Motion Flow Tracking in Unconstrained Videos for Retail Scenario

Eduardo Marques Pereira*, Jaime S. Cardoso, and Ricardo Morla

INESC TEC and Faculdade de Engenharia Universidade do Porto, Portugal
ejmp@inescporto.pt

Abstract. We present a complete and modular framework that extract trajectories in a real and complex retail scenario, under unconstrained video conditions. Two motion tracking algorithms that combine features from crowd motion detection and multiple tracking are presented to build motion patterns and understand customer's behavior. Their evaluation across several datasets show promising results.

Keywords: optical flow, trajectory analysis, shopping behavior.

1 Introduction

Understanding customers' behavior and their purchase decision processes in an automatic way is an inestimable commercial advantage for the retail market. We present a modular framework that uses a CCTV (Closed Circuit Television) system of a shopping store to extract and analyze unstructured motion information in terms of trajectories, using two algorithms that combine features from crowd motion detection and multiple tracking. Specifically, we explore individual movements dictated by global motion to extract the number of common trajectories and their motion pattern representations. Our major contribution is in the comparative performance of the two proposed algorithms that compute global motion field for path learning in a real and challenging scenario: a video with small resolution, low frame rate, and uncontrolled camera deployment process.

The outline of the paper is as follows. In Section 2, we survey the related work. Next, we present our framework structure, Section 3. Section 4 explains the implemented algorithms to compute motion trajectories. A description and discussion about the experimental setup is presented in Section 5, followed by the evaluation methodology, Section 6, which reports the results. Finally, we formulate our conclusions and directions for future work in Section 7.

2 Related Work

Shopping Behavior Analysis: shopping behavior represents the detection and analysis process of semantic human actions involved on the decision of buying

* The first author would like to thank FCT - Fundação para a Ciência e Tecnologia (Portuguese Foundation for Science and Technology) for the financial support for the PhD grant with reference SFRH/BD/51430/2011.

J.M. Sanches, L. Micó, and J.S. Cardoso (Eds.): IbPRIA 2013, LNCS 7887, pp. 340–349, 2013.

products in complex scenes. Both academic and commercial interest has been growing for surveillance systems with human behavior understanding. Some commercial packages, such as SBLX [1] and CUBEA [2] systems, are already available but they only report simple statistics, lacking the support for automatic intensive data analysis. Popa and colleagues [3] followed a participating observation approach to build behavioral models and define types of shoppers, namely: disoriented, goal oriented, looking for support, fun-shopper and duo shopper.
Motion Estimation: motion estimation is often performed using feature tracking or optical flow. In the last years, interest point detectors have been extended from images to videos [4]. A recent work [5] shows that sparse points makes easier the recognition of periodic human actions, and states that dense sampling at regular spatio-temporal positions outperforms state-of-the-art sparse interest point detectors for action recognition and image classification.
Trajectory Analysis: motion can be encoded in trajectory structures useful for tracking objects in complex scenarios, and for action recognition. Sun et al. [6] extracted trajectories by matching SIFT descriptors between consecutive frames, and proposed a hierarchical framework to describe the spatio-temporal trajectory-based context. Lezama et al.' s work [7] shows that long-term motion analysis brings important cues for higher-level scene understanding. Their work was based on Brox and Malik's research [8], which presents consistent spatio-temporal segmentations of moving objects through the use of pair-wise distances between long term point trajectories. They used those distances to build an affinity matrix, passing it to a spectral clustering.

3 Framework Structure

Our scenario represents a physical structured scene with an unstructured motion where customers move randomly in various directions, and whose appearance models changes abruptly from frame to frame due to the low frame rate video. The traditional approach for motion analysis consists of detecting objects, tracking them, and analyze their tracks for event/activity detection. This standard processing does not work well on high density scenes with cluttered environment.

Under the aforementioned conditions and considering the principle of instantaneous motion field, we implemented and evaluated two approaches: *grid-based global dominant motion flow method*, and *kernel-based sink-seeking method*.

In this section, we describe the baseline framework, called *Common steps* (see Fig. 1), composed by the sampling, motion flow estimation, and filtering.

Sampling: this step extracts a set of interest points. We analyzed two type of sampling for our scenario. For dense sampling, we divided the image in a static grid. To minimize the aperture problem and detection of corners points, we considered a feature quality for good tracking [9]. For sparse sampling, we adopted a 2D space domain to avoid the joint spatio-temporal domain, namely the FAST algorithm. We verified empirically that sparse sampling is preferred, since dense sampling introduces noisy points from the cluttered background, and does not add discriminative value for the motion trajectory algorithms.

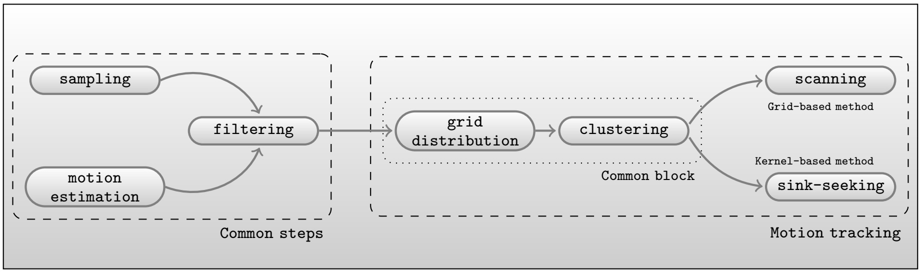


Fig. 1. Framework diagram

Motion Flow Estimation: this step is obtained from existing optical flow algorithms and is computed independently. The resulting motion flow map is used in the filtering step process in conjunction with the sampling points. We tested several algorithms that deal with short and large displacement of motion estimation. In this work, we considered two of them: the short-classical Farnebäck [10], and the large-descriptor matching in variational model (LDOF) [11]. Both of them present good results: the former runs on real-time, but introduces noise on areas with large appearance variations; the latter is an offline method, but permits the extraction of smoother and longer trajectories. We prefer the LDOF.
Filtering: this step consists in filtering the flow vector of each sampling point at each frame. The adopted approximation is a median filtering kernel that removes impulse noise, preserves edges, and smooths points in dense optical flow fields.

The described steps form the common pre-processing block of our framework, and are repeated every frame. Its output is a set of flow vectors at a specific frame, each one represented by $F_i = (x_i, y_i, u_i, v_i)$, where (x_i, y_i) is the sampling point, and (u_i, v_i) are the motion field components in x and y directions, respectively.

4 Motion Flow Tracking Methods

In this section, we describe the Motion tracking module. It is fed by a set of flow vectors that are collected and filtered out by the *Common-block* step (Section 4.1) every frame along a temporal gap, and are used as inputs by the two proposed algorithms to compute global motion flow trajectories at the final frame.

4.1 Common-Block

Shown in Fig. 1, each motion tracking algorithm has a pre-processing step with a two-fold purpose: to reduce the number of flow vectors, still maintaining the geometric structure of the flow field, and to obtain the local dominant motion flows. The image region is divided by a regular spaced grid. Each region is denominated a cell, C_i, and contains the flow vectors positions that lay inside it. Each flow vector is encoded by $F_i = (x_i, y_i, M_i, \theta_i, t_i)$, where (x_i, y_i) is the sampling point, M_i is the flow magnitude, θ_i is the flow angle relative to positive

$_x$ axis, and t_i is the frame. A dual-threshold on flow magnitude is used to remove flow vectors that have extremely small and high motion information.

On each cell C_i a two-step hierarchical clustering approach is applied. The first step considers a full-orientation histogram with 8 bins to express the orientation groups. The groups with weight above the histogram's median value are taken into account for next clustering step. The second step consists on a spatial clustering on each orientation group. A k-means approach with center initialization was adopted. At the end, we get several clusters for each orientation group that are ordered in a descendent-weighted way, considering the number of flow vectors that belong to them. These groups represent the local dominant flows, which are described by $L_i = (x_i, y_i, n_i, \theta_i)$, where (x_i, y_i) is the average position, n_i is the total number of flow vectors, and θ_i is the average orientation angle.

4.2 Grid-Based Global Dominant Motion Flow Method

This method computes the global dominant motion flows derived from local dominant motion flows. The work of [12] is the baseline approach. An extended *scanning* process, that combines local dominant flows from the neighborhood to obtain global motion flows, was implemented. Each cell C_i has a D step depth for looking for neighbors, which are defined as the adjacent cells that are in the direction of the current local flow. The D step factor defines the amount of adjacency shift. At the first iteration the current dominant flow searchs for neighbors that belong to the same orientation group. In case of no returning, the next iteration considers similar orientation groups, which correspond to the adjacent bin orientations. If no neighbor is obtained, any orientation is considered to keep continuity and permit abrupt flow orientation changes. To choose the next local dominant flow from the returned neighborhood's local flows, we use two pair-wise metrics: spatial closeness, and similarity closeness. The spatial measure just takes the euclidean distance between flow positions, while the similarity measure computes a weighted additive distance, as defined on [13].

4.3 Kernel-Based Sink-Seeking Method

This method follows the work of [14] and considers a *sink-seeking* process to extract motion patterns. It is a sliding-window technique that uses a kernel-based estimator to obtain global motion paths. The flow vector's representation in Section 4.1 can be translated to $F_i = (p_i, f_i)$, where $p_i = (x_i, y_i)$ is the flow position, and $f_i = (u_i, v_i)$ is the flow motion. The estimator incorporates neighborhood's flow motion to obtain the representative states of the global motion path, and its formula is given by:

$$W_{t,n} = \exp\Big(- \Big\| \frac{\tilde{f}_{i,t-1} - f_n}{h_{t-1}} \Big\|^2 \Big), \qquad (1)$$

where f_n is the neighbor flow motion, and h_{t-1} is the bandwidth. The motion flow field is $\{F_1, F_2, \ldots, F_n\}$, and the state of the *sink-seeking* process at each point i is $\tilde{F}_{i,t} = (\tilde{p}_{i,t}, \tilde{f}_{i,t})_{t=1,\ldots}$. The states are updated by:

$$\tilde{F}_{i,1} = F_i, \qquad \tilde{p}_{i,t+1} = \tilde{p}_{i,t} + \tilde{f}_{i,t}, \qquad \tilde{f}_{i,t} = \frac{\sum_{n \in Neighbor(\tilde{p}_{i,t})} f_n W_{t,n}}{\sum_{n \in Neighbor(\tilde{p}_{i,t})} W_{t,n}} \tag{2}$$

The relation between the next and previous states is linear, and the next location depends on the position and flow motion of neighboring points. The neighborhood is composed by the flow vectors positions that lay inside the kernel window, and whose angle between the current angle sink state is below a certain threshold, designated as *acceptance angle*, θ_{accep}.

We consider an automatic approach for the initial bandwidth selection and updating process. For the initial selection, we start with an initial bandwidth obtained from empirical experiments. Then, we proceed to a refinement step that estimates the greatest circular density region for a specific number of decreasing and increasing radius iterations. During the *sink-seeking* process statistical measures are computed, and two mapping relations are considered: angle statistics are related to window height, and magnitude statistics are related to window width. The intuition is the larger the window height, the wider the *acceptance angle*, then greater probability to change motion direction substantially, and the larger the window width, the longer the vector flow magnitude, then greater probability to account with a fast motion. The updating step is controlled by:

$$h_{w_t} = h_{w_{t-1}} \frac{\mu_{\alpha_{t-1}} + \sigma_{\alpha_{t-1}}}{\mu_{\alpha_t} + \sigma_{\alpha_t}}, \qquad h_{h_t} = h_{h_{t-1}} \frac{\mu_{\ell_{t-1}} + \sigma_{\ell_{t-1}}}{\mu_{\ell_t} + \sigma_{\ell_t}} \tag{3}$$

where $h_i = (h_{w_i}, h_{h_i})$ are the bandwidth components, μ_{α_i} and σ_{α_i} are the sink state's mean and standard deviation angle, and μ_{ℓ_i} and σ_{ℓ_i} are the sink state's mean and standard deviation magnitude, at frame i.

5 Experimental Setup

5.1 Dataset

Our dataset is composed by two videos with the characteristics shown on Table 1. The videos show a real shopping scene that for copyright reasons both their identity and video content are keep anonymous.

Table 1. Dataset video characteristics

File Format	Video Format	Demuxer	Codec	Bpp	Kbps	Frame Size	Fps	N frames
Non-interleaved AVI	DIVX	avini	ffodivx	24	295.8	352x288	1	301

The video streams and their content scenario have very challenging properties, that to the best of our knowledge were not explored before. It is a short temporal, low resolution and low frame rate video, with a cluttered scenario, and unconstrained illumination. Due to these reasons, manual annotation was done just for one video, which was used in this work to present the validation results. For each person, head and center of mass were considered for tracking. The video has 47 people, which gives a total of 109 trajectories (when a person leaves the scene and enters again, a new trajectory is created). Statistical values were extracted from trajectories to describe their shape information, evaluate scene complexity, and compare results between algorithms (see Table 3).

5.2 Evaluation Framework

From empirical analysis, we reach a baseline to apply a quantitative methodology to evaluate and compare the effectiveness and robustness of both algorithms. The baseline consists of a FAST sparse sampling, a median filtering kernel size of $K = (5, 5)$, LDOF's optical flow algorithm, and a cell size of $C = (13, 13)$.

Motion flow tracking algorithms have a number of factors that affect their results. The so-called *Common block* introduces a factor denominated by *cluster resolution*, which is related to the two-step hierarchical clustering and reflects the decision for each cell's orientation group to account with the most local dominant flow or all local dominant flows. We adopt the most local dominant flow as a good tradeoff between robustness and computational effort.

The Grid-based algorithm increments two factors related to the *scanning* process: i) *neighborhood similarity*, reflects about the proximity between two local dominant flows and varies from the two pair-wise metrics used; ii) *neighborhood continuity*, reflects about the integration of a different orientation group at the final iteration in the continuity decision. From empirical experiments we get better results adopting the spatial metric and avoiding the integration of other orientation groups in the scanning process.

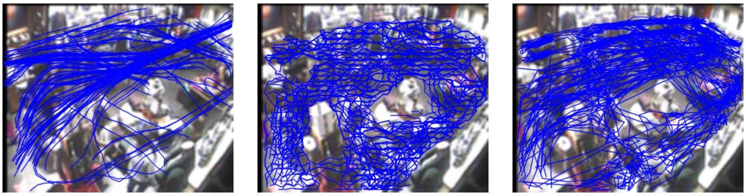
The Kernel-based algorithm presents three factors related to the *sink-seeking* process: i) *bandwidth initialization*, is related to the initialization and circular refinement search process to initialize the bandwidth; ii) *bandwidth optimization*, reflects the integration of an automatic and dynamic bandwidth size adjustment; iii) *acceptance direction*, is related to the *acceptance angle*, θ_{accep}, and reflects about the optimal angle threshold to account with local dominant flow neighbors. From several experiences, we adopt a bandwidth size of $h_w = (53, 10)$, with dynamic optimization, and an *acceptance angle* of $\theta_{accep} = 60$.

An evaluation methodology was designed to cluster trajectories by similarity to obtain the common ones, and to measure correspondence between extracted and annotated trajectories. For clustering, we use several distance functions, that account with trajectory position and shape, to build a similarity matrix to be used as input in a spectral clustering that has a K-means technique on its final step. A good guess about the number of valid partitions, $K = 39$, was obtained using a graphical tool for manually clustering, implemented by us.

We report the quality of correspondence between the most similar extracted trajectory and annotated trajectory with the miss detection and false positive rates. This procedure uses an one-against-all distance function between each annotated trajectory and all auto-generated ones, to obtain a distance matrix and solve the assignment problem with the Hungarian algorithm. The distance matrix entries were truncated by a threshold that is estimated using a K-means approximation. The upper bound of the cluster with the lowest values was taken as the desired threshold. If an auto-generated trajectory is assigned to an annotated trajectory, the match is considered correct if the distance between both is below a certain threshold. This threshold is calculated using the same K-means process. The miss detection rate is the number of unmatched annotated trajectories, and the false-positive rate is the number of annotated trajectories matched minus the correctly classified matches.

6 Results

This section presents the results about the extracted trajectories, a discussion about the evaluation of their similarity with annotated trajectories, and a critical comparison about the proposed algorithms. The total video duration, 301 frames, was used as the temporal gap to estimate motion trajectories (see Fig. 2).



(a) Kernel-based. (b) Grid-based. (c) Annotated.

Fig. 2. Complete set of extracted trajectories

Similarly to Section 5.1, we compute the statistical average values of trajectories extracted from both algorithms (see Table 2). Two measures defined on [15], and reported on Table 4, were computed. From both tables, we verify that manual trajectory set is more complex and less sparse than any of the auto-generated trajectory set. In general, trajectories extracted from the Kernel-based method are smoother and shorter than the ones extracted from the Grid-based method, which is more sensitive to noise and does not keep track of a smooth flow. This is explained because the kernel window does not accept flow vectors with large opposite directions, therefore the sink-seeking process maintains a coherent flow and its sliding window's overlap permits to build smoother trajectories with greater number of points. The Grid-based method produces more trajectories than the Kernel-based method, 339 trajectories against 192. It is related to the short continuity of the *scanning* process involving both *neighborhood continuity* and *neighborhood similarity* effects, whose also affect flow coherence.

To infer trajectory parameters similarity between auto-generated trajectories and annotated trajectories we use histogram comparison. Considering an aggregated distribution from both data histograms, we adopt the Scott's normal

Table 2. Characteristics of trajectories from both motion trajectories algorithms

Parameter		max	min	μ	σ
$\theta\angle_x$	Kernel	345.96	19.02	193.22	102.83
	Grid	335.20	39.56	179.64	48.86
$\theta\angle_{inc}$	Kernel	9.08	-8.59	0.58	3.00
	Grid	51.40	-55.38	2.64	17.61
ℓ	Kernel	10.38	1.06	4.35	2.03
	Grid	12.96	6.54	9.74	1.11
n_p	Kernel	97	25	46.36	17.67
	Grid	78	5	16.70	13.04

Table 3. Characteristics of annotated trajectories

Parameter	max	min	μ	σ
$\theta\angle_x$	330.68	3.67	106.12	67.68
$\theta\angle_{inc}$	44.27	-29.95	-1.64	19.32
ℓ	77.36	0.21	10.77	10.98
n_p	300	4	75.89	74.48

Table 4. Global shape measures on different settings

Parameter	Manual	Grid-based	Kernel-based
$\zeta_{complexity}$	0.48	0.74	0.96
$\zeta_{divergence}$	1.95e+36	2.48e+36	1.96e+36

reference rule to estimate the bin width, and use four metrics to measure histogram matching. The results are reported on Table 5, where we verify that the Kernel-based method is more correlated on length (ℓ) and number of points (n_p) parameters, Grid-based method has a strong shape divergence ($\zeta_{divergence}$) correlation, and mostly complete correlation on shape complexity ($\zeta_{complexity}$) parameter. For the remainders, both methods have equal degree of correlation.

Table 5. Histogram Matching of trajectory parameters for both algorithms

Match Method	$\theta\angle_x$		$\theta\angle_{inc}$		ℓ		n_p		$\zeta_{complexity}$		$\zeta_{divergence}$	
	GT-Kernel	GT-Grid	GT-Kernel	GT-Grid	GT-Kernel	GT-Grid	GT-Kernel	GT-Grid	GT-Kernel	GT-Grid	GT-Kernel	GT-Grid
Correlation	-0.229	**-0.228**	**0.82**	0.36	**0.91**	0.40	**0.86**	0.74	**0.14**	-0.32	0.9995	**0.9996**
Chi-square	**2.90**	3.20	**1.06**	1.12	**0.60**	1.60	**0.50**	1.49	4.54	**3.12**	0.09	**0.05**
Intersection	0.70	**1.10**	1.14	**1.17**	**1.03**	0.58	**1.43**	1.09	0.77	**1.65**	1.00	**1.01**
Bhattacharyya	**0.0**	0.57	0.52	**0.40**	**0.39**	0.63	**0.30**	0.49	0.75	**0.56**	0.20	**0.15**

Through qualitative inspection of clustering for common trajectories (see Section 5.2), we verify good partitioning results using several distance metrics, whose lead us to conclude that the number of common trajectories extracted from both algorithms is similar to the ground truth data, even with different number of trajectories. We use the same trajectory distance measures for both clustering and assignment steps, and we just report the most significant results of the assignment procedure on Table 6. Some measures are uncommon to appear on comparative trajectory evaluation works, but in our case present better results than more common ones: Curve Segment Hausdorff Distance (CHD), based on curve-to-curve matching; Traclus, based on parallel, perpendicular, and angle distances between line segments; Min-Hausdorff-Mod, based in the modified Hausdorff distance but instead of using the maximum in the Hausdorff distance, the minimum is taken; Min-Hausdorff-Mod$_{src-dst}$, represents the same distance measure but instead of using a comparative relation of the symmetrical Hausdorff distance, it just uses the Hausdorff distance of $h(src, dst)$.

For both algorithms, we verify two important conclusions: i) the Min-Hausdorff-Mod$_{Auto-GT}$ is the measure that provides best results with the lowest false positive rate, because, in general, manual trajectories are larger and more complex

Table 6. Distance measures characteristics and assignment results

Distance Metric	Grid-based					Kernel-based				
	max	min	μ	σ	false positive rate	max	min	μ	σ	false positive rate
Min-Hausdorff-Mod	4.67	1.84	3.68	0.82	70/109	10.96	1.55	6.67	2.36	34/109
Min-Hausdorff-Mod$_{GT-Auto}$	25.74	1.84	13.02	6.91	21/109	22.95	1.55	11.55	5.93	28/109
Min-Hausdorff-Mod$_{Auto-GT}$	13.68	1.91	6.43	2.28	**12/109**	20.58	1.37	10.90	4.71	**23/109**
DTW	786.18	157.62	465.19	161.99	40/109	993.29	176.28	569.13	199.04	33/109
LCS	0.19	0.0	0.08	0.07	51/109	0.14	0.0	0.07	0.05	66/109
CHD	231.58	3.62	136.11	67.70	56/109	482.32	3.25	201.03	137.30	65/109
Traclus	927.52	250.27	575.80	187.48	44/109	1103.71	256.38	664.56	203.54	36/109

than the auto-generated; ii) the miss detection rate is equal to the false positive rate since the assignment process always find a correspondence between a manual trajectory and an auto-generated one, which confirms previous conclusion that the number of common auto-generated trajectories is similar to the ground truth data. However, any distance measure was sufficiently accurate to account for all false positives. Fig. 3 illustrates some examples where trajectory assignments were considered valid, but in fact they are not. Fig. 4 shows examples of correct trajectory assignments for both algorithms, which confirm good approximations. Visual inspection of complete set of trajectory assignments leads us to conclude that Kernel-based method presents better similarities with ground truth data, which corroborates with the quantitative analysis presented previously.

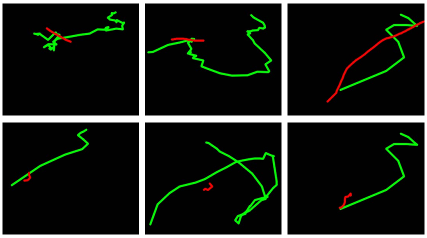


Fig. 3. False positives not detected. (First row: Kernel-based; Second row: Grid-based; Green: GT; Red: Auto)

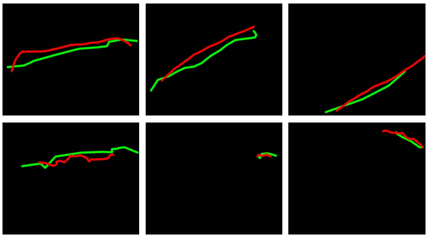


Fig. 4. Correct trajectory assignments. (First row: Kernel-based; Second row: Grid-based; Green: GT; Red: Auto)

7 Conclusions

In this paper we describe our approach to a real and complex retail scenario for understanding shopping behavior through the extraction and clustering of global motion flow trajectories. Our framework incorporates two motion flow tracking algorithms that present good and promising results under special video conditions, where conventional crowd motion and multiple tracking approaches are not useful, and also on common datasets whose results are not shown for lack of space. The future steps involve the validation of exact metrics to evaluate the influence of algorithm's parameters, the integration of an automatic process to select the best algorithm's parameters, the inclusion of temporal information, and the creation of a trajectory encoding scheme to estimate space layout.

References

1. (SBLX), S.B.X.: Axis cameras watch shopper's behavior (2009), http://www.axis.com/files/success_stories/ss_ret_sbxl_36113_en_0907_lo.pdf
2. CUBEA: Cubea customer behavior analysis system (2006-2010), http://www.identrace.hu/products/cubea.html
3. Popa, M., Rothkrantz, L., Yang, C.K., Wiggers, P., Braspenning, R., Shan, C.: Analysis of shopping behavior based on surveillance system. In: Dimirovski, G. (ed.) IEEE Int. Conf. on Systems and Man and and Cybernetics (SMC 2010), pp. 2512–2519. Kudret Press, Instanbul (2010)
4. Laptev, I.: On space-time interest points. Int. J. Comput. Vision 64(2-3), 107–123 (2005)
5. Wang, H., Ullah, M.M., Kläser, A., Laptev, I., Schmid, C.: Evaluation of local spatio-temporal features for action recognition. University of Central Florida, U.S.A (2009)
6. Sun, J., Wu, X., Yan, S., Cheong, L.F., Chua, T.S., Li, J.: Hierarchical spatio-temporal context modeling for action recognition. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2004–2011 (2009)
7. Lezama, J., Alahari, K., Sivic, J., Laptev, I.: Track to the future: Spatio-temporal video segmentation with long-range motion cues. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2011)
8. Brox, T., Malik, J.: Object segmentation by long term analysis of point trajectories. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 282–295. Springer, Heidelberg (2010)
9. Shi, J., Tomasi, C.: Good features to track. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR, pp. 593–600 (June 1994)
10. Farnebäck, G.: Two-frame motion estimation based on polynomial expansion. In: Bigun, J., Gustavsson, T. (eds.) SCIA 2003. LNCS, vol. 2749, pp. 363–370. Springer, Heidelberg (2003)
11. Brox, T., Malik, J.: Large displacement optical flow: Descriptor matching in variational motion estimation. IEEE Trans. Pattern Anal. Mach. Intell. 33, 500–513 (2011)
12. Ozturk, O., Yamasaki, T., Aizawa, K.: Detecting dominant motion flows in unstructured/structured crowd scenes. In: Proceedings of the 2010 20th International Conference on Pattern Recognition, ICPR 2010, pp. 3533–3536. IEEE Computer Society, Washington, DC (2010)
13. Eibl, G., Brändle, N.: Evaluation of clustering methods for finding dominant optical flow fields in crowded scenes. In: ICPR, pp. 1–4 (2008)
14. Hu, M., Ali, S., Shah, M.: Detecting global motion patterns in complex videos. In: ICPR, pp. 1–5 (2008)
15. Zhang, Z., Huang, K., Tan, T.: Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes. In: Proceedings of the 18th International Conference on Pattern Recognition, ICPR 2006, vol. 3, pp. 1135–1138. IEEE Computer Society, Washington, DC (2006)