Distributed Averages of Gradients (DAG): A Fast Alternative for Histogram of Oriented Gradients

M. Hossein Mirabdollah, Mahmoud A. Mohamed(✉), and Bärbel Mertsching

GET Lab, University of Paderborn, 33098 Paderborn, Germany
{mirabdollah,mahmoud,mertsching}@get.upb.de
http://getwww.upb.de

Abstract. We propose a compact descriptor for the purpose of dense image matching and object recognition. The descriptor is calculated based on local gradients about each point in an image. It contains the averages of gradients at four different windows surrounding a center point. The descriptor is calculated much faster than histogram of oriented gradients (HOG). Additionally, it will be shown that it is more discriminative than HOG. We used the new descriptor for two applications needed in RoboCup competitions very often. First, computation of dense optical flows and 3D scene reconstruction from two views. Second, human face detection.

1 Introduction

Finding local descriptors for image points is an active and challenging field of research in image processing and computer vision communities. In this regard, several works can be named, suggesting different types of local descriptors for sparse feature matching or dense matching purposes. For the sparse feature matching, the following methods can be remarked: SIFT [11], SURF [2], BRIEF [4], ORB [17], BRISK [10], LIOP [19] and MRRID [7]. An evaluation of these methods can be found in [1,12]. According to these works, although SIFT and SURF are relatively old methods, they still outperform many of the new methods by delivering higher rates of correct matches. Nevertheless, the SURF method is known to perform poorly in case of in plane rotations. We found out that this is due to a simple neglect in the calculation of the SURF descriptor, where the averaged vectors are not projected on the axis of oriented coordinate frames.

SIFT descriptors are generated based on the histogram of oriented gradients (for 8 bins) at 16 windows surrounding a point. In case of the SURF method, the sum of gradient components and the sum of absolute values of gradient components in 16 windows about a point are used to form a descriptor vector with the length of 64. Unfortunately SIFT and SURF are slow methods and are not appropriate for real time applications. Hence, later, methods based on binary descriptors such as BRIEF and ORB were proposed to meet real time requirements. The BRIEF descriptor is generated based on pairs of test points

S. Behnke et al. (Eds.): RoboCup 2016, LNAI 9776, pp. 97–108, 2017.
https://doi.org/10.1007/978-3-319-68792-6_8

in a neighborhood of a point. As BRIEF is not rotation invariant, later ORB was proposed to achieve the rotation invariant version of BRIEF. In the BRISK method, binary patterns are extracted based on a regular radial symmetric pattern about a point. In this method the intensities of the neighborhood of the point are smoothed with Gaussian kernels of different sizes. These consideration make the precision of BRISK slightly better than BRIEF and ORB. Binary descriptors are fast to create, and also in the matching process, the Hamming distance (XOR operation) and bit counting can be used to speed up the matching process dramatically.

The aforementioned methods work well for sparse matching purposes; nevertheless, the good performances are due to the long descriptors created based on large neighborhoods of features. However, as we may need to use feature descriptors for other purposes such as dense image matching, object and action recognition, using such long descriptors demands high computational loads. In literature, compact descriptors such as census [20], MLDP [14] and HOG [5] were used for the computation of dense optical flows. HOG is also used in object and human action recognition. Among the mentioned descriptors, HOG showed a promising performance because it considers both the magnitudes and the directions of gradients in a local window [16]. Nevertheless, the computation of HOG is expensive as it necessitates the calculation of gradient angles. On the other hand, in HOG the spacial distribution of gradients is neglected, which degrades its discrimination. In this paper, we propose a new compact descriptor with the length of 8 bins, inspired from the SURF descriptor. We name the descriptor distributed averages of gradients (DAG), which outperforms HOG both in the sense of computation time and also discrimination.

The paper is structured as follows: In Sect. 2, the gradient based descriptors are reviewed. Section 3 describes how DAG is constructed. In Sect. 4, the application of DAG for the computation of dense optical flow and face detection is presented. DAG is evaluated in Sect. 5. Section 6 concludes this paper.

2 Gradient-Based Descriptors

In this section SIFT and SURF descriptors are reviewed as they are tightly related to HOG and DAG. In the SIFT method, HOG descriptors at 16 different windows surrounding a feature point are calculated. Then descriptors are formed by ordering the values of bins in vectors with the lengths of 128. Calculation of HOG is based on local gradients of an image about a point in a window. Given an image $I(x,y) : \Omega \rightarrow \mathcal{R}$, gradient vectors for a pixel (x,y) are computed as follows:

$$\mathbf{v}(x,y) = [v_x \ v_y]^T = [I_x(x,y) \ I_y(x,y)]^T \tag{1}$$

where $I_x = \frac{\partial I}{\partial x}$ and $I_y = \frac{\partial I}{\partial y}$. To form HOG, the magnitudes m and angles α of gradients should be calculated:

$$\alpha = atan2(v_y, v_x)$$
$$m = \sqrt{v_x^2 + v_y^2} \tag{2}$$

For the calculation of HOG for a point such as (x, y), the gradients in a rectangular neighborhood of the point is taken into account. In this regard, n bins $(b_1 \dots b_n)$ spanning 0° … 360° should be created. The value of each bin is formed as follows:

$$b_i = \sum_{\beta_i < \alpha_j < \beta_{i+1}} m_j \tag{3}$$

where $\beta_i = \frac{360}{n}(i - 1)$. A popular number for bins is eight, which has been used in SIFT. HOG has two main shortcomings: first it needs relatively high computation loads for the calculation of gradient angles. Second it discards the geometry of the occurrence of gradients. Obviously, in SIFT, HOG descriptors are calculated at different sub-windows about a feature, which means that the geometry of gradients will also be taken into account in another way.

Unlike SIFT, SURF descriptors are generated by averaging gradient components and also their absolute values to form a vector containing of 4 elements $[\sum v_x, \sum v_y, \sum |v_x|, \sum |v_y|]^T$. By calculating such a vector for 16 surrounding windows about a point, a descriptor vector with the size of 64 is obtained.

3 Distributed Averages of Gradients

As mentioned in the introduction section, long descriptors of the feature matching methods are keys to achieve high correct matching rates. However, in case of dense matching based on differential techniques, using such long descriptors gives rise to very high computation loads. On the other hand, compact versions of the descriptors mostly with the lengths of eight have been used for dense optical flow calculations. In this regard, we propose a compact descriptor inspired by the SURF descriptor, in which the averages of gradients in only four surrounding windows about each pixel are utilized (Fig. 1).

Additionally, unlike SURF which utilizes only small sub windows, in our proposed descriptor, window sizes can be changed but the number of windows remains always four. Additionally, these four windows are overlapping which make them robust against abrupt changes of the gradients at the borders. Furthermore, to keep the descriptor compact, we did not utilize the average of absolutes of gradient components. Instead the average of gradients for each sub window $w_i : i = 1, 2, 3, 4$ is simply calculated as follows:

$$\mathbf{v}_i = [v_{i,x} \; v_{i,y}]^T = \frac{1}{N} \sum_{(x,y) \in w_i} \mathbf{v}(x, y) \tag{4}$$

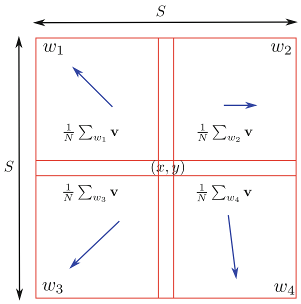
S
S
w_1
w_2
$\frac{1}{N}\sum_{w_1}\mathbf{v}$
$\frac{1}{N}\sum_{w_2}\mathbf{v}$
(x,y)
$\frac{1}{N}\sum_{w_3}\mathbf{v}$
$\frac{1}{N}\sum_{w_4}\mathbf{v}$
w_3
w_4

Fig. 1. DAG construction. Each arrow indicates the average of gradients in a window.

where $N = (S/2+1)^2$. By concatenation of the four vectors, a descriptor vector as following is formed:

$$\mathbf{d} = [v_{1,x}\ v_{1,y}\ v_{2,x}, v_{2,y}, v_{3,x}, v_{3,y}, v_{4,x}, v_{4,y}]^T \tag{5}$$

The descriptor vector can also be normalized to achieve robustness against illumination changes. We abbreviate the normalized version as NDAG.

3.1 Rotation Invariant DAG

To make DAG rotation invariant, a normal vector in the direction of the average of gradients in a neighborhood of each point is taken into account: $\mathbf{g} = [g_x\ g_y]^T$. We also use the orthogonal vector to $\mathbf{g}$, namely $\mathbf{k}$, to form a local coordinate system based on $\mathbf{g}$ and $\mathbf{k}$. The four windows about the keypoint is scanned using four sets of vectors:

$$\{\mathbf{g}_1 = -\mathbf{g},\ \mathbf{k}_1 = \mathbf{k}\}, \{\mathbf{g}_2 = \mathbf{g},\ \mathbf{k}_2 = \mathbf{g}\}, \{\mathbf{g}_3 = -\mathbf{g},\ \mathbf{k}_3 = -\mathbf{g}\}, \mathbf{g}_4 = \mathbf{g},\ \mathbf{k}_4 = -\mathbf{g}\} \tag{6}$$

To address pixels in surrounding windows, we simply use the following equation:

$$\mathbf{x} = [x\ y]^T = h\mathbf{g}_i + w\mathbf{k}_i \tag{7}$$

where i is the index of a window, $h = 0, \ldots, \frac{S}{2}+1$ and $w = 0, \ldots, \frac{S}{2}+1$. After the calculation of the averages of gradients in four windows, it is important to project the averages on the axis of the rotated coordinate system ($\mathbf{g}$ and $\mathbf{k}$). This step is not done in the SURF method which gave rise to its poor performance in case of in plane rotations.

4 Applications of DAG

In this section two applications of DAG are proposed: first for the computation of dense optical flow, and second for face detection.

4.1 Dense Optical Flow Using DAG

We apply DAG for the computation of the optical flow. To this end, we use combined local global methods proposed in [16]. Given two images I and I', DAG descriptors are extracted for all pixels in both images. As a result, for each image, one image with 8 channels is created to store the eight components of DAG descriptor. We name the 8-channel images S and S' associated to the images I and I' respectively. Consequently, we define a cost function consisting of data and regularization terms as explained in [16]:

$$\min_{u,v} \quad E(u,v) = \sum_{\Omega} \left(\lambda E_{data} + \gamma E_{smooth} + E_{dual} \right), \tag{8}$$

where

$$E_{data} = \rho(x,y,u,v) \tag{9}$$

$$E_{smooth} = \parallel \nabla u \parallel + \parallel \nabla v \parallel \tag{10}$$

$$E_{dual} = \frac{1}{2\theta}(u - \hat{u})^2 \tag{11}$$

where λ, γ and θ are the importance weights for each term. $\rho(.)$ is a similarity function as follows:

$$\rho(x,y,u,v) = \sum_{i=1}^{8} \left(S_i'(x+u, y+v) - S_i(x,y) \right)^2 \tag{12}$$

where i is the channel index. To calculate optical flows, the cost function in Eq. 8 can be minimized based on the dual variable technique explained in [16].

4.2 DAG for Face Detection

One of the main task of a rescue robot is exploring a disaster environment to find living survivors. In this regard, the robot should be able to distinguish between the victims and other objects. Obviously, human face is an important feature by which victims can be determined more reliably comparing to other features. We have implemented a face detection framework which can work based on different types of descriptors. In this framework, a descriptor for a region of interest is extracted. Then a SVM (support vector machine) classifier for two classes, face and non-face, is trained. To achieve better detection rates, in the framework a multi-camera system is used, which consists of thermal and video cameras. The thermal images are segmented based on a temperature threshold to provide a list of regions which may contain faces. We align the thermal images with the video camera images using a simple calibration process. Thus, for each region in the thermal image, a corresponding region in the video image can be found (see Fig. 2). For each region in an image from the video camera, a descriptor based on DAG is generated. In this regard, each region is divided into regular blocks of the size $N \times N$. Then for each block a DAG descriptor (or other descriptors) is generated. Afterwards, the descriptors are concatenated to form a global descriptor for each region. The extracted feature vector contains fine detailed information of an image such as edges, spots, corners and other local texture features.

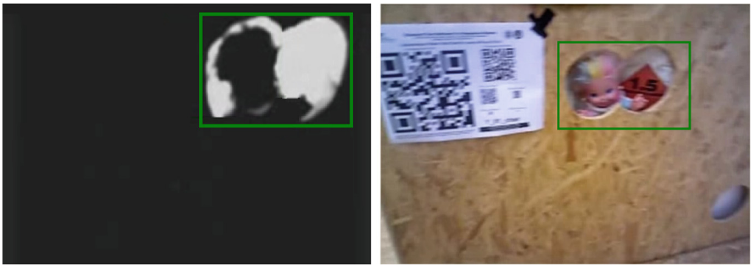

Fig. 2. Face detection using thermal and video images.

5 Evaluation and Experimental Results

5.1 Sparse Matching

To evaluate the discriminating ability of DAG, we used the KITTI training dataset [8] for optical flows. This dataset includes 194 pairs of images and provide the ground truth of flows between each two images. We compared DAG, normalized DAG (N-DAG), HOG, normalized HOG (NHOG) and ORB as a baseline method. The reason that we selected ORB was that we wanted to evaluate the performance of different methods given the same inputs. In this regard, we extracted Shi-Thomasi features [18] at eight levels of pyramids with the scale factor 0.8 and computed descriptors for the features based on different methods. The total average of extracted features was 3953. We used two measures for the evaluation:

$$\text{precision} = \frac{\text{number of correct matches}}{\text{number of matches}}$$

$$\text{recall} = \frac{\text{number of correct matches}}{\text{number of all features to be matched}}$$

To have a uniform matching strategy, we used 2 nearest neighbor (2-nn) matching method. In this method, for each feature in the first image, namely $\mathbf{f}_1$ with the descriptor $\mathbf{d}_1$, two matched features in the second image, namely $\mathbf{f}_{2,1}$ and $\mathbf{f}_{2,2}$, with the descriptors $\mathbf{d}_{2,1}$ and $\mathbf{d}_{2,2}$ are found. If the Euclidean distances of the descriptors are $d_1 = ||\mathbf{d}_1 - \mathbf{d}_{2,1}||$ and $d_2 = ||\mathbf{d}_1 - \mathbf{d}_{2,2}||$, a matching is assumed to be correct if $d_1/d_2 < 0.8$; otherwise, no matching is considered for $\mathbf{f}_1$.

We conducted two experiments to evaluate the performance of upright and rotated descriptors. In the first experiment, we evaluated the upright versions

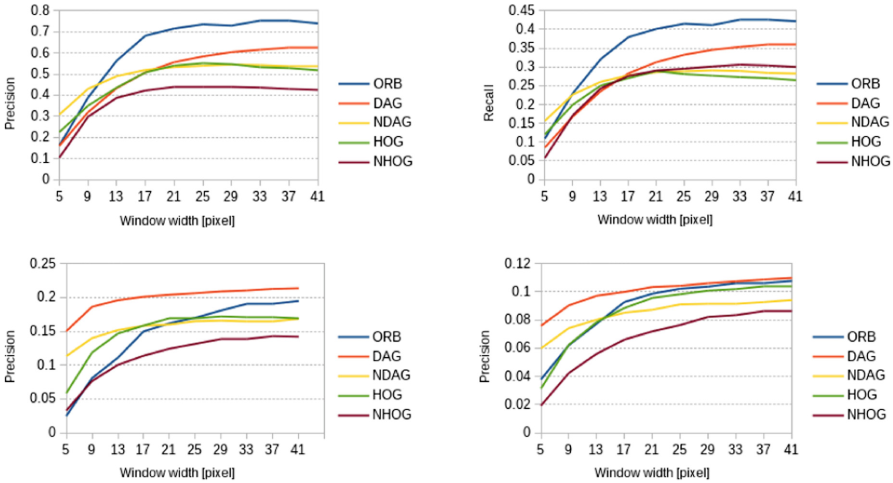

Fig. 3. Precision and recall curves for high quality features (top) and low quality features (bottom).

of all of the methods for two different groups of features: first, features with the qualities more than 0.005 and second features with the qualities between 0.001 and 0.005. We would like to see how the methods work for features with good and bad qualities. It is an important issue for dense matching purposes as typically most of the points have low gradient responses. In Fig. 3, the precision and recall curves for different methods with respect to the window widths can be seen. We can see that for small window sizes NDAG performs much better than the other methods but for larger windows ORB has better performance. Comparison of DAG and HOG shows that the precision and recall of DAG increase more and more as the window size increases; whereas the performance of HOG after the window size 21 degrades. It signifies that DAG is more capable to capture local informations in comparison to HOG at bigger window sizes. The reason that NDAG works better than DAG is that in small windows illumination changes affect the average of gradients adversely. Therefore, as normalization of DAG makes it robust against illumination changes, better performance of NDAG is reasonable. On the other hand, HOG is already robust against illumination changes thanks to its binning process. Therefore, its normalization only leads to losses of information.

Concerning the performances of the methods for low quality features, we see that DAG outperforms all other methods. The poor performance of ORB lies in the fact that most of the low quality features are located on almost homogeneous regions such as roads and in this case the binary descriptors are more vulnerable against the measurement noise rather than the gradient based methods.

In the second experiment, we rotated the second images of each image pairs to evaluate rotation invariance of DAG. In this experiment, we also ran the SIFT method, which might be interesting to readers how it works for these sequences.

Unfortunately, SURF had a very poor performance and was not comparable with any of the other methods. Therefore, we did not take it into account in this experiment. Figure 4 depicts the comparison results. We see that all methods except SIFT experiences relatively large drops at the angles 45°k ($k = 1, \ldots 7$). Nevertheless, ORB and then DAG have the best performances. Surprisingly, SIFT has a relatively poor performance. We investigated the problem and noticed that it originates from the nature of outdoor images in the KITTI dataset. In these images, repeatability of blob features are low, which gives rise to missing many of candidates for correct matching.

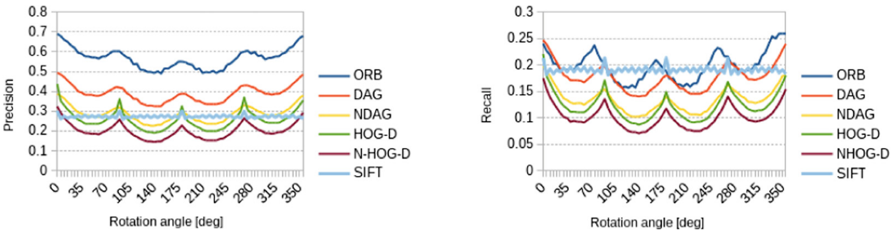

Fig. 4. Precision and recall curves in case of in plane rotations.

5.2 Computation of Dense Optical Flow and 3D Scene Reconstruction

We applied the DAG descriptor to compute dense optical flows for the KITTI [8] dataset as explained in Sect. 4.1. The KITTI dataset provides a very challenging testbed for the evaluation of optical flow algorithms. Pixel displacements in the data set are generally large, the images exhibit less texture regions, strongly varying lighting conditions, and many non-Lambertian surfaces, especially translucent windows and specular glass, and metal surfaces. Moreover, the high speed of the forward motion creates large regions on the image boundaries that move out of the field of view between frames, such that no correspondence can be established.

For evaluating the estimated optical flow, we calculated the average end-point error AEE, the average angular error AAE, and the percentage of pixels with an AEE of more than 3 pixels, which also known as bad pixels [8]. In all experiments, we used the fixed point iteration algorithm to optimize the objective function in Eq. 8. In order to deal with large displacement optical flows, we used the coarse to fine technique [3]. Furthermore, we used the normalized version of HOG and the normalized version of DAG. Because the normalized descriptors are robust against illumination changes, and also thanks to the normalization, a constant smoothness parameter can be applied regardless of the magnitudes of gradients. We used the following parameters setup for all descriptors, pyramids scale factor 0.9, the smoothness parameter $\lambda = 10$, outer fixed point iteration was 4 image warp, and inner fixed point iteration was 10 times. For DAG and HOG, we used the window size 7 and 5 respectively, which yielded the best results.

For KITTI training dataset, Table 1 shows the average AEE and the average of the percentage of bad pixels for 194 sequences. As shown in Table 1 DAG outperforms HOG in the AEE but the percentage of bad pixels for HOG is slightly better than DAG. Furthermore, Table 2 shows a comparison among DAG, HOG, and MLDP for some sequence of KITTI training dataset. As shown in this table DAG outperforms other descriptors. In Fig. 5, the computed flows for the sequence 0 using different methods are visualized. We can see that DAG works very well in the estimation of flows in low texture regions. It supports the previous results for sparse feature matching.

One of the main application of dense optical flow is 3D scene reconstruction based on the structure from motion techniques. These techniques are especially important for outdoor environments, where RGB-D cameras cannot be used due to their limitations an also 3D range sensors cannot be an option due to their high prices. In Fig. 5, the reconstruction of the scene for sequence 0 of KITTI is visualized. We used the techniques proposed in [13] for the camera motion estimation and the 3D scene reconstruction. We observe that the structure of the scene at many places is reconstructed well. Especially, the ground plane is rebuilt very well, showing the ability of the combined techniques in providing fine information for terrain analysis purposes.

We have also computed the optical flows based on the DAG descriptor for the KITTI test dataset and submitted to KITTI website under the name of DAG-Flow. Table 3 shows a comparison among algorithms used local descriptors for optical flow estimation.

Table 1. Average of the AEE and the average of the percentage of outliers for the KITTI training dataset.

	Average AEE	Average percentage of outliers
DAG	2.16	8.16
HOG	2.19	8.12
MLDP	2.85	8.89

Table 2. The AEE and the percentage of bad pixels for some sequences of KITTI training dataset.

Sequence no	0		14		25		101		115		131		139		141	
	AEE	%AEE	AEE	%AEE	AEE	%AEE	AEE	% AAE	AEE	%AEE	AEE	%AEE	AEE	%AEE	AEE	%AEE
DAG	**1.69**	**9.01**	**8.80**	**17.67**	**7.12**	**23.00**	**8.74**	**21.97**	**0.75**	**3.88**	**1.40**	**8.43**	**1.10**	**5.50**	**4.63**	**14.85**
HOG [16]	3.07	13.16	8.98	22.10	7.63	26.99	8.56	26.91	2.05	8.27	2.10	13.29	3.21	10.27	6.53	19.64
MLDP [14]	4.39	19.67	9.54	22.16	9.48	35.76	17.84	39.45	1.30	7.72	2.08	14.42	1.42	8.26	9.09	23.17

The average elapsed time for the computation of the optical flow for each pair of images were obtained. The optimization algorithm model ran on an Intel

Table 3. KITTI test dataset. The average of the AEE and the average of the percentage of outliers for some methods using local descriptors for the optical flow estimation.

	Average AEE	Average percentage of outliers
TVL1-HOG [16]	2.0	7.91
MLDP-OF [14]	2.4	8.67
CRTflow [6]	2.7	9.43
TGV2CENSUS [15]	2.9	11.03
DAGflow	2.3	8.41

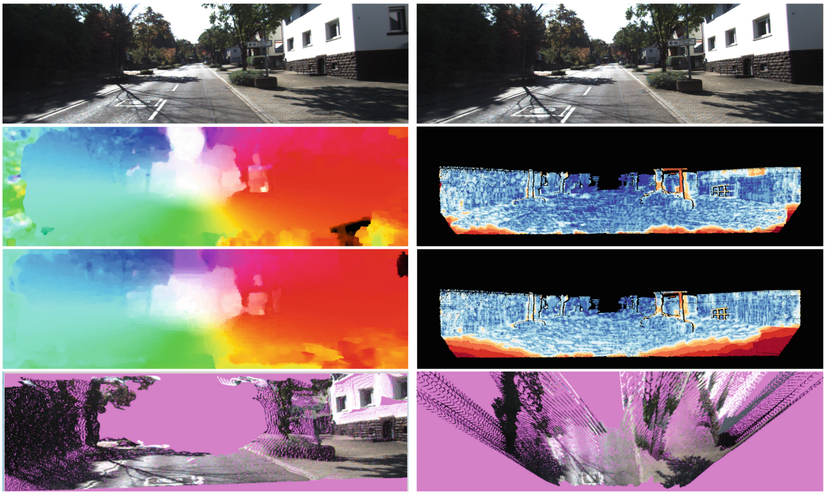

Fig. 5. KITTI training data set. Row 1: sequence 000000_10 and frame 000000_11. Row 2: color coding of the estimated optical flow and the error map using DAG. Row 3: estimated optical flow and error map using HOG [16] descriptor. Row 4: reconstructed scene from two views. (Color figure online)

Core2Duo with a 2.5 GHz processor executing C++ codes. The computation time for DAG was 12.10 s, while for HOG and MLDP was 22.70 s and 16.26 s respectively.

5.3 Face Detection

For evaluating the performance of DAG descriptor for the face detection, we have used the Frontal Face Dataset from the Caltech 101 categories [9]. Caltech data set contains images for 101 objects. The face dataset contains 450 images with the resolution of 896 × 592 pixels. The images were captured from 27 people in different lighting conditions, various face expressions and different backgrounds. In our setup, every image is partitioned into 15 × 15 block. For training and testing the detection algorithm, we used positive images from the Frontal face dataset, while images from other objects as negative images. For training the SVM classifier, we used 100 positive images and 100 negative images. We tested the algorithm using 450 positive image and 450 negative images. We evaluated the accuracy of the algorithm by calculating the false positive and false negative ratios. Interestingly, the face detection algorithm based on DAG detected all the 450 faces correctly and the number of false detection was always zero; whereas, the algorithm based on HOG failed to detect some faces (false negative). In Fig. 6 some samples of faces are presented, which the SVM classifier based on HOG failed to detect.

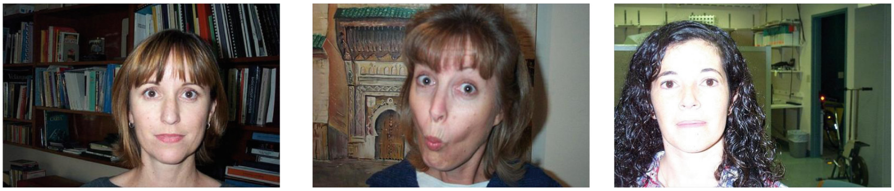

Fig. 6. Samples of false detection of SVM classifier based on HOG feature vector. Left frame 88, middle frame 193, and right frame 338.

6 Conclusion

In this paper a new compact descriptor for the purpose of dense image matching and object recognition is proposed. The descriptor is calculated based on local gradients of each pixel in an image by calculating the average gradients at four different regions surrounding a center point. The descriptor can be calculated much faster than histogram of oriented gradients as there is no need for calculation of the angle of each gradient and also the binning step is skipped. Additionally, it was shown that the proposed descriptor is more discriminative than HOG based on two different types of experiments conducted in this paper.

References

1. Opencv feature descriptor comparison report (2015). http://computer-vision-talks.com/articles/2011-08-19-feature-descriptor-comparison-report/. Accessed 23 Mar 2016
2. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (SURF). CVIU **110**(3), 346–359 (2008)
3. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: Pajdla, T., Matas, J. (eds.) ECCV 2004. LNCS, vol. 3024, pp. 25–36. Springer, Heidelberg (2004). doi:10.1007/978-3-540-24673-2_3

4. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: BRIEF: binary robust independent elementary features. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6314, pp. 778–792. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15561-1_56
5. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceedings of CVPR, pp. 886–893 (2005)
6. Demetz, O., Hafner, D., Weickert, J.: The complete rank transform: a tool for accurate and morphologically invariant matching of structure. In: Proceedings of BMVC (2013)
7. Fan, B., Wu, F., Hu, Z.: Rotationally invariant descriptors using intensity order pooling. TPAMI **34**(10), 2031–2045 (2012)
8. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: the KITTI dataset. IJRR 1231–1237 (2013)
9. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In: Workshop on Generative-Model Based Vision CVPR (2004)
10. Leutenegger, S., Chli, M., Siegwart, R.Y.: BRISK: binary robust invariant scalable keypoints. In: Proceedings of ICCV, pp. 2548–2555 (2011)
11. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV **60**(2), 91–110 (2004)
12. Miksik, O., Mikolajczyk, K.: Evaluation of local detectors and descriptors for fast feature matching. In: Proceedings of ICPR, pp. 2681–2684 (2012)
13. Mirabdollah, M.H., Mertsching, B.: Fast techniques for monocular visual odometry. In: Gall, J., Gehler, P., Leibe, B. (eds.) GCPR 2015. LNCS, vol. 9358, pp. 297–307. Springer, Cham (2015). doi:10.1007/978-3-319-24947-6_24
14. Mohamed, M.A., Rashwan, H.A., Mertsching, B., Garcia, M.A., Puig, D.: Illumination-robust optical flow using local directional pattern. IEEE Trans. Circ. Syst. Video Technol. **24**, 1–9 (2014)
15. Ranftl, R., Gehrig, S., Pock, T., Bischof, H.: Pushing the limits of stereo using variational stereo estimation. In: Proceedings of IV, pp. 401–407 (2012)
16. Rashwan, H.A., Mohamed, M.A., García, M.A., Mertsching, B., Puig, D.: Illumination robust optical flow model based on histogram of oriented gradients. In: Weickert, J., Hein, M., Schiele, B. (eds.) GCPR 2013. LNCS, vol. 8142, pp. 354–363. Springer, Heidelberg (2013). doi:10.1007/978-3-642-40602-7_38
17. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to SIFT or SURF. In: Proceedings of ICCV, pp. 2564–2571 (2011)
18. Shi, J., Tomasi, C.: Good features to track. In: Proceedings of CVPR, pp. 593–600 (1994)
19. Wang, Z., Fan, B., Wu, F.: Local intensity order pattern for feature description. In: Proceedings of ICCV, pp. 603–610 (2011)
20. Zabih, R., Woodfill, J.: Non-parametric local transforms for computing visual correspondence. In: Eklundh, J.-O. (ed.) ECCV 1994. LNCS, vol. 801, pp. 151–158. Springer, Heidelberg (1994). doi:10.1007/BFb0028345