Object Flow: Learning Object Displacement

Constantinos Lalos[1], Helmut Grabner[2], Luc Van Gool[2,3], and Theodora Varvarigou[1]

[1] School of Electrical & Computer Engineering, NTUA, Greece
lalosc@mail.ntua.gr, dora@telecom.ntua.gr
[2] Computer Vision Laboratory, ETH Zurich, Switzerland
{grabner, vangool}@vision.ee.ethz.ch
[3] ESAT-PSI/IBBT, K.U. Leuven, Belgium
luc.vangool@esat.kuleuven.be

Abstract. Modelling the dynamic behaviour of moving objects is one of the basic tasks in computer vision. In this paper, we introduce the *Object Flow*, for estimating both the displacement and the direction of an object-of-interest. Compared to the detection and tracking techniques, our approach obtains the object displacement directly similar to optical flow, while ignoring other irrelevant movements in the scene. Hence, *Object Flow* has the ability to continuously focus on a specific object and calculate its motion field. The resulting motion representation is useful for a variety of visual applications (e.g., scene description, object tracking, action recognition) and it cannot be directly obtained using the existing methods.

1 Introduction

Visual applications often rely on the information extracted by the moving objects inside a scene (e.g. cars, humans, machines etc.) These objects usually interact with other objects or the environment, thus modelling their dynamic behaviour is one of the basic tasks in computer vision.

(a) image pairs (b) object det. (c) optical flow (d) *Object Flow*

Fig. 1. Video captured at Abbey road in London (a). An appearance based object detector (b) can localize the human, however gives no information about its movement. On the other hand, optical flow (c) approaches cannot distinguish between object movement and other irrelevant movements in the scene. Hence, we propose a motion representation (d), which has the ability to focus only on moving objects-of-interest in the scene.

The estimation of the motion field for the whole scene is typically performed using optical flow methods. Works on optical flow start in the early 80's [1, 2] and target on establishing region correspondence between subsequent images[4]. Over the years a significant progress has been made, both in improving computational speed (e.g., [5]) and in dealing with large region displacements, (e.g., [6]). Recently, learning (e.g., [7]) and context [8] based approaches are taken into account in order to overcome the limitations of the classical optical flow formulation. In general, optical flow techniques have many possible applications, such as motion segmentation [9], object tracking [10], collection of statistics of the scene [11] or acting as human computer interface [12].

On the other hand, detection and tracking of individual objects (e.g., persons, cars) is important for several real-life applications including visual surveillance and automotive safety (e.g., [13]). In the last years, a lot of attention is paid to tracking by detection approaches (e.g., [14, 15]). Hereby, a pre-trained object detector is applied on every frame and then the obtained detections are associated together across images. Furthermore, on-line learning methods (e.g., [16]) can be also used to dynamically update the object model and to cope with the variations of the object appearance. The data association problem is further simplified, since a discriminative model is trained in advance, for distinguishing the object appearance from its surrounding background. However, due to the self-learning strategy in place, such approaches might suffer from drifting (see [17] for a recent discussion).

Contribution. We introduce a method for obtaining the displacement of an object – the *Object Flow* – directly whereas other irrelevant movements inside the scene (e.g., other objects or moving background) are ignored (see Fig. 1). Since no on-line learning is performed during runtime, the results are stable (i.e. do not suffer from drifting). Hence, the resulting motion representation is useful for a variety of visual applications and cannot be directly obtained using the existing methods such as optical flow or object detection/tracking.

The remainder of the paper is organized as follows. Firstly, the idea of training a classifier on object displacement is described in detail at Section 2. Then the experimental results and the conclusions are elaborated at Sections 3 and 4 respectively.

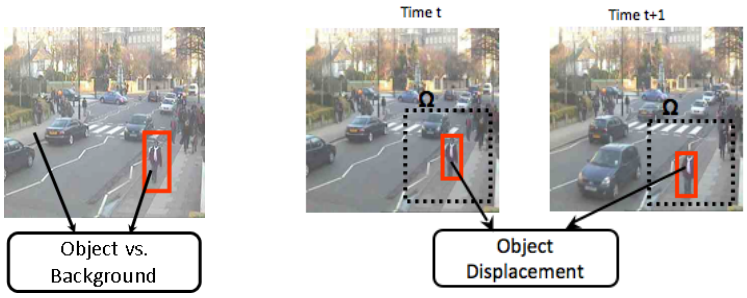
2 *Object Flow*

In this section, we first formulate the learning problem for training a model (classifier), which is then used to deliver the *Object Flow*.

2.1 Problem Formulation and Learning

The goal of object detection is to find a required object in an image. Most state-of-the art methods (e.g., [18]), train a classifier with the appropriate samples in order to

[4] Analogous to optical flow, where images are aligned based on a temporal adjacency, SIFT flow [3] can be exploited to match similar structures across different scenes. Recently it has been shown that parametric models such as affine motion, vignetting, and radial distortion can be modelled using the concept of Filter Flow [4].



(a) object detection (b) *Object Flow*

Fig. 2. Object detection (a) is usually formulated as a binary classification problem distinguishing the object of interest from the background class. In contrast, *Object Flow* considers the problem of learning object displacement locally.

distinguish the object-of-interest from the background, i.e. formulate the task as a binary classification problem. In comparison with the typical object detection approaches, we consider the problem of detecting the displacement and the direction of a moving object locally, i.e. within a certain region Ω, (see Fig. 2). Within this region, pairs of patches from different time intervals are classified. Nevertheless, the size of the search region Ω has its own role in the fulfilment of the object localization and direction estimation task. Especially in the case of abrupt motion or low frame rate video (see Sec. 3.1) an optimal estimation can be achieved by having a quite large search region. However, this size comes in contrast with the required computational complexity and might yield to ambiguities when more than one object are present in the scene.

Problem Formulation. We formulate the learning problem as a problem of learning a distance function, (see [19] for a recent overview). Our technique was inspired by the work of Hertz et al. [20], which learns a distance function for image retrieval by training a margin-based binary classifier (such as Support Vector Machines or Boosting methods) using pairs of samples. Positive pairs derived from the "same" class whereas negative pairs are samples drawn from two "different" classes. The learning problem is then formulated on the product space, i.e., $C : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{Y} = [-1\ , 1]$. Thus, the trained classifier $C(\mathbf{x}_1, \mathbf{x}_2)$ is supposed to give high confidence if the two samples $\mathbf{x}_1$ and $\mathbf{x}_2$ are similar, and low confidence otherwise.

Learning *Object Flow*. The overall learning approach is depicted at Fig. 3. For training a maximum margin classifier on object displacement in an off-line manner, a pool of appropriate samples has to be created. These samples should contain temporal information from pairs of images from the positive $\mathcal{X}^+$ and the negative $\mathcal{X}^-$ set respectively.

Positive set $\mathcal{X}^+$. A positive sample contains information about the way that object appearance transforms through time. Therefore, this sample is created by collecting two patches that derive from two different frames and contain the object under study i.e.

$$\mathcal{X}^+ = \{\langle \mathbf{x}_t^\star, \mathbf{x}_{t+1}^\star \rangle \mid \mathbf{x}_t^\star, \mathbf{x}_{t+1}^\star \in \Omega^{(i)} \text{ and correspond to an object}\} \quad (1)$$

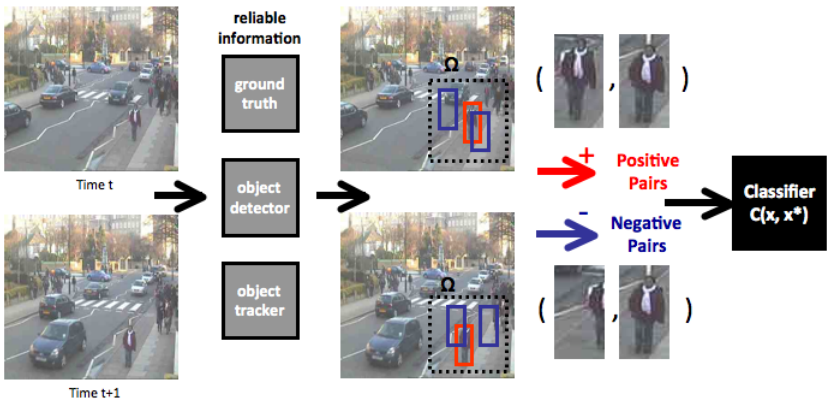


Fig. 3. Learning object's displacement is achieved by training a classifier with positive and negative labelled samples, which are locally extracted and contain temporal information.

The labelling of the object represented by the rectangles $\mathbf{x}^\star$ and $\mathbf{x}^\star_{t+1}$ can be accomplished using some reliable information, such as human labelling (ground truth), or the output from a high precision/recall detector or tracker.

Negative set $\mathcal{X}^-$. The negative set is divided into two subsets, i.e. $\mathcal{X}^- = \mathcal{X}^-_{obj} \cup \mathcal{X}^-_{back}$. The first subset of negative samples contains the object in the current frame with a patch that contains a portion of it in a different frame i.e.

$$\mathcal{X}^-_{obj} = \{\langle \mathbf{x}^\star_t, \mathbf{x}^{(i)}_{t+1} \rangle \mid \mathbf{x}^\star_t, \mathbf{x}^{(i)}_{t+1} \in \Omega^{(i)} \text{ and } \mathbf{x}^\star_{t+1} \text{ correspond to an object}\} \quad (2)$$

These training samples assist the classifier to suppress local maxima around the real object region. On the other hand, the second subset of negative samples contains regions from the background. These samples are particularly useful, when dealing with difficult scenarios, since they can force the classifier to respond with low confidence values on empty regions i.e.

$$\mathcal{X}^-_{back} = \{\langle \mathbf{x}^{(i)}_t, \mathbf{x}^{(j)}_t \rangle \mid \mathbf{x}^{(i)}_t, \mathbf{x}^{(j)}_t \in \Omega^{(i)}\} \quad (3)$$

Examples of a positive and negative samples are depicted in Fig. 4.

(a) pos. samples $\mathcal{X}^+$ (b) neg. samples $\mathcal{X}^-_{obj}$ (c) background samples $\mathcal{X}^-_{back}$

Fig. 4. Illustrative example of the typical training samples for training a classifier on *Object Flow*.

Classifier. In this paper we use the approach of boosting for feature selection. A classifier can be trained in an off-line [21] or in an on-line [16] manner. In order to use pairs of images as an input, we follow the empirical tests of possible adaptations, proposed by Hertz et al. [20]. An approach for learning how the object appearance alters through time is the concatenation of the two patches. Another intuitive approach is by finding the absolute difference of the vectors representing the two patches. Our empirical tests indicate that this classifier works better with the first approach. As features we use the classical Haar-like features [21].

2.2 Flow Estimation

Object Flow is a vector field. In order to estimate it, for each point x, y in the image a local image patch $\mathbf{x}$ is extracted and the displacement magnitude $D_{obj}(\mathbf{x})$ and the angle $\phi_{obj}(\mathbf{x})$ can be calculated. More specifically, let $C(\mathbf{x}, \mathbf{x}')$ be the classifier response for a pair of patches, where $\mathbf{x}$ is a patch in the current image and $\mathbf{x}'$ is a patch belonging to the neighbourhood region of local patches Ω in the previous image. We define the displacement Δx and Δy of an object on the x and y directions respectively, as the weighted sum of distances within the local region Ω. More formally,

$$\begin{pmatrix} \Delta x_{obj}(\mathbf{x}) \\ \Delta y_{obj}(\mathbf{x}) \end{pmatrix} = \frac{1}{\sum_{\mathbf{x}' \in \Omega} C(\mathbf{x}, \mathbf{x}')} \sum_{\mathbf{x}' \in \Omega} C(\mathbf{x}, \mathbf{x}') \begin{pmatrix} dx \\ dy \end{pmatrix} \tag{4}$$

where, dx and dy are the x and y axis distances of the patch $\mathbf{x}$ from $\mathbf{x}'$. Based on this, magnitude and angle can be calculated as,

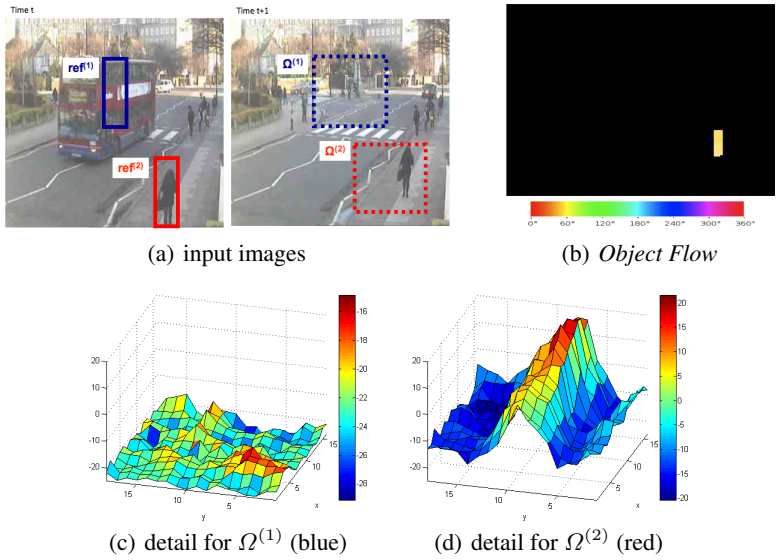
$$D_{obj}(\mathbf{x}) = \sqrt{\Delta x_{obj}(\mathbf{x})^2 + \Delta y_{obj}(\mathbf{x})^2}, \qquad \phi_{obj}(\mathbf{x}) = \tan^{-1}\left(\frac{\Delta y_{obj}(\mathbf{x})}{\Delta x_{obj}(\mathbf{x})}\right). \tag{5}$$

In order to reduce outliers, local region displacements within the region Ω have to extend a significant positive classifier response i.e.,

$$\bar{C_{obj}}(\mathbf{x}) = \frac{1}{|\Omega|} \sum_{x' \in \Omega} \hat{C}(\mathbf{x}, \mathbf{x}')^2, \quad \text{where} \quad \hat{C}(\mathbf{x}, \mathbf{x}') = \max(0, C(\mathbf{x}, \mathbf{x}')). \tag{6}$$

Summarizing, *Object Flow* is only reported, if the average classifier response is above some user defined threshold, which controls the sensitivity, i.e., $\bar{C}_{obj}(\mathbf{x}) > \theta$.

Illustrative Example. Fig. 5 depicts the *Object Flow* and the details for two specific regions. The trained classifier is evaluated on pairs of patches, using a reference patch at time t and patches from the corresponding local regions, $\Omega^{(1)}$ and $\Omega^{(2)}$, respectively at time $t + 1$. We use a grid of overlapping patches of the same size, centred at the reference patch. As we can observe in the resulting 3-D plot for the region $\Omega^{(2)}$, high confidence values represent the regions, on which the object is likely to occur at time $t + 1$. On the other hand, the confidence values are very low for the region $\Omega^{(1)}$, since there are no objects inside. For visualizing the angle $\phi_{obj}(\mathbf{x})$ and the displacement $D_{obj}(\mathbf{x})$ (see Eq. (5)), we use the hue and saturation channel from HSV color space respectively.



(a) input images (b) *Object Flow*

(c) detail for $\Omega^{(1)}$ (blue) (d) detail for $\Omega^{(2)}$ (red)

Fig. 5. Classifier responses for the regions $\Omega^{(1)}$ and $\Omega^{(2)}$ (a). Low classification responses are obtained if no object is present (c). In contrast, a clear peak, which shows the displacement of a particular object, is shown (d). The final *Object flow* field (b) is based on these local responses.

3 Experimental Results

In this section we present qualitative and quantitative experimental results of the *Object Flow* on different objects and datasets, including walking pedestrians, faces and moving coffee mugs. The efficiency of our approach is demonstrated using difficult scenarios that involve low frame rate and motion blurring from a moving camera. Furthermore, we compare our results with common methods, such as an object detector, tracker and optical flow. The proposed motion representation can be used either in a static or in a moving camera configuration. Throughout the experiments we use a dense Grid that comprises of 81×81 overlapping and equally sized cells and we set a threshold $\theta = 0.35$ (see Sec. 2.2). All experiments are performed on a 2.67 GHz PC with 4 GB RAM.

3.1 *Object Flow* for Pedestrians

We captured a dataset from a public camera located on Abbey road in London[5], which consists of $49,000$ frames. This dataset, obtained at a resolution of 384×284, is a challenging low frame rate scenario ($\sim$ 6 fps) that contains a complex background with various moving objects (e.g., cars). Therefore, we use a region Ω that comprises of 12×12 cells, since object motion is quite abrupt due to low frame rate. The first $40,000$ frames of this dataset are used for collecting the appropriate training samples (see Sec. 2.1) and the remaining ones are used for evaluation. More specifically, the

[5] `http://www.abbeyroad.com/webcam/`, 2010/03/03.

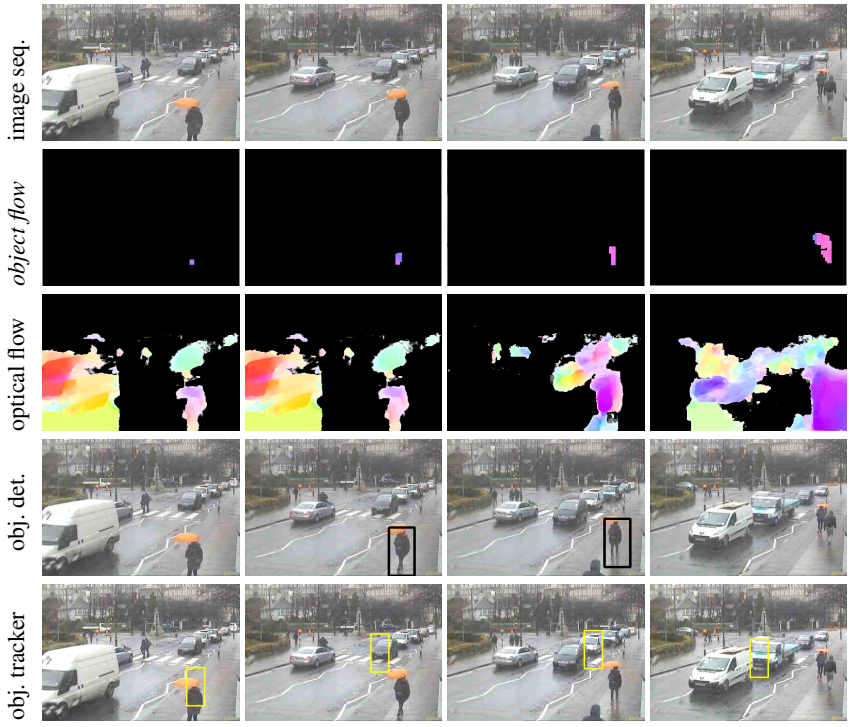


Fig. 6. In this experiment we present the benefits of *Object Flow*. Optical flow approaches disorient when similar objects are moving in the same/different direction with the object-of-interest (third row). In addition, human detection approaches do not have a constant detection rate (fourth row). Object tracking also suffers from drifting in complex environments (fifth row). *Object Flow* (second row) can simulate the motion field of a moving object correctly, by being able to focus only on the object under study.

results described in this section can be produced using a single classification approach that is trained off-line using a pool of $|\mathcal{X}^{+}| \approx 2,000$ positive, $|\mathcal{X}^{-}_{obj}| \approx 15,000$ negative object samples and numerous negative samples $\mathcal{X}^{-}_{back}$ from the background.

We perform illustrative comparisons with optical flow, human detection and object tracking methods. We use the approach described at [5] to calculate optical flow, in order to evaluate its performance against the proposed *Object Flow* technique. For human detection and tracking we adopt the approaches described at [22] and [16] respectively. All the competing techniques are used without modifying any of the input parameters given in their original implementation.

As it can be observed at Fig. 6, our approach has a good performance in human localization. In addition, direction estimation for the moving objects-of-interest (second row) is the same with the one provided by the aforementioned optical flow approach, which focuses in all the moving objects in the scene (e.g. cars, third row). On the other hand, combining optical flow with an object detection approach (fourth row) may lead

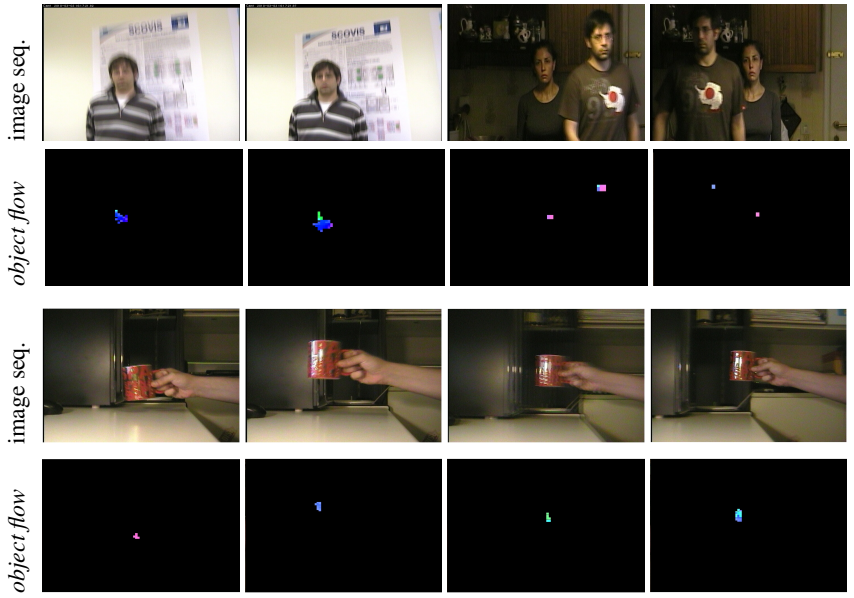


Fig. 7. In this experiment we train the classifier to deliver *Object Flow* for different object classes including faces and a coffee mug. The first and third row depict frames from different test sequences (camera movement, motion blur and multiple objects). The second and fourth row present the estimated *Object Flow*, respectively. (Video is availible at the authors' web-page.)

to possible pitfalls, since detection approaches do not have a constant detection rate, and thus have limited effectiveness in difficult environments. Similarly, tracking approaches (fifth row) disorientate on complex backgrounds, since objects of similar color or structure may appear inside the scene.

3.2 *Object Flow* for Different Objects

The performance of *Object Flow* is also tested using two different object classes, i.e. faces and a specific mug. The algorithm is evaluated on scenarios that contain abrupt motion and on a moving camera configuration. In detail, we use three different video sequences that consist of $1,200$ frames, where $1,000$ frames are used for training the classifier and 200 frames for testing. Two sequences were captured from a moving indoor camera and contain a moving face and mug respectively. Another sequence was captured from a static indoor camera and contains two moving faces. These datasets were taken at 25 fps with a 704×576 resolution using an AXIS 213 PTZ camera.

We evaluate our approach for each patch $\mathbf{x}$ using a region Ω that comprises of 6×6 cells. For the mug and the face sequence the classifier was trained using a pool of $|\mathcal{X}^+| \approx 1,000$ positive, $|\mathcal{X}^-_{obj}| \approx 4,000$ negative object samples and numerous

negative samples from the background. For creating face samples, an off-the-shelve face detection approach is adopted[6].

Illustrative results are depicted in Fig. 7. As it can been seen, the *Object Flow* has the ability to remain focused on the face and the mug even in cases of abrupt camera motion (second and fourth row). Furthermore, the proposed method can deal with more than one objects-of-interest present at the same scene (see Fig. 7 first and second row, in third and fourth column).

3.3 Quantitative Comparison

We adopt the coffee mug dataset, which is a moving camera scenario and consists of 200 frames (see Sec. 3.2). On this sequence, ground truth is created by manually labelling the values for the angle and the displacement. For each frame we calculate the absolute error between the ground truth and the values provided by our approach. Since there is only one object-of-interest in the scene, we consider the angle $\phi_{obj}(\mathbf{x})$ and displacement magnitude $D_{obj}(\mathbf{x})$ of the patch $\mathbf{x}$, for which the classifier has the maximum response $\bar{C_{obj}}(\mathbf{x})$, according to Eq.(6).

For comparison we also implemented a simple baseline approach that combines object detection and optical flow. In that case, the displacement and angle are estimated by finding the average optical flow within the region of a detection (i.e. if a detection is present). Therefore, we first, train a classifier [21] using $1,000$ positive samples for the object and a negative set that contains numerous object-free samples from the background. The resulting detections are fused together by applying non-maximal suppression. Finally, Lucas-Kanade method [2] for optical flow estimation is adopted.

For all the frames in the sequence, we calculate the mean absolute displacement and angle error. More specifically, the average displacement error is decreased from 12 pixels for the baseline approach to 9 pixels using our approach. Similarly, the mean angle error is decreased from 75° to 62°, respectively. The angle error seems to be quite large, which is quite reasonable, by taking into account that the object and camera change abruptly their direction in the chosen test sequence.

4 Conclusions

In this paper, we present the *Object Flow*, a method for estimating the displacement of an object-of-interest directly. Our approach is similar to optical flow, but it has the additional ability to ignore other irrelevant movements in the scene. This is achieved by training a classifier on the object displacement.

Experimental results demonstrate that the proposed approach achieves robust performance for different object classes, including pedestrians and faces. We are confident that *Object Flow* is useful for a variety of applications, such as object tracking or scene understanding. However, one current limitation is the computational complexity, which is going to be addressed in a future work.

[6] `http://opencv.willowgarage.com/wiki/`, 2010/04/28

Acknowledgments. This research was supported by the European Community Seventh Framework Programme under grant agreement no FP7-ICT-216465 SCOVIS.

References

1. Horn, B., Schunck, B.: Determining optical flow. Artificial Intelligence **17** (1981) 185–203
2. Kanade, T., Lucas, B.: An iterative image registration technique with an application to stereo vision. In: Proc. Int. Joint Conf. on Artificial Intelligence. (1981) 674–679
3. Liu, C., Yuen, J., Torralba, A., Sivic, J., Freeman, W.: Sift flow: Dense correspondence across different scenes. In: Proc. ECCV. (2008)
4. Seitz, S., Baker, S.: Filter flow. In: Proc. ICCV. (2009)
5. Werlberger, M., Trobin, W., Pock, T., Wedel, A., Cremers, D., Bischof, H.: Anisotropic Huber-L1 optical flow. In: BMVC, London, UK (2009)
6. Brox, T., Bregler, C., Malik, J.: Large displacement optical flow. In: Proc. CVPR. (2009)
7. Sun, D., Roth, S., Lewis, J., Black, M.: Learning optical flow. In: Proc. ECCV. (2008)
8. Wu, Y., Fan, J.: Contextual flow. In: Proc. CVPR. (2009)
9. Shi, J., Malik, J.: Motion segmentation and tracking using normalized cuts. In: Proc. ICCV. (1998)
10. Jean-Marc Odobez, Daniel Gatica-Perez, S.B.: Embedding motion in model-based stochastic tracking. IEEE Transactions on Image Processing **15** (2006) 3515 – 3531
11. Ali, S., Shah, M.: A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. In: Proc. CVPR. (2007)
12. Santner, J., Werlberger, M., Mauthner, T., Paier, W., Bischof, H.: FlowGames. In: 1st Int. Workshop on CVCG in conjunction with CVPR. (2010)
13. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: Proc. CVPR. Volume II. (1999) 246–252
14. Leibe, B., Schindler, K., Gool, L.V.: Coupled detection and trajectory estimation for multi-object tracking. In: Proc. ICCV. (2007)
15. Breitenstein, M.D., Reichlin, F., Leibe, B., Koller-Meier, E., Gool, L.V.: Robust tracking-by-detection using a detector confidence particle filter. In: Proc. ICCV. (2009)
16. Grabner, H., Bischof, H.: On-line boosting and vision. In: Proc. CVPR. Volume 1. (2006) 260–267
17. Stalder, S., Grabner, H., Gool, L.V.: Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition. In: Proc. IEEE WS on On-line Learning for Computer Vision. (2009)
18. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proc. CVPR. Volume 1. (2005) 886–893
19. Yu, J., Amores, J., Sebe, N., Radeva, P., Tian, Q.: Distance learning for similarity estimation. IEEE Trans. on PAMI (2008)
20. Hertz, T., Bar-Hillel, A., Weinshall, D.: Learning distance functions for image retrieval. In: Proc. CVPR. Volume 2. (2004) 570–577
21. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. CVPR. Volume I. (2001) 511–518
22. Prisacariu, V., Reid, I.: fasthog - a real-time gpu implementation of hog. Technical Report 2310/09, (Department of Engineering Science, Oxford University)