Pixel Behaviour Metrics for Dynamic Background Modelling with the Projected Difference Pattern Method

Peter Pakulski[1], Karl Sammut[1], Fangpo He[1], Matthew Naylor[2]
[1]*School of Informatics & Engineering, Flinders University, Bedford Park, SA 5001*
[2]*Vision Systems, Vision Fire & Security, Technology Park, Mawson Lakes, SA 5095*
[1]{*peter.pakulski| karl.sammut | fangpo.he@flinders.edu.au*}, [2]{*matthew.naylor@adpro.com.au*}

Abstract

This paper presents a new algorithm for modelling the behaviour of dynamic video. The PDP (Projected Difference Pattern) is designed to perform simple spatiotemporal processing with a strong focus on efficiency of real-time implementation. In its simplest implementation, the algorithm is shown to be suitable for generating dynamic background models, motion characterisation, and motion detection. Sample analyses of test video are presented to support this work.

1. Introduction

Most existing techniques employed in video processing are either based on static image processing methodologies or on single-pixel stochastic modelling methodologies. More recently, spatiotemporal modelling methods have been reported in the literature [2-7]. A spatiotemporal model simultaneously accounts for both the appearance of a given pixel relative to its surroundings, and its behaviour over time. By the consideration of such information within a single framework, spatiotemporal approaches and algorithms are able to make fuller use of the inherent coherence of motion in video [8, 9] – something which image-based and pixel-based methods cannot do. There is incentive to consider spatiotemporal approaches over spatial or temporal approaches alone: Approaches based on static image processing cannot meaningfully account for localised changes without the aid of temporal information. Pixel-based models, conversely, cannot handle global changes without spatial information.

Many video processing tasks require both spatial and temporal information to work upon, which is the rationale behind the development of the Projected Difference Pattern (PDP). The PDP by design offers metrics on local motion coherence and scene behaviour at every pixel. These metrics can be used to accomplish a number of result-driven goals.

Background modelling is a popular method of detecting moving objects in video footage. The underlying principle is that if the behaviour of the background is known and can be monitored, then any area exhibiting new or unusual behaviour can be assumed to be foreground. This task is almost trivial in controlled environments where there is little image change, or well-known image change, but becomes considerably more complex where the video contains dynamic background elements [2] as in the case of outdoor video scenes. Modelling the behaviour of outdoor video is used as the application example throughout this paper.

There is a significant body of work on modelling video background as an evolving static image. In this approach, large scale changes (especially those in lighting) are easily detected and accomodated. Parts of a representative short-term model are updated periodically to reflect the current appearance of the scene [10]. It is this class of models, which is most appropriate to video with large static regions. Such approaches are not well suited to video with continuous or frequent variations [2]. Sequences containing dynamic elements such as smoke, steam, fire, or rippling water – elements represented in the MIT Temporal Texture Database [1] - cannot be modelled well.

Another popular approach is that of pixel-based algorithms [11, 12]. These approaches take advantage of the relative maturity of one-dimensional signal processing techniques by considering the changes occurring at each pixel as a function of time. Many of these approaches have been shown to be very effective in certain applications, notably in background detection, but by design they are poorly suited to sequences that exhibit large-scale changes. Temporal changes such as a change in illumination (very common in outdoor scenes) can be accounted for, but each pixel models the change independently. Spatial changes - such as of an object

Proceedings of the Digital Imaging Computing: Techniques and Applications (DICTA 2005)
0-7695-2467-2/05 $20.00 © 2005 IEEE

moving through a scene – are similarly modelled independently. In both cases, potentially useful information is lost, and there is a performance cost in detecting the same changes multiple times.

While static image approaches are based on a two-dimensional spatial process, and stochastic pixel-based methods make use of one-dimensional temporal signal analysis, spatiotemporal analysis exploits the three-dimensional relationship between each pixel and its surrounding neighbours in adjacent frames.

Recent work by Doretto *et al*. in Dynamic Textures [6] has allowed them to not only segment scenes based on consistent spatiotemporal properties [5], but also to go as far as synthesising new footage with different properties to the original [7]. Their algorithms however require pre-processing of the entire target sequence, and have significant computational requirements. The heart of the approach lies in a cross-correlation of the entire scene to see how the value of each pixel affects every other. Though accurate to the point of photorealism in highly self-correlated sequences, these algorithms are not well adapted to tasks such as background modelling. The main reason for this is that the training phase produces a model that is not easily updated or changed after being established.

The PDP on the other hand can be updated on a frame-by-frame basis. Referring back to the example task of background subtraction: the PDP allows an approach which detects changes in the *behaviour* of the background. By ignoring appearance while preserving behaviour, the algorithm becomes largely illumination-independent, but will still detect, for example a torch beam by the sharp shadow edge being cast.

The size of the neighbourhood for consideration in the PDP is parameterised, as is the length of the behaviour history. In the implementation presented in this paper, the PDP presents the following metrics: Local optical flow (both direction and magnitude), confidence of the optical flow metric, and a measure of how consistent the optical flow has been over the last N frames (a support measure of the behaviour). Efficiency of implementation is an important consideration, with a view to running the algorithm in real-time for simple applications.

Section 2 of this paper outlines the concepts that form the basis of the PDP algorithm, and establishes the parameters. Section 3 describes the operation of the algorithm. Section 4 establishes the value of the PDP as a descriptive model by demonstrating video compression and synthesis. Section 5 of this paper contains some sample analyses of footage using the PDP, and Section 6 covers some simple extensions with a view to the possible applications of the PDP.

2. Related Concepts

The PDP is an original algorithm, with properties selected to best extract data from "difficult" footage. With outdoor footage a target of concern, the algorithm specifically needs to be tolerant of overlaid information, noise, warping, illumination changes, and poor contrast. The performance needs to be general and robust.

To have practical application in the chosen example of background modelling, the algorithm must either predict future frames of a sequence (for comparison with actual incoming data), or output a measure of how well the incoming data matches the previous frames, i.e., detect changes.

The origin of the PDP approach is a simplified model of the Human Visual System (HVS) [13, 14], which is a notably good performer at many vision tasks. There has been much work into emulating the HVS in neural networks (e.g. [14]), but little has been accomplished with practical application to video processing. The core of the processing performed by the PDP is analogous to the task performed by the first few neuron layers of the human visual cortex for the detection of linear motion [13].

The core process within the PDP is similarly akin to a simple optical flow algorithm. Whereas many more-advanced optical flow algorithms take care not to assume perfect translation between frames, the core of the PDP does make this assumption in order to highlight discontinuities; the validity of the assumption can be measured, to form one of the algorithm outputs. A major concern in optical flow algorithms is the merging of flow data into a coherent optical flow field [15, 16]. This is an unnecessary computational step in the processing done by the PDP unless the optical flow is a desired result – instead, a measure of optical flow consistency is given.

The relationship of this work to Dynamic Textures [5-7] is limited to the important concept of internal correlation. The work of Doretto *et al*. establishes the limits of what can be synthesised, by performing full autocorrelation across each pixel in the sequence. A model of the appearance of the current frame allows successive frames to be generated based on the correlation data. While Dynamic Textures looks for correlations across the entire image, the PDP is localised. Where Dynamic Textures utilises matrix multiplication, the PDP is integer-based. The core of the algorithm makes use of only addition and subtraction. The PDP correlates successive frames of a sequence to extract only the immediate optical flow. Doretto et al obtain a more sophisticated and flexible ARMA model of the scene behaviour, but this carries a steep computational cost.

The strengths of the PDP can be compared to those of the well-known LBP (Local Binary Pattern [17]). The LBP algorithm is used on static images, and is used primarily in texture analysis. The principle strength of this analysis technique is its simplicity, and its reliance on nothing more than inter-pixel relations – absolute values are unimportant. The PDP exhibits these two strengths also. Where the LBP is localised within a single frame, however, the PDP projects into the next frame. Where the LBP simplifies its surroundings to a binary code, the PDP uses the full difference pattern. Thus the *Projected Difference Pattern* is used to give an estimate of the local optical flow, analogous to how the LBP can be used to estimate local image gradient.

3. Algorithm Description

The core PDP process is what makes the algorithm efficient enough to process video data in real-time.

A fixed region (minimum of 3×3) around a pixel is differenced with the corresponding pixel in the next frame, and the position of the smallest difference in the region is assumed to be the correct optical flow vector. This "projected autocorrelation" (same region, projected in time) is the basis for many optical flow algorithms [15] – usually referred to as "Region Matching". The key to the stability of the algorithm is that unreliable data is cancelled out over multiple frames. Performing the same operation over several frame transitions and summing the difference data eliminates coincidental minima.

The confidence measure of a flow direction is taken directly from the value of the minimum difference in the difference pattern. A low minimum value represents a high confidence that the vector returned is a valid choice, while a high value indicates a relatively even spread across the differenced region; hence no clear direction of optical flow. In the results presented in this paper, there was no scaling of the confidence data to account for contrast, as the contrast was sufficiently even across the sample sequences. Regions of high contrast generate larger minima than regions of low contrast, and thus possess poorer confidence measures. In exchange for a performance loss, the confidence minima can be scaled by the total value of all the grey-level differences stored in the difference pattern.

To calculate the support, or "expectedness" of each returned vector, a history is kept of previous vector choices. If the most recent frame transition analysis returns the same vector as in the last *n* analyses, then the incoming frame is highly supported. If the most recent vector has not been previously chosen, then the new data is poorly supported, and the data is likely to be novel.

The result shown in Figure 1 is based on a single transition. Prediction noise decreases harmonically with the addition of successive frames of data. The best prediction results are obtained for simple first-order linear motion, without necessitating the computational complexity of Dynamic Textures [6].

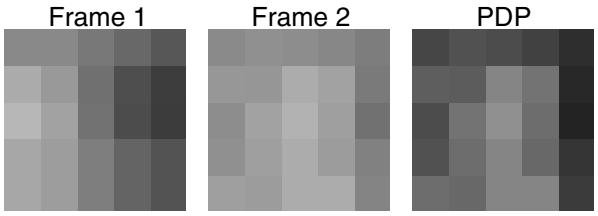


Figure 1. Algorithm Example. The first two frames are taken from the smoke_sub sequence. The last frame shows the difference between the active area of the first frame, and the mid pixel of next (projected) frame. Note that motion is approximately 2 pixels to the right. The PDP shows the darkest pixel (least difference) two pixels to the right of the mid pixel.

The algorithm very deliberately avoids all higher levels of abstraction for speed of processing and greater robustness.

4. Sequence Synthesis / Compression Example

The model obtained by the PDP can be used for iterative generation of new frames. In operation, successive pixels are generated iteratively from each other in branching structures back to those parts of the image where new data is being synthesised - usually at the image border. There, the vector usually turns upon the originating pixel. Unless this pixel is in some way fuelled with new data, generation of successive frames will eventually begin to generate the same image – the simulation grows stale. Use of the PDP to generate a flow map and iteratively generate frames can be applied directly for background subtraction via differencing between the expected and the obtained data. Importantly, compression highlights the effectiveness of a model in representing the original data; that the data can be replaced by application of the model, shows that the model captures the data well.

To generate the sequence in Figure 2 (from "smoke_sub" in the MIT Temporal Texture Database [1]), only 17% of the original data was required. Though certainly "lossy" as a compression method, this shows how the PDP can allow data compression - in this case 83% - of the original data. The sequence was generated iteratively with a single model based on the analysis of

the entire 150 frames, which were later synthesised from the remaining 17% using the model data, and the first entire frame. If the sequence is analysed in multiples of fewer frames, then the effective compression ratio generally falls, but the generated sequence more closely matches the original sequence.

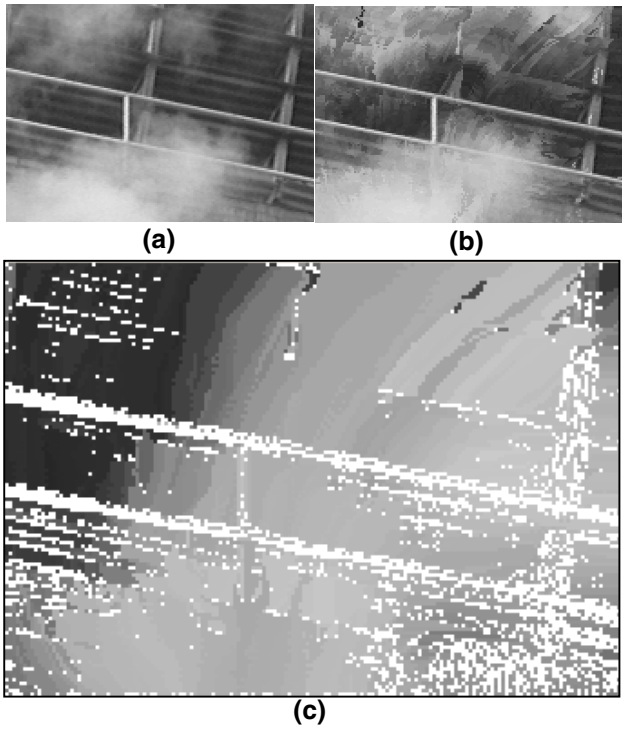


Figure 2. Synthesis Process. **(a)** The last frame of the original 150-frame "smoke_sub" sequence [1]. **(b)** The last frame of the sequence. **(c)** Branch map showing flow paths and originator pixels in white (17% coverage)

The two graphs shown in Figure 3 demonstrate how this effective compression – a good measure of model suitability and performance – varies with analysis length for the smoke_sub sequence. The lower graph shows the performance of the PDP. The upper graph was generated by the same process and fuelled with the same data, only all vectors were set to 0 before generation. This represents the equivalent performance of a static-update model as mentioned in Section 1 for modelling the scene. Low reproduction error can be gained at the cost of compression with the lowest point on both graphs showing the average result of 75 syntheses running over 2 frames each. Better compression is gained at the cost of representation of the data, right up to one synthesis running over the entire 150 frames of the original sequence. The highest point on the lower graph shows the performance of the PDP for Figure 2(b), above. The data required for the first frame at the start of each

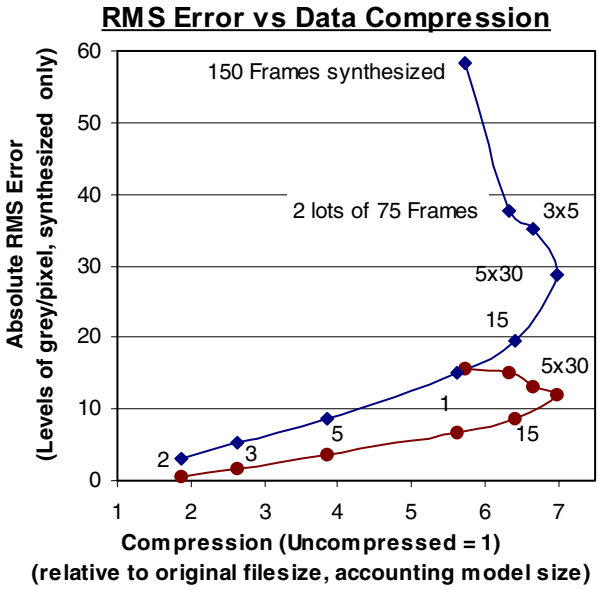


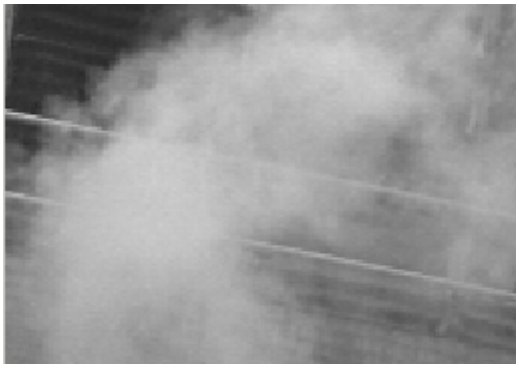
Figure 3. The compression / error trade-off for the smoke_sub sequence, comparing the performance of the PDP (lower graph) and a repeating static-image (reaches to top)

synthesis run is included in the compression factor. The original data "seeding" the synthesis is excluded from the RMS error calculation. The reason for the effective loss in compression for longer syntheses relates to the dataset used; a very stable scene might be synthesized accurately indefinitely, whereas a scene with more complex motion may require shorter analyses to prevent the build-up of incremental errors. As the synthesis length increases, the longer-term changes in the scene require more of the original data to be present in order to be represented. The low compression at lower synthesis lengths is related to the number of entire frames of data required to start each synthesis.

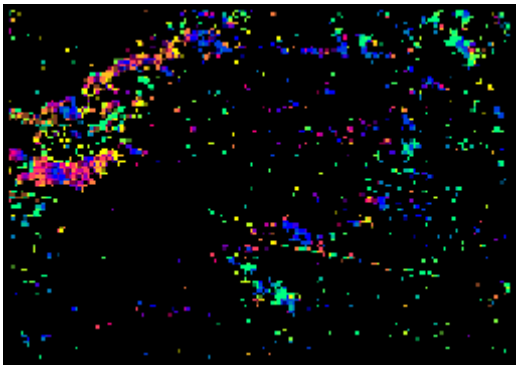
5. Analysis Example

Both Figures 4 and 5 were generated with a locale of 5x5 pixels, and a history of 50 frames for analysis. These settings allow for the modelling of optical flow with a velocity of up to 2 pixels per second, and will adapt to new behaviour within 25 frames.

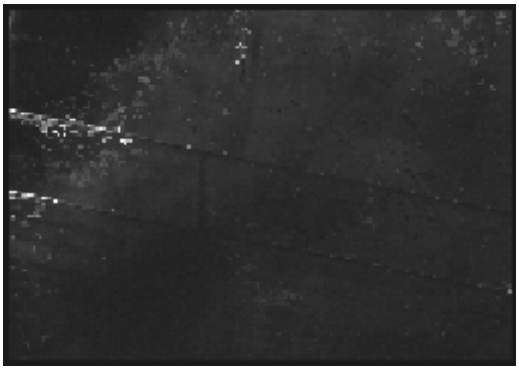
Starting with the smoke_sub sequence as analysed in Section 4, Figure 4(a) shows frame 54, at which point the smoke at the upper left has just begun to billow out turbulently. It can be seen in Figure 4(b), that there is a constant direction of flow around the top right of the image, and this is reflected in the colouring of the vector map. Around the turbulent area, however, there are many competing directions, and specifically the bright patch at mid-left is the result of an anticlockwise flow in the



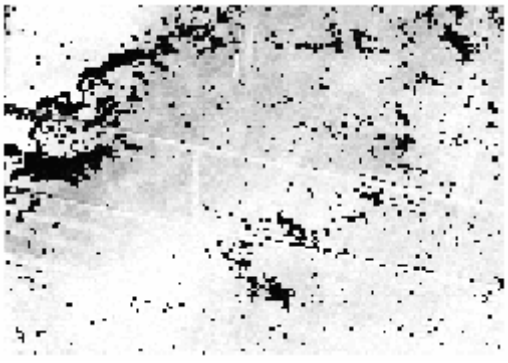
(a)



(b)



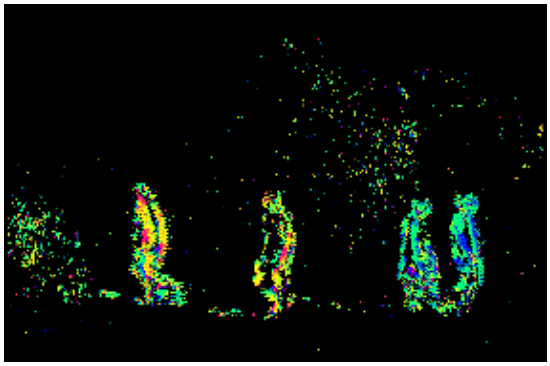
(c)



(d)

Figure 4. (a) Frame 54 of the Smoke sequence. **(b)** Colour-Coded vector map of optical flow. **(c)** Confidence. **(d)** Support.

(a)



(b)

(c)

(d)

Figure 5. (a) Frame 54 of the Walk-away sequence. **(b)** Colour-Coded vector map of optical flow. **(c)** Confidence. **(d)** Support.

smoke. Of most interest is Figure 4(c) - the Confidence data, where it can be seen that the sequence is being confidently modelled across all but the turbulent region. This matches the expectation in Figures 2 and 3 that the sequence behaviour is captured well. Figure 4(d) shows immediately that the behaviour of the entire central swathe has been consistent over the last 50 frames, and that the smoke is turbulent and changing only at its outside edges. Notably there is an area moving across the bottom right corner which has been highlighted in (d) but not (c). This can be interpreted as a recent change in the flow direction, which is nonetheless well modelled.

The same set of analyses is shown in Figure 5, but the walk-away sequence contains solid moving objects - the people. There has been no additional processing on the vector map.– note the coherence of the people against the background. Although the algorithm is set to a 50 frame history, only the pixels whose behaviour has changed in the last frame are highlighted in the support and confidence figures –bar the trails visible in Figures 5(c) and 5(d). When the history length is set below approximately 15 frames, the default behaviour of a pixel adapts to the person walking through it by the time the person leaves. Instead of the entire person being clearly highlighted in the support measure, we instead see those entry and exit trails which so plague pixel-based approaches [11, 12]. The trade-off for history length is shown clearly here – a longer history gives more stable results, but is less tolerant of gradual change. This is a well-known problem in background detection algorithms, and a matter normally left to implementation trials. Note that depending on the motion to be detected, there is a very short bootstrapping period for this algorithm.

The calculation and rendering of the support and confidence data can be performed iteratively. Hence, there is no significant performance penalty for a longer history, just a data storage penalty - either iterative changes must be saved to be eventually removed, or all frames must be saved and then the changes recalculated. There is however a direct trade-off between the other parameter, the size of the pixel locale, and performance. The order of the PDP algorithm is $M \times L$ where M is the number of pixels to process, and L is the size of the locale in pixels. The locale does not need to be square, but where it is, the processing time grows by the square of the locale width. As the locale grows smaller, fewer available predictor pixels are examined and the noise in the support and confidence maps grows. As the locale grows larger, more predicting values are searched, and the implementation approaches that of the Dynamic Texture [6].

6. Simple Extension of Application

The PDP was designed to be easily extended to more complex behaviours. Some simple extensions to the algorithm behaviour have been presented here. The use of the PDP as a background subtraction algorithm has already been covered in Section 2, and its use for data compression is covered in Section 3.

For a simple first-order sequence like the smoke_sub sequence in Section 4, there is no direct need to use the original data when synthesising. With an appropriate self-generating model for the values of the originator pixels, any first-order sequence can be generated continuously and maintain the appearance and behaviour of the original. This would allow simple sequence extension, or artificial sequence synthesis.

7. Summary

The PDP offers a solid basis for developing algorithms with spatiotemporal performance requirements. Though simple and efficient, it can be directly adapted to achieve several common video processing tasks, and can be used to extract valuable information about a sequence. The algorithm has been shown to perform robustly with highly temporal data, and importantly it gives a direct measure of its own suitability to given data. It is easily extended and can be built upon to arbitrary complexity within the same structural framework.

8. References

[1] M. Szummer, "MIT Temporal Texture Database," vol. 2004-2005, 1995, pp. Online resource of standardised temporal textures, http://vismod.media.mit.edu/pub/szummer/temporal-texture/raw/.

[2] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh, "Background Modelling and Subtraction of Dynamic Scenes." Real-Time Vision and Modeling lab: Siemens Corporate Research, 2003.

[3] P.-M. Jodoin and M. Mignotte, "Unsupervised Motion Detection Using a Markovian Temporal Model With Global Spatial Constraints," pp. 2591-2594, 2004.

[4] H. Greenspan, J. Goldberger, and A. Mayer, "Probabilistic Space-Time Video Modeling via Piecewise GMM," *IEEE Transactions on Pattern Analyses and Machine Intelligence*, vol. 26, pp. 384-396, 2004.

[5] D. Cremers, G. Doretto, P. Favaro, and S. Soatto, "Dynamic Texture Segmentation," *Department Of Computer Science, UCLA, LA*, 2003.

[6] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, "Dynamic Textures," *International Journal of Computer Vision*, vol. 51, pp. 91-109, 2002.

[7] G. Doretto and S. Soatto, "Editable Dynamic Textures," presented at CVPR, Madison, Wisconsin, USA, 2003.

[8] D. W. Dong and J. J. Atick, "Statistics of Natural Time-Varying Images," *Network: Computation in Neural Systems*, vol. 6, pp. 345-358, 1995.

[9] P. O. Hoyer and A. Hyvärinen, "A Multilayer Sparse Coding Network Learns Contour Coding From Natural Images," *Vision Research*, 2002.

[10] F. Kahl, R. Hartley, and V. Hilsenstein, "Novelty Detection in Image Sequences with Dynamic Background," presented at 2nd Workshop on Statistical Methods in Video Processing, European Conference on Computer Vision, Prague, 2004.

[11] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Background Modeling and Subtraction by Codebook Construction." Computer Vision Lab, University of Maryland, 2004, pp. 3061-3064.

[12] F. Porikli and C. R. Wren, "Change Detection by Frequency Decomposition: Wave-Back." www.merl.com: Mitsubishi Electric Research Laboratories, 2005.

[13] D. H. Hubel, *Eye, Brain, and Vision*. New York: Scientific American Library, 1988.

[14] T. L. Huntsberger, J. R. Rose, and D. Girard, "Neural Systems for Motion Analysis: Single Neuron and Network Approaches," *Intelligent Systems Laboratory, Dept. of Computer Science, University of South Carolina*, 2000.

[15] J. L. Barron, S. S. Beauchemin, and D. J. Fleet, "On Optical Flow," presented at AIICSR, Bratislava, Slovakia, 1994.

[16] I. Patras, M. Worring, and R. v. d. Boomgaard, "Dense Motion Estimation Using Regularization Constraints on Local Parametric Models," *IEEE Transactions on Image Processing*, vol. 13, 2004.

[17] T. Mäenpää, "The Local Binary Pattern Approach to Texture Analysis - Extensions and Applications," in *Electrical and Information Engineering*. University of Oulu: University of Oulu, 2003, pp. 80.