Track validation using gradient-based normalised cross-correlation

Darren Caulfield
Darren.Caulfield@cs.tcd.ie
Kenneth Dawson-Howe
Kenneth.Dawson-Howe@cs.tcd.ie

GV2 Group
School of Computer Science
Trinity College Dublin
Ireland

1 Gradient-based normalised cross-correlation

Normalised cross-correlation (NCC) is often used as the similarity measure in template matching trackers. However, NCC, when used with a brute-force search strategy, is computationally expensive. In this paper we develop a gradient ascent version of NCC; it retains the robustness of its brute-force counterpart, but it executes much more quickly. Indeed, its speed is comparable to that of another data-driven tracker – the mean-shift method – but it is much less likely to lose track of its target.

We begin our derivation of the gradient-based NCC tracker by defining certain entities, following the image sequence notation of Hager and Belhumeur [2]. Let $I(\mathbf{x},t)$ be the image at time t, where $\mathbf{x} \triangleq (x,y)$ is a point in the image. Let $Q \triangleq I(\mathbf{x},0)$ be the target (image template) we wish to track, and let $I \triangleq I(\mathbf{x}+\mathbf{u},t)$ be a candidate image region in the current frame. The similarity $O(\mathbf{u})$ of a template Q and an image patch I displaced from the template by $\mathbf{u}$ is:

$$O(\mathbf{u}) \triangleq \frac{1}{n}\sum_{\mathbf{x}\in R}(I-\bar{I})(Q-\bar{Q}) = \frac{1}{n}\sum_{\mathbf{x}\in R} I(Q-\bar{Q}) \tag{1}$$

where n is the number of pixels in the image patch and R is the set of pixel locations in the template. The above formula is the same as that for normalised cross-correlation, except that we do not divide by the standard deviations of Q and I. Furthermore, equation 1 applies to single-channel images; for multiple-channel data we simply sum the contributions from the individual channels.

We can approximate I by the low-order terms of its Taylor series around $(\mathbf{x},t)$:

$$I = I(\mathbf{x}+\mathbf{u},t) \approx I(\mathbf{x},t) + u_1 I_x(\mathbf{x},t) + u_2 I_y(\mathbf{x},t) + (t-t)\times I_t(\mathbf{x},t) \tag{2}$$

where $\mathbf{u} \triangleq (u_1,u_2)$. The expressions I_x, I_y and I_t are the spatial and (unneeded) temporal image derivatives, respectively. We use image differences to approximate the required derivatives:

$$I_x(\mathbf{x},t) \triangleq I(\mathbf{x},t) - I\Big((x+1,y),t\Big); \qquad I_y(\mathbf{x},t) \triangleq I(\mathbf{x},t) - I\Big((x,y+1),t\Big) \tag{3}$$

Simplifying equation 2, we obtain:

$$I(\mathbf{x}+\mathbf{u},t) \approx I(\mathbf{x},t) + u_1 I_x(\mathbf{x},t) + u_2 I_y(\mathbf{x},t) \tag{4}$$

Our goal is to find a local maximum of the similarity measure O as we vary the displacement $\mathbf{u}$; therefore, we differentiate O with respect to $\mathbf{u} = (u_1,u_2)$:

$$\frac{\partial O}{\partial u_1} \approx \frac{1}{n}\sum_{\mathbf{x}\in R} I_x(\mathbf{x},t)(Q-\bar{Q}); \qquad \frac{\partial O}{\partial u_2} \approx \frac{1}{n}\sum_{\mathbf{x}\in R} I_y(\mathbf{x},t)(Q-\bar{Q}) \tag{5}$$

At each iteration we move the tracker to the neighbouring (integer) pixel location "pointed to" by the gradient $\nabla(O)$:

$$\nabla(O) = \left(\frac{\partial O}{\partial u_1}, \frac{\partial O}{\partial u_2}\right) \tag{6}$$

We terminate the iterations as soon as a move results in a decrease in the similarity measure. (The last tracker move, which led to the decrease, is also reversed.)

We have used 21 short video sequences – from the CAVIAR and PETS datasets – in order to perform a quantitative evaluation of the gradient-based and brute-force NCC trackers. At each frame, we determine the target's scale by specifying the horizon line in the scene and exploiting the effects of perspective [1, pp. 57–60].

We assess a tracker's *robustness* by counting the number of sequences in which it successfully follows its target. A *lost track* is recorded whenever the overlap between the tracker's rectangle and the object's ground truth bounding box falls below 10% of the area of the latter [3].

The gradient-based NCC tracker lost track of its target in 3 of the sequences, while the brute-force NCC tracker recorded 4 lost targets. For comparison, mean shift loses track on 9 of the sequences.

Our gradient-based NCC tracker is, on average, 4.8 times faster than the brute-force version, while being slightly more robust. It is also found that the mean-shift approach, which is much less robust than the other methods, is only 20% faster, on average, than our technique.

2 Track validation

In real-world systems, it is often important to know when tracking has failed so that we may take corrective action. Ideally, such notifications will not rely on ground truth data or human monitoring of the system. We propose the alternative approach of using our gradient-based NCC technique as the basis of a *track validation* algorithm. By tracking a target forwards in time through the sequence, reinitialising the tracker with a new model taken from the end of the video, and following the target backwards in time, we can judge, in the absence of ground truth data, whether or not the tracking was successful: a large divergence between the forwards and backwards trajectories (*failed validation*) indicates that the object was lost by the tracker at some point. In such a situation the algorithm iteratively attempts to validate shorter subsequences of the video. A successful validation allows the method to switch to a new model of the target, which it uses in an effort to validate the object's trajectory in the remainder of the sequence.

Figure 1: (Left) An instance of successful track validation (using the gradient-based NCC tracker). (Right) An instance where tracking is not validated (using the mean-shift tracker). The forwards trajectory (red) and the backwards trajectory (green) should match very closely for validation to occur.

By switching models in this way, track validation allows us to follow objects whose appearance changes drastically over time. Even when our algorithm is unable to track an object for the full length of a video clip, it is providing valuable information to the higher-level processes that invoked it. Specifically, a failure of track validation indicates that tracking has become very difficult in that particular section of the video, and that other techniques and strategies should be considered.

[1] D. Caulfield. *Mean-Shift Tracking for Surveillance: Evaluations and Enhancements*. PhD thesis, University of Dublin, 2011.

[2] G. D. Hager and P. N. Belhumeur. Real-time tracking of image regions with changes in geometry and illumination. In *Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition*, pages 403–410. IEEE Computer Society, 1996.

[3] J.C. Nascimento and J.S. Marques. Performance evaluation of object detection algorithms for video surveillance. *Multimedia, IEEE Transactions on*, 8(4):761–774, Aug. 2006. ISSN 1520-9210. doi: 10.1109/TMM.2006.876287.