An Improved Mean-Shift Tracker with Kernel Prediction and Scale Optimisation Targeting for Low-Frame-Rate Video Tracking

Zhidong Li[1,2], Jing Chen[1,2], Nicol N. Schraudolph[1]
[1]NICTA, Australia; [2]CSE, UNSW, Australia
{Zhidong.Li, Jing.Chen}@nicta.com.au

Abstract

The mean-shift (MS) algorithm is widely used in object tracking because of its speed and simplicity. However, it assumes certain overlap of object appearance and smooth change in object scale between consecutive video frames. This assumption is usually violated in a low-frame-rate (LFR) video, which contains fast motion and scale changes. An LFR video is widely adopted in applications such as surveillance systems, where real-time object tracking is highly desirable but the traditional MS algorithm does not perform well. We addressed this problem by proposing a novel and enhanced mean-shift tracker, named SMDShift, that uses kernel prediction and Stochastic Meta-Descent (SMD) optimization method to deal with the kernel position and scale variation when tracking objects in an LFR video. In our experiments, the SMDShift can track fast moving objects with significant scale change in an LFR video sequence on which the traditional mean-shift and Camshift algorithms fail.

1. Introduction

Low-frames-rate (LFR) video has been widely used in many practical applications such as surveillance system, video streaming, and video archiving etc [4]. Compared with a normal video sequence, an LFR video has a lower frame rate and worse continuity. Therefore, issues of fast motion and abrupt change of object-appearance scale in an LFR video sequence will degrade the tracking performance of a traditional mean-shift (MS) tracker designed for a normal video sequence.

Generally there are two kinds of major approaches in object tracking [3]. One uses the prediction theory to evaluate the probabilistic hypotheses, yielding filtering techniques such as Kalman filters [11] and particle filters [2]; the other, exemplified by the MS tracker [3], uses the statistic distribution of features to localize the object according to the target appearance. Because of its low computation cost and parameter-free nature, the MS tracker has been widely used in many real-time constrained applications such as object tracking, video/image segmentation, and etc.

However, it has been also pointed out that while a MS tracker operates relatively well on a smoothing video sequence with nice continuity, its performance drops significantly in an LFR video sequence [4]. The reasons are, first, the MS tracker relies very much on the sufficient appearance overlap of the object under tracking in consecutive frames. Although Porikli and Tuzel [4] proposed a multi-kernel MS tracker by using a background modelling approach, the issue still remains in a real-time application with complicated background and change illumination. Secondly, the MS tracker applies a fixed or limited-freedom window (kernel) scale by assuming a smooth change in object scale, thus the tracking accuracy will be affected once the MS tracker is applied in an LFR video where the adaptation of the kernel scale cannot catch up with the scale variation of the object appearance. Bradski et al. [5] proposed a continuously adaptive MS (Camshift) algorithm to determine the window scale based on the second moment of the tracked area but it still only suits a case with gradual scale change.

In this paper, we propose an enhanced MS approach, namely SMDShift, by using a novel kernel prediction method to predict the initial kernel position and Stochastic Meta-Descent (SMD) optimisation [10] to dynamically adjust the MS kernel scale while tracking objects in an LFR video. The final testing results indicate that our proposed approach outperforms the traditional MS and Camshift methods very much in terms of tracking accuracy and efficiency in an LFR video sequence.

The remaining parts of the paper are organized as the follows. In section 2, we introduce the main parts of the algorithm, which include the MS algorithm, the kernel prediction (KP) method, the SMD algorithm, and the SMDShift algorithm; Section 3 shows and discusses the experimental results, followed by a conclusion in section 4.

2. Algorithm Description: SMDShift

The processes of the MS algorithm are, 1) manually set a template area as the kernel window; 2) extract the colour histogram q from the area; 3) move the centre of the kernel window to the densest point according to the colour histogram within a given searching area; 4) repeat step 3) to iteratively shift the kernel window till a similarity threshold between colour histogram in the kernel window and q is satisfied [3]. As mentioned before, such a traditional MS tracker requires certain overlap of object appearance and graceful change in object-appearance scale between the consecutive frames. However such requirements are usually unachievable in an LFR video containing fast motion and abrupt change in object scale. To seek a feasible and robust MS tracker to track objects in a LFR video, we therefore in the paper propose the SMDShift algorithm by using a quick kernel prediction method together with a SMD-based kernel scale adaptation scheme to deal with the difficulties when employing a traditional MS tracker in an LFR video.

2.1 SMD Algorithm

Heaps of research is working at improving the efficiency and accuracy of a traditional MS tracker. For instances, [3] adjusts the kernel scale by using predefined scales, however the method is not robust and efficient enough. [6] computes the kernel bandwidth by convolving the image with a set of Gaussian kernels at various scales, however it still targets at a normal video and is expensive in computation, etc. Therefore, to provide a fast and robust approach for object tracking in an LFR video, we use a gradient descent (GD) [7] based method named SMD to cope with the kernel scale issue when using the MS tracker in an LFR video.

In the GD, We define the current kernel scale in the i^{th} iteration to be $h_i = (x_i, y_i)$, a vector comprising the width x_i and height y_i of the kernel window. A cost function $E(h_i)$ is calculated to measure the distance between the optimized scale and the current scale h_i. $E(h_i)$ is iteratively minimized by adjusting h_i according to g_i that is defined as the gradient of $E(h_i)$ on h_i. More detail about $E(h_i)$ will be given in section 2.4.

To speed up the convergence process, for each $h_i = (x_i, y_i)$, a corresponding step size (learning rate) factor a_i is introduced and a *Hadamard* product of a_i and g_i is used to update the parameter h_i:

$$h_{i+1} = h_i - a_i \cdot g_i \tag{1}$$

The GD uses a fixed a_i and results in a low convergence speed [8]. Therefore, given a visual tracking with strict real-time requirement, an advanced scheme with faster convergence speed is desirable. Our literature survey shows that in order to accelerate the GD convergence, Jacobs [9] uses the sign of the autocorrelation of the gradient in the current and the previous iteration to adjust a_i, which is very sensitive to noise; Conjugate gradient [7] uses an adaptive mixture of the current and previous gradients; and Bray [8] tried to speed up the convergence by reasoning that the current frame strongly depends on the previous one, etc. In 1999 Schraudolph introduced SMD [10], which was then employed by Bary [8] in tracking the human articulated structures. The SMD introduces an adaptive step factor a_i and comprises of the following 5 steps in iteration i:

1. Calculate g_i;
2. Update the step size a_i by:
$$a_i = a_{i-1} \cdot \max(0.5, 1 - \mu v_i \cdot g_i) \tag{2}$$
3. Update h_i by (1).
4. Optionally, apply constraints to h_i
5. Update v_{i+1} by:
$$v_{i+1} = \lambda v_i - a_i \cdot (g_i + \lambda H_i v_i) \tag{3}$$

where v_i is an auxiliary vector measuring the impacts of the previous step size at the current parameter values overtime and being constrained by the constant $0<\lambda<1$. H_i is the Hessian (matrix of second derivatives) of $E(h_i)$ in iteration i, and μ is a constant learning rate for the step size a_i.

Nevertheless, the original SMD model successful applied in hand tracking [8] still cannot be employed directly in a histogram-based approach such as the MS approach that uses histogram gradient as its inputs. In following, before describing the proposed SMDShift algorithm, we will discuss how to merge the SMD optimization method into a MS tracker with the kernel prediction functionality.

2.3. Kernel Prediction

Predicting the kernel position is a key process in tracking object in an LFR video where the overlap of object appearance between consecutive frames becomes invisible. Porikli in [4] proposed a background modelling approach to estimate the kernel positions by using a multi-kernels MS method to locate the real kernel. But the method highly relies on the background modelling and becomes less efficient under real-time requirement and complicated environment.

Here, we propose a fast kernel prediction (KP) method aiming to real-time detect kernel position in an LFR video. The main idea of the KP method is to find a block that maximally overlaps with the target appearance, then to initiate the kernel position to the block centre where a normal MS is triggered to shift the kernel window to its optimum. Obviously, the size of the block is critical in determining the efficiency of KP. In the paper, for each new frame, the back-projected frame is divided into MxN blocks with their sizes equal to the kernel scale finalized in the previous tracking; Then the zeroth moment of each block is calculated and the block with maximum zeroth moment is finally initiated as new kernel position in the new frame. In the experiment, we find that such a KP approach is very effective in catching the kernel position in a new frame without a time-consuming background modelling approach.

2.4. SMDShift

To speed up the tracking process, in the SMDShift, a two-steps approach is used by first employing KP to obtain the initial kernel position in the new frame followed by running a standard MS to shift the kernel to its optimal position; then in the second step, the SMD is literately employed to adjust the kernel scale till a distance criteria is satisfied. Figure 1 shows the flow of the SMDShift in detail. A threshold θ_2 is set to terminate a SMD optimization process.

```
SMDShift algorithm
Input: the tracking kernel scale h from the previous frame
1.  Run the KP algorithm with h to locate the block and set
      c0 as the block centre;
2.  Run the MS tracker to renew c0;
3.  i=2;
4.  while( |hi - hi-1| > θ2 or Max Iteration not reached)
5.      Based on hi-1, use the SMD steps to update hi;
6.      i++;
7.  end while
8.  c0' = c0;
9.  Run the MS tracker by c0 and hi;
10. if(|c0'-c0| <θ1 or Max Iteration reached)
11.     break;
12. else
13.     go to step 1;
14. end if
15. return c0' and h
End SMDShift algorithm
```

Figure 1: The process of the SMDShift algorithm

To minimize the $E(h_i)$, we denote function F^j *as* the normalized value of the j^{th} bin in the predicted histogram distribution p_i in the i^{th} SMDShift iteration, where $p_i=\{p_i^1, p_i^2 \ldots p_i^m\}$, m is the number of the bins being used. Thus, given c_0 and h, for each p_i^j, we have:

$$p_i^j = F^j(c_0, h_i) \tag{4}$$

In the paper, the distance between p_i and q is measured by the non-negative Jeffrey Divergence C:

$$C = \sum_j (p_i^j \log \frac{p_i^j}{q^j} + q^j \log \frac{q^j}{p_i^j}) \tag{5}$$

Given (4) and (5), we have cost function:

$$E(h_i) = \sum_j (F^j(h_i) \log \frac{F^j(h_i)}{q^j} + q^j \log \frac{q^j}{F^j(h_i)}) \tag{6}$$

Note that the original $E(h_i)$ derived from a natural image usually contains lots of noises therefore we apply a Gaussian filter to remove the noises and smooth the distribution $E(h_i)$. Thus, we have gradient

$$g_i = \frac{\partial E}{\partial h_i} = \frac{\partial E}{\partial p_i} \frac{\partial p_i}{\partial h_i} \tag{7}$$

Where $\partial E / \partial p_i$ is computed analytically as the derivative of (5). To obtain $\partial p_i / \partial h_i$, we calculated the histogram based on kernel scale h_i+l by applying a small perturbation l=(l_x, l_y) to both width and height components of the h_i and got approximation

$$\frac{\partial p_i}{\partial h_i} \approx \frac{F(c_0, h_i + l) - F(c_0, h_i)}{l} \tag{8}$$

Similarly, Hessian H_i of $E(h_i)$ in iteration i can also be calculated from g_i using the same method.

However, a large a_i in the SMD may over-adjust the kernel scale to exceed the frame scale. We therefore strictly constrain h_i and h_i+l to lie within the video frame scale to keep the histogram distribution retrievable, which is processed in the 4th step of the SMD processing.

Besides, we found that a same change step in the SMD has more impact at a smaller-scale kernel than a larger one. This is due to that in most cases the smoothed $E(h_i)$ is a concave function that is generally not amenable for a gradient-based optimization, therefore in the SMDShift the concave $E(h_i)$ function is converted to a corresponding convex function $E^{'}(h_i)$ before further processing.

3. Experiments

We compared the performance of the SMDShift with the fixed-scale MS and the Camshift implemented from OpenCV [1] by using three LFR video testing scenarios with resolution of 320x240 and frame rate of 1fps. The SMD parameters were set to μ=0.05 and λ=1.

Figure 2 shows some testing results. Each row represents a testing scenario. Scenario (a) contains an object with shrinking scale while (b) contains an object with increasing scale; in (c), we deliberately tracked a fast moving face with significant scale changes; in (d), we tracked a fast moving object strictly without any object-appearance overlap between the neighboring frames. All final results indicate that the SMDShift produces better results than the other two. In scenario (a) and (b), the SMDShift can more quickly adjust the object scales to their optimums and thus improve the tracking accuracies when compared with the other two. In (c), the SMDShift demonstrates its robustness by locking the target face firmly and obviously outperforms the other two algorithms that totally failed in capturing the fast moving face during tracking. In the (d), it further demonstrates the efficiency of using SMDShift in a typical LFR video sequence where object-appearance overlap does not exist strictly in the consecutive frames, while the other two totally got loss again during tracking.

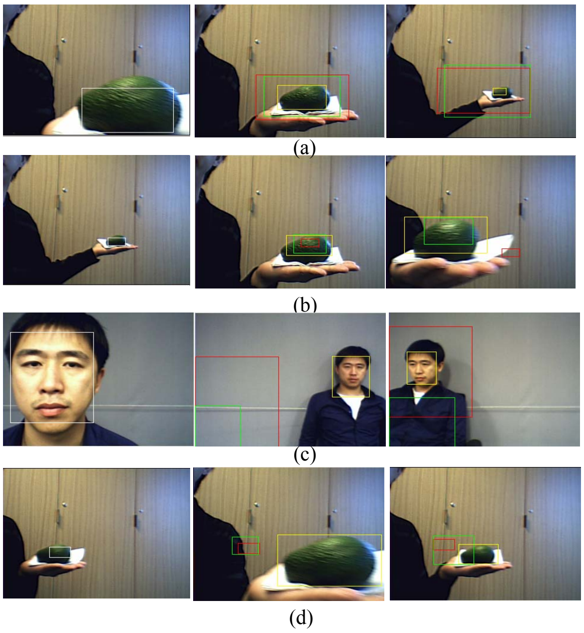

Figure 2: Comparison of tracking fast moving and scaling targets in LFR videos. Each row is a result from a testing scenario. The most left images are the initial image respectively. The yellow bounding box indicates results from the SMDShift, the red one from the fixed scale MS, and the green one from the Camshift implemented.

To further evaluate the efficiency of the SMDShift, we compare the average number of iterations used in getting convergence between the SMDShift and the GD by using identical system parameters a_0 and θ_2. When repeating the three testing scenarios, it shows that the SMD can significantly reduce the iteration amount and thus achieve a faster tracking speed, of which the results are illustrated in Table 1.

4. Summary and Future Works

We present a novel algorithm named SMDShift to deal with the popular kernel position and scale issues when employing a traditional MS tracker in an LFR video containing fast object movement and scale change. The novelties of out paper are the kernel prediction scheme KP and the optimization scheme in updating the kernel scale. The testing results indicate that the SMDShift performs better in an LFR video when compared with the state of the arts. In addition, the SMDShift can also accelerate the tracking process to make it very suitable for a real-time application.

In the future, apart from working at multi-object tracking, we will also enhance the algorithm by parallel running multiple SMDs with stochastic initial kernel scales to overcome the situation that the SMDshift may converge at a local minimum sometimes.

Table 1: Comparisons between the SMDShift and the GD -- number of iterations to get convergence. The Max Iteration is set to 300.

	GD	SMDShift
Scenario 1	40.5	14
Scenario 2	Max Iteration	19.33
Scenario 3	Max Iteration	7.33

Acknowledgment. National ICT Australia (NICTA) is funded by the Australian Government's *Backing Australia's Ability* initiative, in part through the Australian Research Council.

References

[1] Intel OpenCV library http://www.sourceforge.net/projects/opencvlibrary.

[2] M. Isard and A. Blake, "Condensation-Conditional Density Propagation for Visual Tracking," International Journal of Computer Vision, vol.29, no.1, pp.5-28, 1998.

[3] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," IEEE Trans. on Pattern Analysis and Machine Intelligence, pages 564–577, May 2003.

[4] F. Porikli and O. Tuzel, "Object tracking in low-frame-rate video," SPIE Image and Video Communications and Processing, 5685:72–79, 2005.

[5] G. R. Bradski, "Computer Vision Face Tracking for Use in a Perceptual User Interface," IEEE Workshop on Applications of Computer Vision, Princeton, NJ, p.214-219, 1998.

[6] R. T. Collins, "Mean-shift blob tracking through scale space," CVPR, 2003.

[7] W. Press, S. Teukolsky, W. Vetterling and B. Flannery, "Numerical Recipes in C," second ed. Cambridge Univ. Press, 1992.

[8] M. Bray, E. Koller-Meier, P. Müller, L. Van Gool and N. N. Schraudolph, "3D Hand Tracking by Rapid Stochastic Gradient Descent using a Skinning Model," 1st European Conference on Visual Media Production (CVMP), pp.59-68, 2004.

[9] R. A. Jacobs, "Increased Rates of Convergence Through Learning Rate Adaptation," Neural Networks, vol., no.4, pp.295-307, 1988.

[10] N. N. Schraudolph, "Local gain adaptation in stochastic gradient descent," in Proceedings of the 9th International Conference on Artificial Neural Networks, Edinburgh, Scotland, pp. 569-574, 1999.

[11] Y. Boykov and D. Huttenlocher, "Adaptive Bayesian recognition in tracking rigid objects," CVPR, 2000