the problem of drift [26] and inclusion of occluded parts as a part of the template may occur. There is no universal solution for these problems, and so we propose a simple compromise by using a slowly changing template T . Specifically, for the k th frame,

$$T_k = \alpha T_{k-1} + (1 - \alpha) \widehat{T}_k, \quad (3.6)$$

where $\alpha \in [0, 1]$ is the smoothing factor proportional to RSNCC [44] and $\widehat{T}_k$ is the target where $\alpha \in [0, 1]$ is the smoothing factor proportional to RSNCC image estimated in the current frame. In general, for slowly varying targets α should be very close to 1.

In the following section, we provide the details for the implementation of the particle filtering algorithm and discuss the choice of parameters for the algorithm.

4. Tracking algorithm

The algorithmic implementation for the approximate evaluation of the posterior distribution $P(X_k|Z_1, \dots, Z_k)$ in each frame and the estimation of the target's location are described below.

Parameters of algorithm

The algorithm uses the parameters N , M , Σ , s_{max} , which are selected in advance. There are no exact rules for choosing these parameters, but the following considerations may help:

- In general, the empirically chosen $N \in [30, 120]$ works for the videos that were tested. An increase in the number of particles increases the computation time and does not significantly affect the quality of tracking.
- The parameter M should be chosen to provide template tiling with patches (with possible overlapping). Assuming that the patches are distributed uniformly, it is easy to verify that M does not depend on the template size; practically, $M \in [10, 20]$.
- The matrix Σ should be diagonal. We need to estimate the standard deviation of the state that can occur between the frames, i.e., how much the target can move between the frames, how much the target can change its velocity, and what is the possible change in scale between two consecutive frames. The values of the estimated standard deviations are the values of the main Σ diagonal.
- From the simple fact that the scaled template cannot be larger than the image, it is clear that $s_{max} \lesssim \min(M_x/N_x, M_y/N_y)$.
- The smoothing factor α that is used for template adaptation can be set to 1 for targets with negligible deformations and to $0.95 < \alpha < 1$ for targets with larger deformations.

5. Experimental results and discussion

In this section, we demonstrate some of the results of our tracking algorithm, Algorithm 2, with challenging real-world highly cluttered scenes. The results are compared to the mean-shift [9, 10], CONDENSATION [16], and adaptive IMED [27] methodologies. In the figures, we have shown only the centroids obtained by the CONDENSATION and mean-shift algorithms. The examples presented here show the efficiency and advantages of our algorithm for target tracking and elucidate its behavior under some difficult scenarios.

In each example, the following hold:

- No color information is available.
- The video resolution is 240×320 pixels with 30 frames per second.
- The target is selected manually in the first video frame, and the template remains fixed along the video sequence.
- The number of particles in the embedded particle filter is set to be $M = 15$.
- The location standard deviation is 1 pixel per frame.
- The scale change deviation is 0.1 pixel per frame.
- The threshold for the number of effective particles is set to be $N_{thr} = 0.5N$.

5.1. Sequence 1: Two people walking together (with partial occlusion)

In the first video sequence, two people are walking together in outdoor conditions. At certain points of the video, they are partially or fully hidden behind the trees. Both persons are similar in their appearance. The number of particles that were used in our algorithm is fixed at $N = 80$.

Figure 3 shows the tracking results of four different trackers: our tracker (marked by solid white rectangle), adaptive scale-invariant IMED-based algorithm (marked by dashed rectangle), CONDENSATION algorithm (marked by white “o”), and the mean-shift algorithm (marked by white “+”). The selected frame number appears in the bottom-left corner of the appropriate frame. The constant reference template that is used by our algorithm appears at the bottom-right corner of frame 16.

All the tested trackers managed to follow the target until frame 160. The mean-shift algorithm was misled by the partial occlusion (frame 160) and by the second person (frames 400–530). The CONDENSATION algorithm lost track at the second tree (frame 530), and the IMED-based algorithm lost track after 200 frames, while our algorithm robustly tracked the target for the entire sequence.

5.2. Sequence 2: Walking person (with full occlusion and scaling)

In this sequence, we track a person who is approaching the camera and walking behind a tree. The scene clutter and complex back illumination make this scene challenging for the tested trackers. The number of particles that were used in our algorithm is $N = 100$.

Figure 4 shows the results. In the first few frames all the algorithms tracked the target correctly, but the IMED-based, mean-shift, and CONDENSATION algorithms ceased functioning after frames 80, 70, and 95, respectively. Our tracker temporarily lost track when the full occlusion occurred (frame 95), but afterwards it proceeded with correct tracking. Also, the trajectory obtained by our tracker was smoother than the trajectories of the other three algorithms.

To better understand the patch distribution over the template, we show some examples of some intermediate results of Algorithm 1 tracking through occlusion. The brightness of the patch borders is approximately proportional to the weight of the appropriate patch.

It can be seen that after resampling (from A to B), most of the irrelevant patches are moved to locations of higher correlation. At this stage, some patches have the same location.

Then the patches are diffused by the prediction procedure (Figure 5C). It is clear that after the resampling (Figure 5D) most of the patches are distributed over the unoccluded region. Similar results are obtained when the target exits the occluded region (see Figure 6).

5.3. Sequence 3: Crowded party (with occlusions and strong clutter)

In this sequence, we track a person at a crowded party. The scene is strongly cluttered, and the target is partially or fully occluded. The number of particles that were used in our algorithm is $N = 80$.

Figure 7 shows the results. From frame 120, the CONDENSATION and the mean-shift algorithms lose track because of strong clutter. Our algorithm tracked successfully for the entire sequence, although in some frames the results of the above algorithms were closer to the real target (see frame 80). The IMED-based algorithm managed to track the target better than our algorithm until frame 150. After that frame incorrect scale estimation caused a wrong lock from which the algorithm was unable to recover.

5.4. Sequence 4: Maneuvering vehicle (with scaling)

In this sequence, we track a maneuvering vehicle. The camera zooms, and at the end of the video the target vehicle occupies the entire frame. The number of particles that were used in our algorithm is $N = 30$.

Figure 8 shows the results of all trackers. Despite the changes in scale and appearance, our algorithm robustly tracked the target with a small number of particles. The IMED-based algorithm was also able to robustly track the target. In contrast, as the result of various terrain features, the CONDENSATION and mean-shift algorithms provided nonsmooth trajectories, and the CONDENSATION tracker lost track after frame 110.

6. Conclusions

In this paper, we presented an algorithm for robust target tracking in video sequences. The examples provided indicate that our algorithm is able to track scaled and partially occluded (or fully occluded for short times) targets without the need for learning or adaptation. The algorithm can handle clutter, noise, low contrast, and camera motion. The incorporation of modified cross-correlation in the particle filtering framework makes it possible to get smooth target trajectories. The examples clearly show the advantages of our approach over the traditional CONDENSATION and mean-shift algorithms as well as recent adaptive algorithms. Our algorithm may have a wide area of applicability due to its generality, since the method does not impose many constraints on the target of interest.

There are still several key limitations on our target tracking scheme. The algorithm is unable to redetect lost targets and to follow rotated targets or those that are occluded for many frames. Also, the choice of the parameters is a nontrivial task, though they can be estimated by learning the motion patterns in typical sequences. The solutions to these problems as well as tracking multiple interacting targets should be addressed in future work. Also in future research, we will explore whether other similarity measures are more appropriate for the proposed tracking framework.

REFERENCES

- [1]. Adam, A.; Rivlin, E.; Shimshoni, I. Robust fragments-based tracking using the integral histogram. Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR); 2006. p. 798-805.
- [2]. Anuta PE. Digital registration of multispectral video imagery. Soc. Photo-Opt. Instrum. Eng. J. 1969; 7:168-175.
- [3]. Arulampalam MS, Maskell S, Gordon N, Clapp T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans. Signal Process. 2002; 50:174-188.
- [4]. Aschwanden, P.; Guggenbuhl, W. Robust Computer Vision: Quality of Vision Algorithms. Wichmann; Karlsruhe, Germany: 1992. Experimental results from a comparative study on correlation-type registration algorithms; p. 268-282.
- [5]. Bar-Shalom, Y.; Fortmann, TE. Tracking and Data Association. Academic Press; San Diego: 1988.
- [6]. Berthilsson, R. Affine correlation. In: Jain, AK.; Venkatesh, S.; Lovell, BC., editors. Proceedings of the 14th International Conference on Pattern Recognition; Los Alamitos, CA: IEEE Computer Society; 1998. p. 1458-1460.
- [7]. Blackman, S.; Popoli, R. Design and Analysis of Modern Tracking Systems. Artech House; London: 1999.
- [8]. von Seelen, U. M. Cahn; Bajcsy, R. Adaptive correlation tracking of targets with changing scale. In: Firschein, O.; Strat, T., editors. Reconnaissance, Surveillance, and Target Acquisition for the Unmanned Ground Vehicle. Morgan Kaufmann; San Francisco: 1997. p. 313-322.
- [9]. Comaniciu, D.; Ramesh, V. Mean shift and optimal prediction for efficient object tracking. Proceedings of the IEEE International Conference on Image Processing (ICIP); Vancouver. 2000. p. 70-73.
- [10]. Comaniciu, D.; Ramesh, V.; Meer, P. Real-time tracking of non-rigid objects using mean shift. Proceedings of the 2000 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR); Hilton Head Island, SC. 2000. p. 142-149.

- [11]. Daum, F.; Huang, J. Curse of dimensionality and particle filters. *Proceedings of the 2003 IEEE Aerospace Conference*; 2003. p. 1979-1993.
- [12]. Doucet, A.; de Freitas, N.; Gordon, N. *Sequential Monte Carlo Methods in Practice*. Springer; New York, London: 2001.
- [13]. Gordon NJ, Salmond DJ, Smith AFM. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*. 1993; 140:107-113.
- [14]. Guerra, C.; Hernandez, M.; Dominguez, A.; Hernandez, D. A new approach to the template update problem. In: Marques, J.; de la Blanca, N.; Pina, P., editors. *Pattern Recognition and Image Analysis, Lecture Notes in Comput. Sci.* Vol. 3522. Springer-Verlag; Berlin, Heidelberg: 2005. p. 217-224.
- [15]. Guo, G.; Dyer, CR. Patch-based image correlation with rapid filtering. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; Minneapolis, MN. 2007. p. 2883-2888.
- [16]. Isard M, Blake A. CONDENSATION—Conditional density propagation for visual tracking. *Int. J. Comput. Vision*. 1998; 29:5-28.
- [17]. Jurie, F.; Dhome, M. Real time robust template matching. *Proceedings of the 13th British Machine Vision Conference*; Cardiff, UK. 2002. p. 123-132.
- [18]. Kawamoto, K. Affine modeling of targets in video sequences by particle filters. *Proceedings of the World Automation Congress*; Budapest, Hungary. 2006; New York: IEEE; 2007. p. 1-6.
- [19]. Kolmogorov V, Criminisi A, Blake A, Cross G, Rother C. Probabilistic fusion of stereo with color and contrast for bi-layer segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 2006; 28:1480-1492. [PubMed: 16929733]
- [20]. Kong A, Liu JS, Wong WH. Sequential imputations and Bayesian missing data problems. *J. Amer. Statist. Assoc.* 1994; 1:278-288.
- [21]. Lewis JP. Fast template matching. *Proceedings of Vision Interface 95, Canadian Image Processing and Pattern Recognition Society, Quebec City*. 1995:120-123.
- [22]. Lim, J.; Ross, D.; Lin, RS.; Yang, MH. *Advances in Neural Information Processing Systems*. Vol. 17. MIT Press; Cambridge, MA: 2005. Incremental learning for visual tracking; p. 793-800.
- [23]. MacCormick J, Blake A. A probabilistic exclusion principle for tracking multiple objects. *Int. J. Comput. Vision*. 2000; 39:57-71.
- [24]. MacCormick, J.; Isard, M. Partitioned sampling, articulated objects, and interface-quality hand tracking. *Proceedings of the 6th European Conference on Computer Vision—Part II*; Springer-Verlag, London. 2000. p. 3-19.
- [25]. Maggio E, Smerladi F, Cavallaro A. Adaptive multifeature tracking in a particle filtering framework. *IEEE Trans. Circuits Syst. Video Technol.* 2007; 17:1348-1359.
- [26]. Matthews I, Ishikawa T, Baker S. The template update problem. *IEEE Trans. Pattern Anal. Mach. Intell.* 2004; 26:810-815. [PubMed: 18579941]
- [27]. Mei, X.; Zhou, SK.; Porikli, F. Probabilistic visual tracking via robust template matching and incremental subspace update. *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*; Beijing, China. 2007. p. 1818-1821.
- [28]. Mitani K, Saji H. Robust template matching by using variable size block division. *Proceedings of the SPIE*. 2005; Vol. 6051:60510X.
- [29]. Nakhmani, A.; Tannenbaum, A. Particle filtering using multiple cross-correlations for tracking occluded objects in cluttered scenes. *Proceedings of the 47th IEEE Conference on Decision and Control*; Cancun, Mexico. 2008. p. 652-657.
- [30]. Nakhmani, A.; Tannenbaum, A. Scale-invariant visual tracking by particle filtering. *Proceedings of the SPIE, Image and Signal Processing for Remote Sensing XIV*; Cardiff, UK. 2008. p. 71090L
- [31]. Nguyen HT, Smeulders AWM. Fast occluded object tracking by a robust appearance filter. *IEEE Trans. Pattern Anal. Mach. Intell.* 2004; 26:1099-104. [PubMed: 15641740]
- [32]. Ooi, J.; Rao, K. New insights into correlation-based template matching. *Proceedings of the SPIE, Applications of Artificial Intelligence IX*; Orlando, FL. 1991. p. 740-751.

- [33]. Pan P, Schonfeld D. Dynamic proposal variance and optimal particle allocation in particle filtering for video tracking. *IEEE Trans. Circuits Syst. Video Technol.* 2008; 18:1268–1279.
- [34]. Rathi Y, Vaswani N, Tannenbaum A. A generic framework for tracking using particle filter with dynamic shape prior. *IEEE Trans. Image Process.* 2007; 16:1370–1382. [PubMed: 17491466]
- [35]. Ristic, B.; Arulampalam, S.; Gordon, N. *Beyond the Kalman Filter: Particle Filters for Target Tracking*. Artech House; London: 2004.
- [36]. Ross DA, Lim J, Lin RS, Yang MH. Incremental learning for robust visual tracking. *Int. J. Comput. Vision.* 2008; 77:125–141.
- [37]. Shao, J.; Zhou, SK.; Chellappa, R. Tracking algorithm using background-foreground motion models and multiple cues. *Proceedings of the 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing*; Philadelphia, PA. 2005. p. 233-236.
- [38]. Sullivan J, Blake A, Isard M, MacCormick J. Bayesian object localisation in images. *Int. J. Comput. Vision.* 2001; 44:111–135.
- [39]. Toyama K, Blake A. Probabilistic tracking with exemplars in a metric space. *Int. J. Comput. Vision.* 2002; 48:9–19.
- [40]. Trujillo, M.; Izquierdo, E. A robust correlation measure for correspondence estimation. In: Aloimonos, Y.; Taubin, G., editors. *Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVI '04)*; Washington, DC: Thessaloniki, Greece, IEEE Computer Society; 2004. p. 155-162.
- [41]. Wang H, Suter D, Schindler K, Shen C. Adaptive object tracking based on an effective appearance filter. *IEEE Trans. Pattern Anal. Mach. Intell.* 2007; 29:1661–1667. [PubMed: 17627053]
- [42]. Wang L, Zhang Y, Feng J. On the Euclidean distance of images. *IEEE Trans. Pattern Anal. Mach. Intell.* 2005; 27:1334–1339. [PubMed: 16119271]
- [43]. Wang, X.; Han, TX.; Yan, S. An HOG-LBP human detector with partial occlusion handling. *Proceedings of the IEEE 12th International Conference on Computer Vision (ICCV)*; 2009. p. 32-39.
- [44]. Wong, S. Advanced correlation tracking of objects in cluttered imagery. In: Masten, MK.; Stockum, LA., editors. *Proceedings of the SPIE, Acquisition, Tracking, and Pointing XIX*; Orlando, FL. 2005. p. 158-169.
- [45]. Yilmaz A, Javed O, Shah M. Object tracking: A survey. *ACM Comput. Surv.* 2006; 38 article 13.
- [46]. Zhao, F.; Huang, Q.; Gao, W. Image matching by normalized cross-correlation. *Proceedings of the 2006 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; Toulouse, France. 2006. p. II-729-II-732.
- [47]. Zhou SK, Chellappa R, Moghaddam B. Visual tracking and recognition using appearance-adaptive models in particle filters. *IEEE Trans. Image Process.* 2004; 13:1491–1506. [PubMed: 15540457]

Algorithm 1

- Initialization

Sample M particles

$$\{r_0^i = (m_i, n_i, \text{height}_i, \text{width}_i)\}_{i=1}^M$$

from the initial uniform distribution (minimal subregion size is 2×2):

$$\begin{aligned} m_i &\sim U[0, N_x - 2], \\ n_i &\sim U[0, N_y - 2], \\ \text{height}_i &\sim U[2, N_x/2 - 1], \end{aligned} \quad (2.22)$$

$$\begin{aligned} \text{width}_i &\sim U[2, N_y/2 - 1], \\ w_0^i &= 1/M, \end{aligned} \quad (2.23)$$

where U is the uniform discrete distribution. Note that $m + \text{height}$ and $n + \text{width}$ should be limited to the intervals $[1, N_x]$ and $[1, N_y]$ accordingly (reject sampling), and thus the actual probability of getting small subregions is higher than the probability of getting large subregions.

- Prediction

For each particle in the k th frame, do the following:

- Draw $r_k^i \sim p(r_k | r_{k-1}^i)$ according to (2.21).
- Reject particles that do not fit the size of the template T . Redraw these particles from the uniform distribution (2.22).
- Proceed to the next frame.

- Measurement

Evaluate the importance weights:

$$\begin{aligned} \tilde{w}_k^i &\propto w_{k-1}^i p(Z_k | r_k^i), \\ w_k^i &= \frac{\tilde{w}_k^i}{\sum_{i=1}^M \tilde{w}_k^i}. \end{aligned} \quad (2.24)$$

To compute the probability distribution $p(Z_k | r_k^i)$, compute RSNCC for each patch:

$$p(Z_k | r_k^i) = p^{(RSNCC)}(Z_k | r_k^i), \quad (2.25)$$

where $p^{(RSNCC)}(Z_k | r_k^i) = \max(RSNCC)$ is the maximum of the SNCC between the reference template T and the patch defined by r_k^i . Note that we use RSNCC as a heuristic proxy for a likelihood.

- Resampling

Resample the particles according to the computed weight w_k (see [35, p. 42]).

- Repeat from the *prediction* stage if needed.
- Compute MSNCC according to (2.8).

Algorithm 2

- *Initialization*
 - Select the reference template in the first frame, and sample N particles $\{x_0^i, w_0^i\}_{i=1}^N$ at the same (selected) location, zero velocity, and unit scale.
 - Distribute the particle weight w_0^i uniformly: ($w_0^i = 1/N$).
 - Sample M uniformly distributed particles $\{r_0^i\}_{i=1}^M$ for the embedded particle filter according to (2.22).

- *Prediction*

For each frame k :

- Distribute the particles according to $p(X_k|X_{k-1})$ by the motion equation (3.2).

- *Measurement*

- For each particle, update the weight w_k^i according to (3.3), and normalize:

$$\begin{aligned}\tilde{w}_k^i &\propto w_{k-1}^i p(Z_k|X_k^i), \\ w_k^i &= \frac{\tilde{w}_k^i}{\sum_{i=1}^M \tilde{w}_k^i}.\end{aligned}\quad (4.1)$$

- *Resampling*

- If the number of effective particles $N_{eff} < N_{thr}$ (see (3.4)), then the particles X_k are resampled according to the weight w_k , and $w_k^i = 1/N$.

- *Estimation*

- Estimate the state (which includes the location) by

$$\hat{X}_k = \sum_{i=1}^N w_k^i X_k^i. \quad (4.2)$$

- If the estimated target location is close enough to the real one, then this location can be used to refine the embedded particle filter particles in order to improve the occlusion detection. This is made by reweighting these particles via Algorithm 1 in accordance with the estimated location.

- *Update*

- Update the template according to (3.6).

- Repeat from the *prediction* stage.

Figure 1.
Definition of image, reference template, and patch P_i dimensions.

Figure 2.
Comparison of RSNCC and MSNCC for the same image and template.

Figure 3.

Scene of two people walking together. The number in the bottom-left corner denotes the frame number. The solid white rectangle shows our tracker's result, the dashed rectangle shows the result of the IMED-based adaptive tracker, the white "o" shows the CONDENSATION algorithm result, and the white "+" shows the mean-shift algorithm result. The white arrows emphasize wrong tracking results.

4. Scene of walking person. The results are marked as in Figure 3.

Figure 5. Patch distribution through entrance behind the occlusion. A: initial distribution; B: distribution after resampling in the last frame; C: distribution after the prediction; D: distribution after resampling in the current frame.

Figure 6. Patch distribution through exit from the occluded region. A: initial distribution; B: distribution after resampling in the last frame; C: distribution after the prediction; D: distribution after resampling in the current frame.

Figure 7.
Scene of crowded party. The results are marked as in Figure 3.

Figure 8.
Scene with maneuvering vehicle. The results are marked as in Figure 3.