TRACKING BASED SOLELY ON LOCATION OF INTEREST POINTS

G. Dauphin

Laboratoire de Traitement et Transport de l'Information
Institut Galilée, Université Paris 13, France
gabriel.dauphin@univ-paris13.fr

ABSTRACT

This paper proposes two algorithms for tracking a region of interest. Given a fairly small number of spatial points at each frame and a high level of outliers, a bounding box is drawn around the region of interest. The first algorithm searches the bounding box having the greatest density. The second algorithm searches the bounding box whose theoretical 2D random distribution matches best with the 2D empirical random distribution. An estimate of the error is also provided by the second algorithm.

These two algorithms are tested on synthetic data and real data: the second algorithm has better performance on synthetic data, whereas the first algorithm has better performance on real data and could be used in real-time applications.

As information related to a feature can be represented as a set of points, these algorithms may prove useful to computer vision designers in feature selection or in data fusion (fusion of different tracks derived from different features).

Index Terms— Tracking, Interest Points, Feature Selection, Data Fusion.

1. INTRODUCTION

There is an increasing number of video-based contents available and a growing demand for extracting high level information from these videos: tracking, video retrieval from large databases, video-based recognition, action recognition.

Tracking plays an important role in computer vision, (a survey can be found in p. 8-12 of [16]). With the substantial literature on the topic, a wide choice exists when designing a tracker. Low-level features can be colour [13] (choice of a colour space), texture [12], intensity and orientation of motion (optical flow or interest points), edges [7], related to the background [18]. Such information can be collected in various ways so as to be robust with respect to small changes: histograms, local binary pattern [12]. Information can be collected over the whole region of interest, or restricted to edges [22] or interest points [15]. Information can be first fused in spatial- and then in time-domain (mean-shift with scale adaptation [5]). Information can also be fused in time- and then in space-domain (particle filtering). Hybrid methods exist too [10, 4].

Unifying these algorithms in a common framework is an issue: softwares are being developed [14]. In the future, it should become a major issue, as it would allow the use of the best algorithms with no restrictions on how a designer may choose the relevant features, the assumptions related to the motion and the target's shape. It would help applying feature selection as in [13]. When commenting on the survey, [1] states: *Given the fact that something as simple as face detection requires sophisticated models that are often dedicated to a given task in terms of parameter settings rather than generic in nature, confirms the highly complex nature of studying human dynamics. The development of technology that minimizes parameter settings and works in any unconstrained environment, is the holy grail of most research.*

This paper is an opportunity to promote the use of spatial points as the generic way of collecting the information provided by any given feature. In terms of the trade-off between complexity and precision, the arguments in favour are listed.

- Spatial-points-based methods should be better than density-based methods, just like edge-based methods can seem better than region-based methods [22].
- Spatial points may entail sufficient information, as D.M. Gravila in [8] recalls: *some experiments in human perception have been made with moving light displays attached to body parts; the studies have shown "human observers can almost instantly recognize biological motion patterns even when presented with only few of these moving dots"*.
- Using non-equally spaced points allows high precision in some areas and low precision in other areas, this has been a research issue in image compression, [23].

The promoted viewpoint is illustrated in [15], where action recognition is achieved on the KTH database [20] and the Weizmann database [2]. Interest points are first computed by averaging locally the optical flow and selecting the points having a residual motion, most of the points being near the

person and some points being all over the image. Outliers are pruned out using the RANSAC algorithm recalled on p. 3 of [15]. The position of the person is then estimated by fitting the maximum number of interest points in a BB of *known size*. Features related to the action are extracted by computing the average empirical distribution of these interest points inside the window.

The viewpoint is also illustrated in a tracking algorithm in [7], where spatial points are computed using a hysteresis thresholding on the gradient's magnitude and on the change in the gradient's direction. The use of a probabilistic model enables to set a relationship between the size of the BB and the threshold. A split and merge algorithm is used to find the BB.

This paper proposes two algorithms that find the unique BB given a set of points, most of these points being near the target and some being anywhere in the image. These two algorithms can be considered as trackers when combined with a Kalman filter designed with appropriate assumptions on the target's motion and when combined with an appropriate Hidden Markov Model (HMM) to find track initiation and termination. These two algorithms and some others are presented in section 2, together with synthetic data. In section 3 the two algorithms are transformed into trackers and tested on real data. Section 4 ends the paper with a conclusion.

Fig. 1. Frame extracted from a video of the KTH database; the white crosses mark the interest points associated with pixels having a residual motion.

2. DESCRIPTION OF THE SYNTHETIC DATA AND OF THE PROPOSED ALGORITHMS

Figure 1 shows a frame extracted from a video of the KTH database where the white crosses mark the interest points associated with pixels having a residual motion (steps 3 and 4 of the Tracking Algorithm in section 3). Most of these points are near the walking person, some are quite far apart. The challenge is to find an estimate of the true BB given the set of points and algorithms solving this challenge appear not to be available in the literature. The set of points is denoted M_n and the BB is defined by the coordinates of the lower left point (x_m, y_m) and the upper right point (x_M, y_M). The main assumption is that the points are drawn from a mixture of two uniform distributions, the first covers the entire image and the second is restricted to the BB. Note that in [7], the binomial distribution used is derived also from a uniform distribution:

$$M_n = (1 - \chi_n)U_n + \chi_n W_n \tag{1}$$

where notations are defined as:

$$\begin{array}{lcl} U_n & \rightsquigarrow & \mathcal{U}([0,1] \times [0,1]) \\ W_n & \rightsquigarrow & \mathcal{U}([x_m, x_M] \times [y_m, y_M]) \\ \chi_n & \rightsquigarrow & \mathcal{B}(1,p) \quad \text{(i.e. } P(\chi_n = 1) = p) \end{array} \tag{2}$$

Five algorithms are presented in the order of increasing complexity. Their Matlab implementation is available at `http://www-l2ti.univ-paris13.fr/~dauphin/`. The first three algorithms find seperately estimates of x_m, x_M and y_m, y_M using seperately the x-coordinates and the y-coordinates of the M_n-points.

The first algorithm, named MAX, finds the smallest rectangle enclosing all the M_n-points:

$$\left\{ \begin{array}{l} x_m^{(1)} = \min_{1 \le n \le N} x_n \\ x_M^{(1)} = \max_{1 \le n \le N} x_n \\ y_m^{(1)} = \min_{1 \le n \le N} y_n \\ y_M^{(1)} = \max_{1 \le n \le N} y_n \end{array} \right. \tag{3}$$

The second algorithm, named BTC, is inspired by the Block Truncation Coding (BTC) technique used in image compression [6]. Statistical evaluations are computed using the x- and y-coordinates of M_n: mean ($< x_n >$, $< y_n >$), standard deviation (σ_x, σ_y), ratio of the number of points whose x-coordinates/y-coordinates are greater than their mean ($\frac{N_x}{N-N_x}$, $\frac{N_y}{N-N_y}$). The BB is derived from these statistical evaluations in the same way as a BTC decoder gives a bilevel approximation of a $n \times n$ block [6]:

$$\left\{ \begin{array}{l} x_m^{(2)} = < x_n > - \sqrt{\frac{N_x}{N-N_x}} \sigma_x \\ x_M^{(2)} = < x_n > + \sqrt{\frac{N-N_x}{N_x}} \sigma_x \\ y_m^{(2)} = < y_n > - \sqrt{\frac{N_y}{N-N_y}} \sigma_y \\ y_M^{(2)} = < y_n > + \sqrt{\frac{N-N_y}{N_y}} \sigma_y \end{array} \right. \tag{4}$$

where x- and y-notations are defined as

$$\begin{array}{l} < x_n > = \frac{1}{N} \sum_{n=1}^{N} x_n \\ N_x = \#\{n | x_n \ge < x_n >\} \\ \sigma_x = \sqrt{< (x_n - < x_n >)^2 >} \end{array} \tag{5}$$

The third algorithm, named LR, is a Linear Regression; it takes advantage of the great quantity of data available in this stochastic definition of the challenge. The regression formula are two multivariate polynomials of order three. The five features have been chosen for their diversity: the median ($\tilde{x}$, $\tilde{y}$), the coordinates ranked to be the fifth smallest ($x_{[5]}, y_{[5]}$) and the fifth greatest ($x_{[N-5]}, y_{[N-5]}$), the mean absolute error ($< |x_n - \tilde{x}| >$, $< |y_n - \tilde{y}| >$), the mean squared error w.r. to the median ($\sqrt{< (x_n - \tilde{x})^2 >}$, $\sqrt{< (y_n - \tilde{y})^2 >}$).

The fourth algorithm, named DMX (Density Maximisation), finds the BB by maximising the ratio of the density of points inside the BB to the density of points outside the BB. Maximizing the density of points inside the BB as in [15] would yield too small BB, (note that [15] searches BB of fixed size). [7] is slightly different as it assumes that a local measure of meaningfulness is available, which is indeed appropriate when a large amount of data is available. [7] achieves also multi-object tracking. The objective function of DMX is defined as:

$$[x_m,\ y_m\ , x_M\ , y_M] = \arg\max_I \left[\frac{\#\{n|(x_n, y_n) \in I\}}{\#\{n|(x_n, y_n) \notin I\}} \frac{S - S_I}{S_I} \right] \quad (6)$$

This objective function is not concave and mean-shift like algorithms should not be used. The searching space is first reduced to BB whose four sides are set on some of the M_n-points. The algorithm used is a tree traversal algorithm [17].

	The DMX algorithm
1:	Select the MAX BB.
2:	Find the 4 BB by moving inwards each side of the BB.
3:	Select the best of the 4 BB.
4:	Repeat 2 and 3 as long as the BB is not empty.
5:	Select the best of all BB.

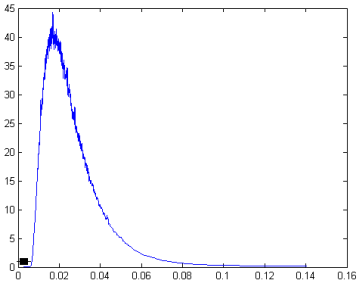


Fig. 2. Density of probability of J defined in (11) and down on the far left minimal values of J for several given sets of points.

The fifth algorithm, named BHM for Bivariate Histogram Matching, finds the BB whose theoretical 2D distribution matches best the empirical 2D distribution. This idea is usually implemented using the Bhattacharyya factor [5]. The use of the Bhattacharyya factor has two drawbacks.

- It is necessary to choose an appropriate scale at which values are quantified and binned.
- It is difficult to estimate the error.

BHM avoids these drawbacks: computing the empirical cumulative distribution function needs not choosing any scale; the statistical estimator used seems to be distribution free which helps estimating the error.

The statistical criteria is an extension of the Cramér-von Mises (CVM) criteria to bivariate distributions as in [11]. CVM has been chosen rather than the more often used Kolmogorov-Smirnov criteria. The reason is that the gradient of the CVM-objective function is regular: CVM is defined with a square norm whereas the Kolmogorov-Smirnov is defined with a maximum norm.

As explained in [11] and regarding bivariate distributions, there are four definitions of causality and hence four definitions of bivariate cumulative distributions:

$$\begin{array}{llll} P(X \leq x & \text{and} & Y \leq y) \\ P(X \leq x & \text{and} & Y > y) \\ P(X > x & \text{and} & Y > y) \\ P(X > x & \text{and} & Y \leq y) \end{array} \quad (7)$$

The criteria compares the empirical cumulative distribution with the theoretical cumulative distribution at the M_n-points and for each of these four definitions of causality. The four sets of values of the four empirical cumulative distributions are defined as:

$$\begin{array}{l} HN^{++}(n) = \frac{1}{N}\#\{k|(x_k \leq x_n) \text{ and } (y_k \leq y_n)\} \\ HN^{-+}(n) = \frac{1}{N}\#\{k|(x_k > x_n) \text{ and } (y_k \leq y_n)\} \\ HN^{--}(n) = \frac{1}{N}\#\{k|(x_k > x_n) \text{ and } (y_k > y_n)\} \\ HN^{+-}(n) = \frac{1}{N}\#\{k|(x_k \leq x_n) \text{ and } (y_k > y_n)\} \end{array} \quad (8)$$

The four theoretical cumulative distributions are defined as:

$$\begin{array}{l} HT^{++}(x,y) = (1-p)xy + pF_{x_m,x_M}(x)F_{y_m,y_M}(y) \\ HT^{-+}(x,y) = \\ \quad (1-p)(1-x)y + p(1-F_{x_m,x_M}(x))F_{y_m,y_M}(y) \\ HT^{--}(x,y) = \\ \quad (1-p)(1-x)(1-y) + \\ \quad p(1-F_{x_m,x_M}(x))(1-F_{y_m,y_M}(y) \\ HT^{+-}(x,y) = \\ \quad (1-p)x(1-y) + pF_{x_m,x_M}(x)(1-F_{y_m,y_M}(y)) \end{array} \quad (9)$$

where $F_{m,M}$ is the cumulative distribution of a uniform distribution:

$$F_{m,M}(s) = \frac{\mathbf{1}_{[m,M]}(s)}{M-m}(s-m) + \mathbf{1}_{(M,1]}(s) \quad (10)$$

The objective function is defined as the mean square difference between the four empirical distributions and the four theoretical distributions:

$$\begin{array}{l} J = \frac{1}{4N}\sum_{n=1}^{N} J_n^{++} + J_n^{-+} + J_n^{--} + J_n^{+-} \\ (x_m, y_m, x_M, y_M) = \arg\min J\,(x_m, y_m, x_M, y_M, p) \\ \text{where} \quad \begin{array}{l} J_n^{++} = (HT^{++}(x_n, y_n) - HN^{++}(n))^2 \\ J_n^{-+} = (HT^{-+}(x_n, y_n) - HN^{-+}(n))^2 \\ J_n^{--} = (HT^{--}(x_n, y_n) - HN^{--}(n))^2 \\ J_n^{+-} = (HT^{+-}(x_n, y_n) - HN^{+-}(n))^2 \end{array} \end{array} \quad (11)$$

This minimisation is computed with Matlab's implementation of the Levenberg Marquardt algorithm, the gradient being defined in a seperate formula.

Algorithm	Time Per Frame	Distance
MAX	1.5×10^{-4}	38.5
BTC	3.3×10^{-4}	28.0
LR	4.4×10^{-3}	26.1
DMX	1.7×10^{-2}	24.0
BHM	2.8×10^{-1}	15.4

This table shows for each of the five algorithms the processing time and the error (defined as the average distance between the two extreme points of the estimated BB and the corresponding points in the true BB, this distance is measured in pixels and should be compared to the size of the images that is 160×120); and as expected, better precision is achieved at the cost of an increased processing time.

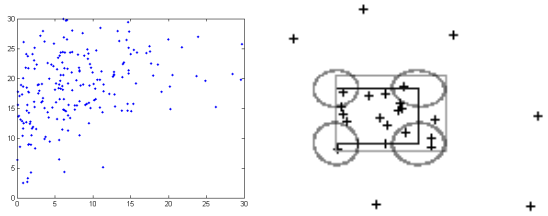


Fig. 3. Left: estimated error of $\hat{x}_m\, \hat{y}_m\, \hat{x}_M\, \hat{y}_M$ derived from (16) compared to the true errors. Right: M_n (crosses), true BB (grey rectangle), estimated BB (dark rectangle), error bounds (ellipses).

An algorithm computing an approximation of the error of the BHM's estimate of the BB is now presented. Numerical simulations show that J is distribution free and follows a distribution that depends only on $N = \#\{M_n\}$ (CVM is not necessarily distribution free [11]). This distribution is denoted $J \mapsto f_J^*(J)$, it is shown on figure 2 for $N = 30$.

Figure 2 shows also down on the far left that the obtained values of J are far below the average values of J. The reason is that the algorithm finds a theoretical cumulative distribution that overfits the empirical cumulative distribution. Many other values of p, x_m, y_m, x_M, y_M are equally likely even when the objective function is slightly increased. In a Bayesian framework, the mean and the covariance of the estimate is given by:

$$\begin{array}{l} K = \int_\Omega f_J^*(J(X))P(X)|d^4X| \\ E[X] = \frac{1}{K}\int_\Omega X f_J^*(J(X))P(X)|d^4X| \\ \Sigma = E[(X-\hat{X})(X-\hat{X})^T] \\ \quad = \frac{1}{K}\int_\Omega (X-\hat{X})(X-\hat{X})^T f_J^*(J(X))P(X)|d^4X| \end{array} \tag{12}$$

where $\Omega = \{X^T = (x_m, y_m, x_M, y_M) | x_m < x_M \& y_m < y_M\}$ is the set of candidate solutions of the BB and $P(X) \propto 1_\Omega(X)$ is the prior.

The approximations consists in replacing f_J^* by an exponential distribution which models the decay of f_J^*:

$$f_{J_N}(J) \approx \gamma_N e^{-\frac{J}{\gamma_N}} \tag{13}$$

where $\gamma_N = \frac{1}{\sqrt{N}}\left(0.011 + \frac{0.298}{N} - \frac{0.348}{N^2} + \frac{0.289}{N^3}\right) \approx E[J]$

A second-order Taylor expansion of the objective function is computed:

$$J \approx J(\hat{X}) + \tfrac{1}{2}(X-\hat{X})^T H_+ (X-\hat{X}) \tag{14}$$

where $X^T = [x_m\, y_m\, x_M\, y_M]$ and H_+ is a positive approximation [9] of the hessian. The prior and the searching space are also enlarged. (12) is approximated by:

$$\begin{array}{l} K \approx \int_{\mathbb{R}^4} \gamma_n e^{-\frac{J(\hat{X})}{\gamma_n}} e^{-\frac{1}{2\gamma_n}(X-\hat{X})^T H_+ (X-\hat{X})} |d^4X| \\ E[X] \approx \frac{1}{K}\int_{\mathbb{R}^4} \gamma_n e^{-\frac{J(\hat{X})}{\gamma_n}} X e^{-\frac{1}{2\gamma_n}(X-\hat{X})^T H_+ (X-\hat{X})} |d^4X| \\ \Sigma \approx \frac{1}{K}\int_{\mathbb{R}^4} \gamma_n e^{-\frac{J(\hat{X})}{\gamma_n}} (X-\hat{X})(X-\hat{X})^T \\ \quad e^{-\frac{1}{2\gamma_n}(X-\hat{X})^T H_+ (X-\hat{X})} |d^4X| \end{array} \tag{15}$$

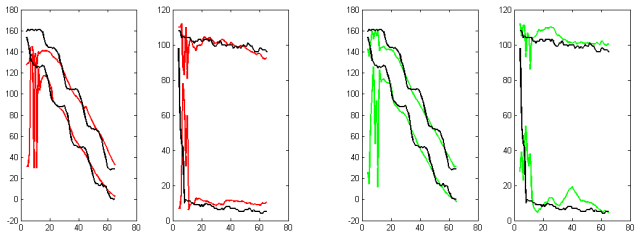
From all these computations, a simple expression of the covariance error is derived:

$$\left\{ \begin{array}{l} E[X] \approx \hat{X} \\ \Sigma \approx \gamma_n H_+^{-1} \end{array} \right. \tag{16}$$

Figure 3 shows on the left the estimated error of $\hat{x}_m$, $\hat{y}_m$, $\hat{x}_M$, $\hat{y}_M$ derived from (16) compared to the true errors. On the right the crosses mark the points, the grey rectangle indicates the true BB, the dark rectangle indicates the estimated BB and the ellipses indicate the error bounds. This figure shows that the estimated covariance error is a little overestimated.

3. APPLICATION TO A REAL PROBLEM

The video extracted from the KTH database is composed of a set of short videos where a person walks from left to right. To illustrate the promoted viewpoint, only motion has been chosen as a feature and interest points as a data representation. The overall tracking algorithm is defined in the following table.



G1 G2 G3 G4

Fig. 4. G1: x-coordinates of left and right sides of the BB as a function of time, (red/grey is the DMX estimation). G2: y-coordinates of lower and upper sides of the BB as a function of time (red/grey is the DMX estimation). G3: x-coordinates of left and right sides of the BB as a function of time, (green/grey is the BHM estimation with estimated error). G4: y-coordinates of lower and upper sides of the BB as a function of time (green/grey is the BHM estimation with estimated error). The dark line is the ground truth.

	The Tracking Algorithm
1:	Computations of interest points at each frame with the SURF algorithm [3].
2:	Use of the RANSAC algorithm [15] to find the dominant motion.
3:	Subtraction of each frame with the preceding shifted frame (according to the dominant motion).
4:	Computation of the new set of interest points, named M_n.
5:	Computation of the mean square error of the standard deviation of x- and y-coordinates.
6:	HMM processing using the implementation from [19].
7:	Optimization of parameters (using the assumption that there should be a small number of segmented shots).
8:	Temporal morphological filtering of the temporal segmentation.
9:	Application of the DMX and the BHM to the M_n-points in each shot to find the BB.
10:	Linear correction of BB with a true BB from a given frame.
11:	Kalman smoothing of the BB using the error bound estimate (16): implementation from K. Murphy and choice of state system inspired by [5].

Figure 4 shows on G1 the x-coordinates of left and right sides of the BB as a function of the time frame number, (red/grey line is the DMX estimation). It shows on G2 the y-coordinates of lower and upper sides of the BB as a function of the time frame number, (the red/grey line is the DMX estimation). It shows on G3 the x-coordinates of left and right sides of the BB as a function of the time frame number,

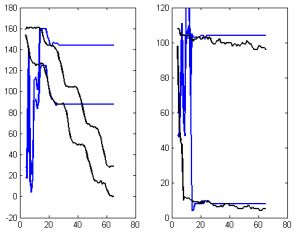


Fig. 5. Left: x-coordinates of left and right sides of the BB as a function of time, (red/grey is the RMT estimation). Right: y-coordinates of lower and upper sides of the BB as a function of time (blue/grey is the RMT estimation). The dark line is the ground truth.

(the green/grey line is the BHM estimation with estimated bounding error). The dark line is the ground truth. It shows on G4 the y-coordinates of lower and upper sides of the BB as a function of the time frame number (the green/grey line is the BHM estimation with estimated bounding error). The BB accounts for the size and position of the person walking (here from the left to the right), BB also accounts for the interior movements (the arms are swinging and an oscillation appears on the ground truth curves. Tracking using DMX exhibits good precision, however tracking using BHM exhibits less precision.

This overall tracking algorithm is tested against a real-time Robust Motion Tracking algorithm [21] that is here named RMT and which achieves background subtraction, connected component region segmentation and Kalman filtering. The implementation is from Fabian Wauthier. Parameters have not been modified with respect to that video. Post-processing consists in selecting the greatest BB. Figure 5 shows on the left the x-coordinates of left and right sides of the BB as a function of the time frame number, and on the right the y-coordinates of lower and upper sides of the BB as a function of the time frame number. The blue/grey line is the RMT estimation and the dark line is the ground truth. Hence RMT tracking is achieved with high precision during a short period of time. It then keeps record of the last estimate, yielding flat lines on figure 5.

4. CONCLUSION

This paper attempts to illustrate how spatial points can be used as a unifying framework to represent data extracted from any feature. Such a framework would help doing information fusion and feature selection. Two algorithms are proposed, they transform a set of points into an estimate of the bounding box (BB). The first algorithm finds the BB for which the density of points inside the BB is highest while the density of points outside is the smallest. The second algorithm finds

the BB whose theoretical 2D cumulative distribution matches best the empirical 2D cumulative distribution. The second algorithm has better results on synthetic data, however the first algorithm has better results on real data.

This discripancy between the performance on synthetic data and on real data raises important questions. Should the distribution inside the BB and in the neighbourhoods be less deterministic? Should there be more constraints on how spatial points are extracted from features? Should there be a greater number of points, less outliers; should these points follow more precisely a given random distribution?

5. REFERENCES

[1] "Video analysis of human dynamicsa survey," *Real-Time Imaging*, vol. 9, no. 5, pp. 321–346, 2003.

[2] S. Alpert, M. Galun, R. Basri, and A. Brandt, "Image segmentation by probabilistic bottom-up aggregation and cue integration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2007.

[3] H. Bay, A. Ess, T. Tuytelaars, and L. Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, Jun. 2008.

[4] A. Cavallaro, O. Steiger, and T. Ebrahimi, "Tracking video objects in cluttered background," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 4, pp. 575–584, Apr. 2005.

[5] K. Chen, B. Hu, R. Yang, and C. G. Jhun, "A Bhattacharyya-factor based Camshift application for video fast mobile target tracking," in *Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, vol. 4, Aug. 2010, pp. 1867–1871.

[6] E. Delp and O. Mitchell, "Image compression using Block Truncation Coding," *IEEE Transactions on Communications*, vol. 27, no. 9, pp. 1335–1342, Sep. 1979.

[7] F. Dibos, S. Pelletier, and G. Koepfler, "Real-time segmentation of moving objects in a video sequence by a contrario detection," in *IEEE International Conference on Image Processing, ICIP*, vol. 1, Sep. 2005.

[8] D. M. Gavrila, "The visual analysis of human movement: a survey," *Computer Vision Image Understanding*, vol. 73, no. 1, pp. 82–98, Jan. 1999.

[9] Q. Guan, J.-B. Fang, Z.-X. Chen, and J. Tao, "An approximation method of positive semi-definite matrix based on weighted F-norm," in *2012 International Conference on Industrial Control and Electronics Engineering. ICICEE*, Aug. 2012, pp. 1397–1400.

[10] B. Han and L. S. Davis, "Probabilistic fusion-based parameter estimation for visual tracking," *Computer Vision Image Understanding*, vol. 113, no. 4, pp. 435–445, Apr. 2009.

[11] U. Hanebeck and V. Klumpp, "Localized cumulative distributions and a multivariate generalization of the Cramér-von Mises distance," in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems. MFI*, Aug. 2008, pp. 33–39.

[12] M. Heikkila and M. Pietikainen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 657–662, Apr. 2006.

[13] D. Liang, Q. Huang, S. Jiang, H. Yao, and W. Gao, "Mean-shift blob tracking with adaptive feature selection and scale adaptation," in *IEEE International Conference on Image Processing, ICIP*, vol. 3, Oct. 2007, pp. 369–372.

[14] T. Lochmatter, P. Roduit, C. Cianci, N. Correll, J. Jacot, and A. Martinoli, "SwisTrack - a flexible open source tracking software for multi-agent systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, Sep. 2008, pp. 4004–4010.

[15] U. Mahbub, H. Imtiaz, and M. Rahman Ahad, "An optical flow based approach for action recognition," in *14th International Conference on Computer and Information Technology (ICCIT)*, Dec. 2011, pp. 646–651.

[16] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer Vision Image Understanding*, vol. 81, no. 3, pp. 231–268, Mar. 2001.

[17] M. Muja and L. D.G., "Fast matching of binary features," in *Proceedings of the 2012 Ninth Conference on Computer and Robot Vision*.

[18] J. Ning, L. Zhang, D. Zhang, and C. Wu, "Robust mean-shift tracking with corrected background-weighted histogram," *Computer Vision, IET*, vol. 6, no. 1, pp. 62–69, Jan. 2012.

[19] L. R. Rabiner, "Readings in speech recognition," A. Waibel and K.-F. Lee, Eds., 1990, ch. A tutorial on hidden Markov models and selected applications in speech recognition, pp. 267–296.

[20] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *17th International Conference on Pattern Recognition, ICPR'04*.

[21] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, 2000. [Online]. Available: http://www.cs.berkeley.edu/ flw/tracker/

[22] H. Tang, Y. Shi, J. Xia, and H. Yin, "A fast and accurate boundary tracking of moving objects in video," in *International Conference on Information and Automation. ICIA*, Jun. 2008, pp. 681–685.

[23] A. Zergainoh, N. Chihab, and J.-P. Astruc, "Construction of orthonormal piecewise polynomial scaling and wavelet bases on non-equally spaced knots," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, no. 1, 2007.