People Tracking using Robust Motion Detection and Estimation

Markus Latzel, Emilie Darcourt, and John K. Tsotsos
Department of Computer Science, Center For Vision Research
York University
Toronto, Ontario, Canada
{markus,emilie,tsotsos}@cs.yorku.ca

Abstract

Real world computer vision systems highly depend on reliable, robust retrieval of motion cues to make accurate decisions about their surroundings. In this paper, we present a simple, yet high performance low-level filter for motion tracking in digitized video signals. The algorithm is based on constant characteristics of a common, 2-frame interlaced video signal, yet results presented in this paper show its applicability to highly compressed, noisy image sequences as well. In general, our approach uses a computationally low-cost solution to define the area of interest for tracking of multiple, moving objects. Despite its simplicity, it compares very well to exisiting approaches due to its robustness towards environmental changes. To demonstrate this, we present results of processing a sequence of JPEG-compressed monocular images of a parking lot in order to track pedestrians, cars and bicycles. Despite a high level of noise and changing lighting conditions, the algorithm successfully segments a moving object and tracks its position along a trajectory.

Keywords: Interlace Filter, Motion Tracking, Motion Detection, Surveillance

1. Introduction

Motion detection and estimation in image sequences has multiple applications ranging from image stream data compression to artificial intelligence or automatic surveillance problems and is considered a vital component of any of these systems. Accordingly, a large amount of work has been done to identify moving objects within a video sequence and estimate their motion parameters. In the following, a novel approach for this task is presented, for the reason of its astonishing simplicity and robust performance, as shown in experimental applications.

Motion in a sequence of images is defined as a situation where part of the scene is moving in front of a non-uniform background. This moving part may not be connected, as in the case of multiple moving objects, each drifting in different directions. Furthermore, additional to a simple translation of static image data, the objects may be rotating, non-rigidly deforming (i.e., experience local motion within themselves) or temporarily occlude each other. On top of this, the capturing device may be submitted to ego-motion, which results in a moving background altogether.

A motion estimation algorithm identifies connected objects within the image that experience motion relative towards the background, and estimate motion parameters, i.e., direction, speed, and occlusion events. In partial solutions, a simple detection of ”something in motion” suffices. In many motion alarm systems for surveillance applications, this is the case.

1.1. Background Subtraction

A trivial motion detection works by subtracting a previously stored ”background” image, or *reference frame* from the current input, and identify areas of high response as disturbances in the image. Obviously, background subtraction relies on zero ego-motion of the system, and no undesired changes to the observed background. Lighting changes for example, have to be accommodated for by an adaptive background extraction algorithm, which iteratively updates the stored reference frame. In [7], A. Makarov compares background extraction algorithms. Common to his paper and [10] is that the choice of threshold values for discrimination between noise and real changes to the scene is crucial to eliminate false positives during motion detection. In [1], an automatic reference frame update is given to be able to detect motion even after sudden lighting changes occur in an image sequence.

1.2. Interlaced Video Signals

A notion generally left out of consideration is that motion (as temporal changes) is inherently encoded in the im-

IEEE COMPUTER SOCIETY

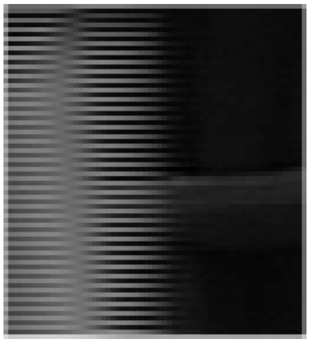


Figure 1. Motion artifacts in interlaced video signals.

age sequence signal acquired from commercial video cameras. The NTSC signal scans 525 lines of image data per frame, with a frame rate of 30fps. To eliminate flicker, two interlaced frames are sent alternatively with a rate of 60fps. Accordingly, alternating lines of the captured image are scanned with a delay of $16.7ms$. If we consider an object moving horizontally before a background, then *interlacing artefacts* can be observed along edges of the object, since each even scan line has been captured exactly one frame later than its preceding and following odd line. Figure 1 shows a magnified snapshot of such an artefact. The horizontal extent of these artefacts are directly proportional to the horizontal component of the edge velocity. In terms of the image window, the objects velocity v_o is thus

$$v_o = \frac{l}{16.7ms}$$

in pixel per second, with l being the length of an interlacing artefact.

2. A Motion Bandpass Filter

Detecting the interlace artefacts can be achieved by applying a vertically oriented bandpass filter with the characteristic frequency

$$f_{BP} = 1\frac{1}{pixel}$$

Since none of the original image should be preserved, the filter thus has a low frequency component of 0, i.e. weight of the filter kernel $k_{BP} = 0$. In order to discriminate weight of scan lines further from the center of the filter kernel, a standard band pass filter was multiplied with an approximated Hamming window function:

$$\begin{aligned} k_{BP} &= k_{stdrd} * k_w \\ &= [\ -1 \quad 1 \quad -1 \quad 1 \quad -1 \quad 1 \quad -1\ ] \\ &*[\ 1 \quad 2 \quad 3 \quad 4 \quad 3 \quad 2 \quad 1\ ]^T \\ &= [\ -1 \quad 2 \quad -3 \quad 4 \quad -3 \quad 2 \quad -1\ ] \end{aligned}$$

With the windowing function k_w approximated by a triangle function.

Applied to a raw video image $I(x, y)$, the given filter responds to columns of pixels with alternating intensities. In particular, if the centre of the kernel is located on a white pixel within a motion artefact, the response will be positive, while on a black pixel, the response is negative. In areas with a frequency other than f_{BP}, the output $I_{BP}(x, y)$ is close to 0.

Theoretically, the filter responds to any high frequency along a vertical axis such as sharp, horizontal edges, for example. Increasing the filter length dy will increase its discrimination towards such 'false responses', and ensure only truely alternating lines produce a response. However, if slanted lines moving with slow velocity produce an artefact length of less than $dy/2$, sensitivity of the filter is reduced. Experimentally, we found that a filter length $dy = 7$ was optimal for the motion detection applications presented in the following.

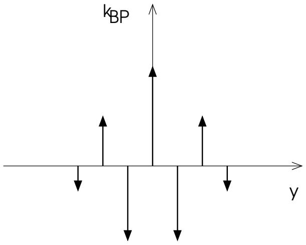


Figure 2. A possible bandpass filter for k_{BP}.

To decide whether moving edges are present in an image or not, the complete filter takes the absolute value of the output and generates a threshold image thereof:

$$M(x, y) = \begin{cases} 1 & \text{if } |I_{BP}(x, y)| > \text{threshold} \\ 0 & \text{else} \end{cases} \qquad (1)$$

In an experimental application, we were able to use the filter for a simple intrusion detection system with this method. An intrusion into the field of view was detected if

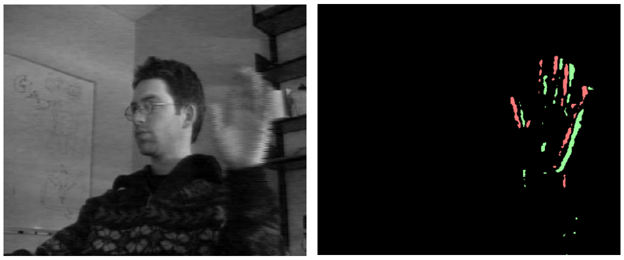
$$\sum_{x,y} M(x, y) > k,$$

with k being a sensitivity parameter, depending on dimensions of the acquired image and expected size of a moving object. A larger object will generally produce longer edges and thus increase positives in M.

Since this algorithm does not store any reference data, it does not react with false positives towards gradual changes in the image. In other words, very slow and global changes do not trigger the bandpass filter at all. As a result, lighting changes do not result in any false intrusion alarms.

2.1. Thresholding

Selecting the threshold to attain $M(x, y)$ proved to be quite robust. In fact for both the indoor and outdoor images shown, the same threshold t was used. However, a simple threshold adjustment scheme was used to automatically increase the threshold on a static image input until the $M(x, y) = 0$ for all x, y. The computed threshold was afterwards treated as a system dependent variable and needed no further updating. With k_{BP} as given in figure 2, and an input value range of $[0..255]$ for I, I_{BP} can assume values within $[-8 \cdot 255..8 \cdot 255]$. The histograms in figure 4 show a recording of $|I_{BP}|$ over this period, with the computed threshold marked. Note that for the still reference image, no values above t exist (i.e. $M(x, y) = 0 \forall x, y$). Again, the histogram does not change significantly for extreme changes in lighting conditions (indoor/outdoor).

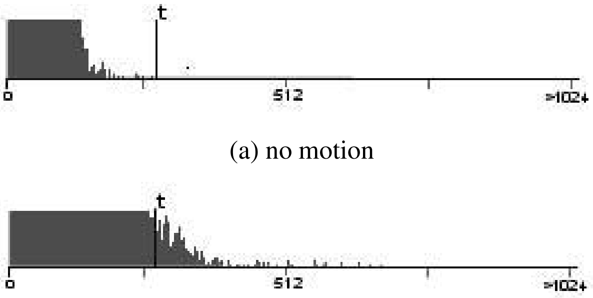


(a) raw image (b) $|I_{BP}|$

Figure 3. Raw image data and filtered output of a hand in waving motion. Note the low contrast of the acquired image

3. Motion Parameter Estimation

In order to provide information for higher level computer vision systems, it is necessary to estimate motion parameters such as location, velocity and direction of moving objects from the filter output. Below, an outline of this is achieved and some consideration on issues are given:



(a) no motion

(b) waving motion

Figure 4. Value histogram of $|I_{BP}|$. (b) was captured with a waving motion as in figure 3.

3.1. Direction of motion

In order to utilise the polarisation property of the filter, the output signal I_{BP} was further processed to display this information: a three colour image $N(x, y)$ was defined as:

$$N(x, y) = M(x, y) \cdot sign(I_{BP}(x, y))(-1)^y,$$

with y being the scan line number. Figure 3 shows $N(x, y)$ encoded in three different colours, black for $M(x, y) = 0$.

$N(x, y)$ denotes a gradient function of image values: a transition from lower intensity to a higher one results in a positive value. Thus, the direction of motion can be computed from information derived from a single frame.

3.2. Motion velocity

Motion velocity $v_{o,h}$ along the horizontal trajectory can be estimated from a single image snapshot by measuring the artefact length l, as mentioned earlier. However, a vertical motion will also yield filter response at slanted and horizontal edges of the object. Thus, any object motion yields an edge response of the filter, but the exact component of vertical motion can not be estimated with this algorithm unless a more sophisticated approach is taken, such as operating on a vertically interlaced input image. However, real time performance of the simple horizontal motion estimation suggests that the given approach will assist a higher level decision system to in image processing, identified frames and areas of interest, with a preliminary motion velocity estimation algorithm.

Also, given the high temporal resolution of the motion filter approach (an inter-frame approach may - with appropriate hardware - at best analyse motion in intervals of 33.3ms), *fast motion* is easy to detect and estimate due to the scan line interval of 16.7ms. Thus, for example in a traffic surveillance application, speeding drivers are captured even if they appear only in a single frame of the image sequence. In contrary, *very slow* motion produces a dimmed output of the filter. This drawback can be accommodated by periodically *interlacing* previously stored key frames with the current image. In other words, to virtually triple the speed of passing objects, a composite frame $J(x, y, n)$ consists of interlaced frames $I(n)$, and $I(n-1)$, with n being the number of the current frame:

$$J(x, y, n) = \begin{cases} I(x, y, n) & \text{, if } y \text{ even} \\ I(x, y, n-1) & \text{else} \end{cases}$$

Applying the bandpass to J amplifies the sensitivity towards slow motion.

4. Application to a sample dataset

In order to confirm our finding, we demonstrated application of the motion detector to a pre-recorded sequence of images. The dataset available was actually a sequence of highly compressed images with view of pedestrians, cars or bicycles in a parking lot scenario. Below find some considerations on implications of this constaint with respect to application of our motion detection filter.

4.1. Considerations on Data Properties

In general, the high frequency components inherent to the motion artefacts described above are not preserved during compression of video data. Thus, the interlace detection filter cannot be applied directly on the acquired images. Rather, two subsequent images in a sequence had to be interlaced artificially to re-create the same properties as discussed in section 3.2.

However, interlacing compressed images bears the disadvantage of added noise due to spatial inconsistencies of the particular compression scheme used for each individual image of the sequence. The common 'bit-flicker' seen in any digitized video signal is amplified as a result of this process. Nonetheless, the motion filter showed reliable responses even for small moving objects as shown in figure 5.

4.2. Tracking Multiple Targets

Multiple target tracking relies on robust clustering of motion cues relevant to each moving object and sufficient supression of noise. Binary output of the motion filter shows that responses for one moving object are not necessarily connected. One approach to increase sub-component size and thus try to merge into one component would be to decrease the filter threshold to enlargen the response areas. However, experiments with this approach yielded nonsatisfactory results in that noise was not surpressed sufficiently and moving objects merged when the distance between them was too small. A dilation-erosion process showed similar results with the additional drawback of loss of detail around the target boundary.

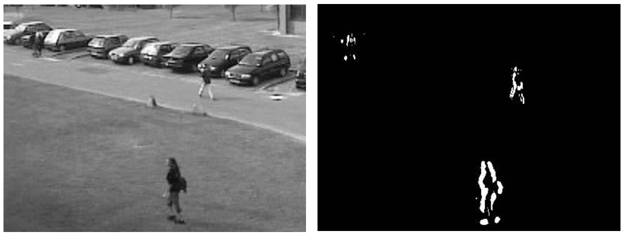

(a) raw image (b) $|I_{BP}|$

Figure 5. Raw image data and filtered output of a frame in the PETS sequence

To deal with this problem, we chose to employ a *modified component labeling algorithm*. The component labeling algorithm assigns labels to 1-pixels connected directly to each other and thus clusters connected pixels within one group. In our approach, we increased the distance pixels may be apart by introducing a neighbourhood size D. Each 1-pixel was regarded of the same connected component to a 1-pixel, if it was located within the neighbourhood distance D. Recursively, the algorithm was then applied to its respective neighbours within distance D. This approach successfully assigned filter responses a grouping label and sufficed for attaining a centre of gravity for each moving object as well as a bounding-box segmentation.

Noise produced by the motion filter typically are stray groups with a size significantly smaller than the smallest real target and could thus be culled by restricting moving objects to a minimum number of response pixels. In practice, this process culled any noise specks for the entire sequence of the given video footage, while preserving all moving objects.

Object Outline After grouping filter responses and culling small groups, a coarse outline for each object is calculated to serve for motion segmentation and attentional purposes. The outlines shown in figure 6 are defined as an n-set of points defined by (ρ, φ), denoting radius and angle from the centre of gravity $c = (c_x, c_y)^T$ of each particular object. The values of (ρ, φ) are defined as:

$$r_{max}(\varphi_k) = \max r : pixel(r, \varphi_k) = 1$$
$$k = (0, 1, ..n)$$
$$\rho(\varphi_k) = \frac{1}{3}(r_{max}(\varphi_{k-1}) + r_{max}(\varphi_k) + r_{max}(\varphi_{k+1}))$$

$pixel(r, \varphi)$ being the input image value given in polar coordinates from origin c. $(\varphi_0, ...\varphi_n)$ spans $-\pi$ to π in equidistant steps. Thus, for each angle k there is a distance to a 1-pixel furthest from the centre of gravity. In order to achieve a smooth outline, this distance is averaged over three neighbouring radius values. Figure 6 shows the result of this outline definition. Note that for outline points close to c the outline is fairly accurate, while for extensions of the object further away from the centre the averaging effect compromises accuracy. Accordingly, legs and arms of walking people are more likely to inaccurate segmentation.

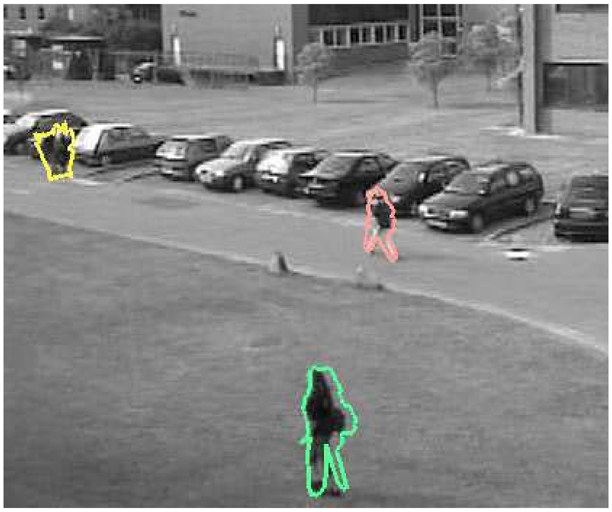


Figure 6. The outline $\rho_{1..n}$ shown as connected points. Distinct objects are marked with different colours.

Consistent Target Tracking In order to track moving objects as individual instances, a state of each filter response cluster is preserved from the last frame in the video sequence. The state vector encompasses centre of gravity in two-dimensional space c, the set of outline points $\rho_1...\rho_n$, and a bounding box around all response pixels of the object.

For each frame, the centre of gravity is used as starting point for the modified component labelling algorithm described above. This ensures that component clustering does not originate at noise clusters that were within boundaries of a moving object in the previous frame. Our experiments on the described dataset showed very good results using this technique.

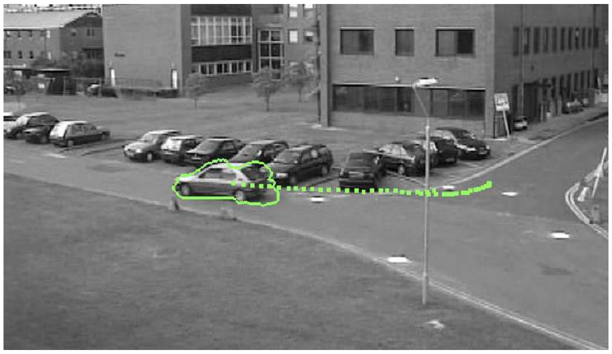


Figure 7. Sample application of the segmentation and tracking algorithm

5. Results and Conclusion

The motion detection filter described in this paper was applied to the common problem of tracking multiple objects in a surveillance scenario. The proposed filter showed to be robust with respect to lighting conditions, changes in the environment and discontinuous motion parameters. Additionally, an effective outline computation based on a circular sweep algorithm was presented that accurately segments walking pedestrians and cars. We propose that in particular, the robustness of the interlaced motion filter shows benefits in applications of changing global conditions.

In some cases, large objects outside of the typical range of interest (such as a pedestrian walking up to the camera) were segmented as several objects moving in unison. Also, since the motion tracker disengaged from a non-moving object, the identity of an object was lost once it stopped.

Our future work will include a memory-based tracker that will keep identity objects while not in motion, as well as build super-clusters of objects that are in unified motion.

Acknowledgements: Funding of the work was gratefully received from Communication Information Technology Ontario (CITO).

References

[1] E. Durucan, F. Ziliani, O.N. Gerek, *Change Detection with Automatic Reference Frame Update and Key Frame Detector*, in Proceedings of the IEEE-Eurasip Workshop on Nonlinear Signal and Image Processing (NSIP'99), vol. 1, pp. 57-60, Antalya, Turkey, June 20-23, 1999

[2] A. Francois, G. G. Medioni, *Adaptive Color Background Modeling for Real-Time Segmentation of Video Streams*, In Proc. Int. Conf. on Imaging Science, Systems, and Technology, pp. 227-232, Las Vegas, NA, June 1999

[3] B. Jähne, *Digital Image Processing*, Springer, Berlin, 1997

[4] A.K. Jain, *Fundamentals of Image Processing*, Prentice-Hall, 1989

[5] K.P. Karmann, A. Brandt, R. Gerl,*Moving Object Segmentaion Based on Adaptive Reference Images*, Proc. 5th European Signal Processing Conference, pp.951-954, Barcelona, 1992

[6] M. Latzel, J.K. Tsotsos, *A Robust Motion Detection and Estimation Filter for Video Signals*, Proc. International Conference on Image Processing ICIP 2001, IEEE Press, 2001

[7] B. Makarov, *Comparison of Background Extraction Based Intrusion Detection Algorithms*, IEEE International Conference on Image Processing (ICIP'96) pp. 521-524, 1996

[8] B. G. Schunck, *The Image Flow Constraint Equation*, Computer Vision, Grapihcs, and Image Processing, 35:20-46, 1986

[9] D. Toth, T. Aach, V. Metzler, *Illumination-Invariant Change Detection*, Southwest Symposium on Image Analysis and Interpretation April 2-4, 2000, Austin

[10] F. Ziliani, A. Cavallaro, *Image Analysis for Video Surveillance Based on Spatial Regularization of a Statistical Model-Based Change Detection*, ICIAP'99, pp1108-1110, Venezia, 1999