Strathprints Institutional Repository

This version is available at http://strathprints.strath.ac.uk/40277/

Adaptive template matching algorithm based on SWAD for robust target tracking

G. Di Caterina and J.J. Soraghan

The Sum of Absolute Differences (SAD) is widely used in video coding and disparity computation for its simplicity. However SAD is not very common in tracking applications due to issues like partial occlusion and target changes, which can dramatically affect its performance. In this Letter we present a novel adaptive template matching algorithm for target tracking, based on a Sum of Weighted Absolute Differences (SWAD). The target template is updated using an infinite impulse response filter, while a weighting kernel is adopted to reduce the effects of partial occlusion. Simulation results demonstrate that the proposed tracker outperforms conventional SAD in terms of efficiency and accuracy, and its performance is comparable with more complex trackers, such as the Mean Shift algorithm.

Introduction: Visual tracking is the process of inferring information between detections of a selected target in consecutive frames, and it is an important task in many video processing systems. Many algorithms with medium-high complexity have been proposed in literature, as reported in [1]. As the majority of them deal with partial occlusion, we argue that, for the same level of tracking performance, simpler algorithms are preferable. The Sum of Absolute Difference is a well known metric and it is often used in video coding and disparity computation for stereo images, due to its simplicity. However SAD is not very common in tracking applications, as issues like partial occlusion and target template changes can dramatically affect its performance. In this Letter we present a novel adaptive template matching algorithm for robust target tracking based on a modified SAD called Sum of Weighted Absolute Differences (SWAD), where a Gaussian weighting kernel is employed to reduce the effects of partial occlusion. The target template is updated using an infinite impulse response filter approach, to implicitly deal with visual changes of the target, such as rescaling and rotation. Simulation results demonstrate that the proposed SWAD-based tracker outperforms conventional SAD in terms of efficiency and accuracy. The proposed algorithm is also compared with the widely used Mean Shift tracker [2], to show that its performance is comparable

to more complex algorithms. The simplicity and algorithmic structure of the SWAD-based tracker makes it suitable for real time implementations in embedded systems with fixed-point architecture.

Tracking Algorithm: The novel tracking algorithm presented in this Letter is based on template matching, and in particular on the minimization of a Sum of Weighted Absolute Differences (SWAD) in a region of interest (ROI) $\mathbf{R}_i$ within the current frame $\mathbf{F}_i$, as shown in Fig. 1b. The target model is represented by a template $\mathbf{T}_i$ of $N_T \times N_T$ pixels, and computed as:

$$\mathbf{T}_i = (1 - \alpha)\mathbf{T}_{i-1} + \alpha\hat{\mathbf{T}}_{i-1} \tag{1}$$

where $\alpha \in [0,1]$ is a blending factor, and $\mathbf{T}_{i-1}$ and $\hat{\mathbf{T}}_{i-1}$ are respectively target template and best match in the previous frame $\mathbf{F}_{i-1}$ (Fig. 1a). In the first frame $\mathbf{F}_0$ the target template $\mathbf{T}_0$ is initialized as the portion of $\mathbf{F}_0$ containing the selected target. The ROI $\mathbf{R}_i$ in position $\mathbf{q}_i$ is selected as the neighbouring area around $\hat{\mathbf{T}}_{i-1}$ in the previous frame $\mathbf{F}_{i-1}$, with size of $N_R \times N_R$ pixels, where $N_R = N_T + 2N_S$ and N_S is a spatial offset around $\hat{\mathbf{T}}_{i-1}$. The proposed algorithm requires single colour plane images, therefore it can be applied to both gray scale and colour video sequences. In the case of colour images, $\mathbf{F}_i$ can be the luminance component. The absolute differences in the SWAD metric are weighted by a kernel $\mathbf{K}$ of $N_T \times N_T$ pixels. In the current implementation a Gaussian weighting kernel has been employed, to assign high weights to central pixels and low weights to peripheral ones, as these pixels might belong to background or even occluding objects. Assuming integer pixel values in the range $[0, 255]$, the kernel $\mathbf{K}$ is:

$$K(x,y) = \left\lfloor 255 \times \frac{g(x,y)}{g(\lfloor \mu \rfloor, \lfloor \mu \rfloor)} \right\rfloor \tag{2}$$

where $x, y \in [0, N_T - 1]$ and $g(x,y)$ is 2-dimensional Gaussian function with mean $\mu = (N_T - 1)/2$ and standard deviation $\sigma = N_T/5$, defined as:

$$g(x,y) = \exp\left(-\frac{(x-\mu)^2}{2\sigma^2} - \frac{(y-\mu)^2}{2\sigma^2}\right) \tag{3}$$

Given the current template $\mathbf{T}_i$, its best match $\hat{\mathbf{T}}_i$ in $\mathbf{F}_i$ is searched within $\mathbf{R}_i$ (Fig. 1c). For this purpose the SWAD coefficient $\psi(x,y)$ at location (x,y) within $\mathbf{R}_i$ is computed as:

$$\psi(x,y) = \sum_{m=0}^{N_T-1} \sum_{n=0}^{N_T-1} K(m,n)\Delta(x,y,m,n) \tag{4}$$

where $\mathbf{K}$ is the Gaussian weighting kernel and $\Delta(x, y, m, n)$ is the pixel difference:

$$\Delta(x, y, m, n) = |R_i(x + m, y + n) - T_i(m, n)| \tag{5}$$

The position of the best match $\hat{\mathbf{T}}_i$ within $\mathbf{R}_i$ for the given template $\mathbf{T}_i$ is represented by the 2-dimensional index $\mathbf{b}_i$, obtained by minimizing the SWAD coefficient as in (6):

$$\mathbf{b}_i = \underset{x, y \in [0, N_R - N_T]}{\arg\max} \psi(x, y) \tag{6}$$

The index $\mathbf{b}_i$ refers to the top-left pixel of the best matching block $\hat{\mathbf{T}}_i$. The position of $\hat{\mathbf{T}}_i$ within $\mathbf{F}_i$ is computed as $\mathbf{p}_i = \mathbf{q}_i + \mathbf{b}_i$, as illustrated in Fig. 1c. Once the best match $\hat{\mathbf{T}}_i$ in the current frame has been found, the new target model $\mathbf{T}_{i+1}$ for the next frame $\mathbf{F}_{i+1}$ is computed using equation (1), which describes an infinite impulse response filter and it ensures that the target template is updated and adapts to changes, while guaranteeing tracker stability by reducing the effects of partial occlusion. Moreover, updating the template as in (1) implicitly deals with possible rescaling and rotation of the target. The value of the blending factor α mainly depends on the approach one wants to follow to update the target template. In a high frame rate implementation, α can be set to 0.5, giving equal weight to current and past templates, so that $\mathbf{T}_i$ can adapt to changes fairly quickly. For a more conservative approach, i.e. to preserve the template and let it adapt slowly, the blending factore can be $\alpha < 0.5$. Finally a new region of interest $\mathbf{R}_{i+1}$ for the next frame $\mathbf{F}_{i+1}$ is defined around the position $\mathbf{p}_i$ of $\hat{\mathbf{T}}_i$, within $\mathbf{F}_i$ (Fig. 1d). In particular the position $\mathbf{q}_{i+1}$ of $\mathbf{R}_{i+1}$ in $\mathbf{F}_{i+1}$ is computed as:

$$q_{i+1}(1) = \min\left(\max\left(p_i(1) - N_S, 1\right), W - N_R\right)$$
$$q_{i+1}(2) = \min\left(\max\left(p_i(2) - N_S, 1\right), H - N_R\right) \tag{7}$$

where W and H are respectively width and height of the frame. Equation (7) ensures that $\mathbf{R}_{i+1}$ is entirely within the frame $\mathbf{F}_{i+1}$. The steps of the proposed tracking algorithm for each frame $\mathbf{F}_i$ are summarized as follows:

1. Given $\mathbf{R}_i$ and $\mathbf{T}_i$, minimize the SWAD coefficient $\psi(x, y)$ as in (4)-(6), to obtain the position $\mathbf{b}_i$ of the best match $\hat{\mathbf{T}}_i$ in $\mathbf{F}_i$.

2. Given $\mathbf{b}_i$ and the position $\mathbf{q}_i$ of $\mathbf{R}_i$ within $\mathbf{F}_i$, compute the target position $\mathbf{p}_i$.

3. Given $\mathbf{T}_i$ and the best match $\hat{\mathbf{T}}_i$, update the target model using (1).

4. Compute the position $\mathbf{q}_{i+1}$ of the ROI $\mathbf{R}_{i+1}$ in the next frame $\mathbf{F}_{i+1}$ from (7).

Evaluation: A comparison between Matlab implementations of the adaptive template matching algorithm with both SWAD and SAD has been carried out to assess relative tracking capability. Visual results for the PETS2006 video sequence S1-3 are shown in Fig. 2, where the first and second rows refer respectively to SAD and SWAD. Both trackers are initialized in frame #968, on the person with the backpack entering the scene from the right, as shown in Fig. 2a and Fig. 2d. It can be seen that the SAD tracker begins to lose the target in frame #1087 (Fig. 2b) and loses track at #1117 (Fig. 2c), while the SWAD algorithm keeps tracking the target (Fig. 2e-f). The SAD-based tracker is obviously confused with the tile structure on the background.

The SWAD tracker was also compared against the Mean Shift tracker on the PETS2007 sequence S06-1, with the selected target being the blonde woman in red clothing entering the scene from the top. From Fig. 3 it can be seen that the error distances from the manually labelled ground truth in both trackers have roughly the same behaviour. In this experiment, the error associated with MS has mean value $\mu_e = 12.57$ pixels and variance $\sigma_e = 6.23$ pixels, while the error associated with SWAD has mean value $\mu_e = 9.47$ pixels and variance $\sigma_e = 5.69$ pixels. The error with MS is higher than the error with SWAD in 88% of the frames, while the error with SWAD is higher in 12% of the frames. This experiment shows that the performance of the SWAD-based tracker is comparable to the more complex Mean Shift tracker approach.

Finally, the proposed tracking algorithm has been implemented on a DM6437 Evaluation Module [3], equipped with a fixed-point DSP from Texas Instruments. The SWAD algorithm takes advantage of the fixed-point architecture of such DSP and it runs at 16 ms per frame, largely meeting the real-time requirements of an embedded application. The work in [4] presents an optimized implementation on the same architecture of the Camshift face tracker, which is based on the Mean Shift tracker. Such implementation on the DM6437 runs at about 120 ms per frames, which is almost an order of magnitude slower than the adaptive template matching algorithm proposed in this Letter.

Conclusion: This Letter has presented a novel adaptive template matching algorithm

4

for tracking, based on the minimization of the Sum of Weighted Absolute Differences. A Gaussian weighting kernel is used to assign different weights to pixels distant from the target centroid, as these pixels may experience partial occlusion. The target template is updated using an infinite impulse response filter to adapt to template changes, such as rescaling and rotation. Despite its simplicity, experimental results have shown the robustness of the tracker and its higher accuracy with respect to SAD-based tracking. Moreover, the tracking performance of the proposed algorithm is comparable with the ones of more complex trackers. The SWAD-based tracker is ideal for real-time implementations on devices with low computational capabilities, as in the case of fixed-point embedded DSP platforms.

References

[1] Yilmaz, A., Javed, O., and Shah, M.: 'Object tracking: a survey'. ACM Computing Surveys, 2006, Vol. 38 (4), pp. 1–45

[2] Comaniciu, D., Ramesh, V., and Meer, P.: 'Kernel-based object tracking'. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2003, Vol. 25 (5), pp. 564–577

[3] Spectrum Digital Inc.: 'TMS320DM6437 Evaluation Module Technical Reference', 2006

[4] Varfolomieiev, A., Antonyuk, O., and Lysenko, O.: 'Camshift object tracking algorithm implementation on DM6437 EVM'. European DSP Education & Research Conference, 2010, pp. 96–100

Authors' affiliation

G. Di Caterina and J.J. Soraghan (Centre for excellence in Signal & Image Processing, Department of Electronic and Electrical Engineering, Strathclyde University, 204 George Street, Glasgow, G1 1XW, UK)
Email: gaetano.di-caterina@strath.ac.uk

Figure captions:

Fig. 1: Layout of the current frame $\mathbf{F}_i$ showing the best match $\hat{\mathbf{T}}_i$, the region of interest $\mathbf{R}_i$ and their positions within $\mathbf{F}_i$.

Fig. 2: Comparison between SAD and SWAD: the first row refers to SAD tracking, while the second row refers to SWAD tracking, where SWAD maintains high accuracy in tracking, especially for frame #1087 and frame #1117.

Fig. 3: Error distance from ground truth for MS and SWAD trackers.

Figure 1

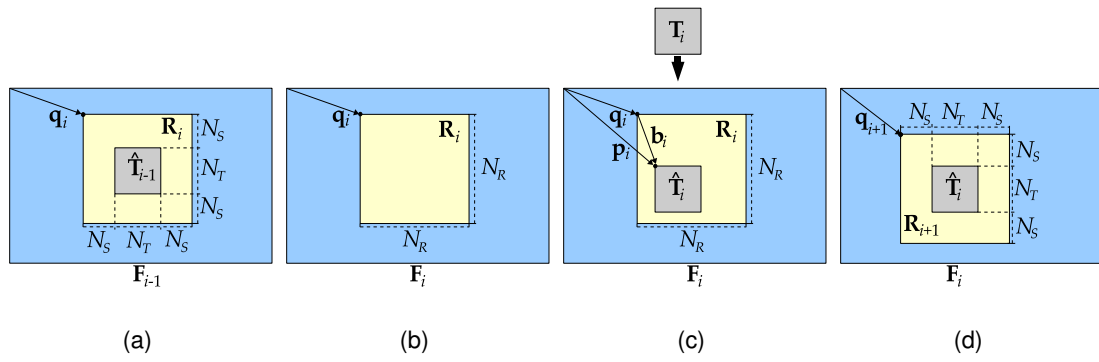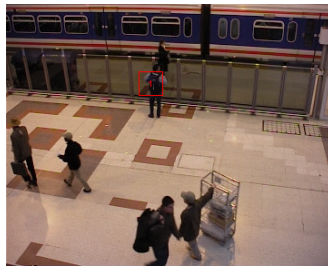(a) SAD - #968   (b) SAD - #1087   (c) SAD - #1117

(d) SWAD - #968   (e) SWAD - #1087   (f) SWAD - #1117

Figure 3