Histogram of Oriented Gradients and Region Covariance Descriptor in Hierarchical Feature-Distribution Scheme

Vildana Sulić, Janez Perš, Matej Kristan, Stanislav Kovačič
Faculty of Electrical Engineering, University of Ljubljana
Tržaška 25, SI-1000 Ljubljana
E-pošta: vildana.sulic@fe.uni-lj.si

Abstract

Hierarchical feature-distribution scheme is a recently proposed framework for distribution of features in visual-sensor networks. It is intended for tasks, where one needs to establish a correspondence between two objects, seen by different cameras at different occasions. In visual-sensor networks, such pair of cameras may be very distant in network terms. Therefore, the hierarchical scheme results in significant reduction of network traffic, compared to naive approaches, which rely on flooding. In this paper we explore the performance of two state-of-the-art feature descriptors (histogram of oriented gradients and region covariance descriptor) in such feature-distribution scheme. Both methods are compared in the terms of network load on the COIL-100 data set. Results show that even state-of-the-art feature descriptors benefit from hierarchical feature-distribution scheme.

1 Introduction

In visual systems we usually deal with large amounts of digital image data. Data has to be archived or exchanged between numerous users and systems [1], consuming expensive resources, such as storage space or transmission bandwidth. This problem is especially important in visual-sensor networks (VSN), where data-intensive modality (images) is usually combined with network connections of limited capacity and usually, limited range.

In our previous work [2, 3] we focused on problem of object recognition in VSN. In our case, we define distributed object recognition as follows: *given the acquired image of an object, find all the images of visually similar objects that have been acquired by any of the nodes on any previous occasion.* In visual-sensor network (VSN), a single node (camera) does not have all the relevant features to make such decision. Features, belonging to the objects seen by the other network nodes reside in local memory of those nodes. If particular node wants to recognize previously seen object, all features from the network have to be requested for comparison. This way, distributed feature comparison results in non-negligible amount of network traffic.

The result of our research [2, 3] was hierarchical feature-distribution (HFD) scheme for object recognition in a network of visual sensors, which utilizes network in a more balanced way than trivial network flooding. In a nutshell, the HFD for VSNs is based on hierarchical distribution of the information, where each individual node retains only a small amount of information about the objects seen by the network. However, this amount is sufficient to efficiently route queries through the network without any degradation in the recognition performance. The amount of data transmitted through the network can be significantly reduced using our hierarchical distribution, as demonstrated in [3].

HFD does not rely on particular object recognition method or a particular feature descriptor. It only provides the algorithm for feature distribution during the learning phase and corresponding algorithm for feature routing during the recognition phase. It also specifies the requirements, which have to be fulfilled by particular recognition method to be used in our distributed scheme. Those requirements concern abstraction, storage space, existence of metric and convergence [2, 3]. We already confirmed the fulfillment of those requirements for several basic recognition methods (template matching, histogram matching, PCA and random projection [4]). The performance of those methods within HFD scheme was also established experimentally. It was demonstrated that HFD indeed results in significant savings in network traffic, while preserving recognition rates.

In this paper we applied HFD scheme to two state-of-the-art feature descriptors, that is histogram of oriented gradients (HOG) [5] and region covariance (COV) descriptor [6]. The remainder of this paper is organized as follows. In the Section 2 we provide theoretical background of both methods, along with the implementation details. Experiments in the VSN simulator and the results of tests are reported and discussed in the Section 3. Section 4 concludes the paper.

2 HOG and COV descriptors

In this section we briefly present two region descriptors that we used in our work. In accordance with requirements of HFD scheme [2] we also define the appropriate mappings $f : \mathbf{x}^{(n)} \mapsto \mathbf{x}^{(n+1)}, 0 < n \leq N$, which translates a level n feature vector $\mathbf{x}^{(n)}$ into a higher, more abstract, level $(n+1)$ feature vector $\mathbf{x}^{(n+1)}$. N denotes the highest level of abstraction. We also define the metrics $d^{(n)}(\mathbf{x}_1^{(n)}, \mathbf{x}_2^{(n)})$, which provides a measure of the *similarity* between two feature vectors $\mathbf{x}_1^{(n)}$ and $\mathbf{x}_2^{(n)}$ of the same level n.

2.1 Histogram of Oriented Gradients

HOG features have been introduced by Dalal and Triggs in [5]. Authors have shown that HOG descriptors significantly outperform other feature sets, such as Haar wavelets. They have studied influence of several variants of HOG descriptors (R-HOG and C-HOG), with different gradient computation and normalization methods. HOG descriptors are based on the idea that local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions. Implementation of these descriptors in practice is as follows: the image window is divided into small spatial regions (cells) and for each cell a 1D histogram of gradient directions or edge orientations is accumulated over the pixels within the cell. The combination of these histograms then represents the descriptor [7]. For better performance (e.g., invariance to the illumination or shadowing) the local histograms can be contrast normalized.

HOG descriptor is based on the first order derivatives with respect to x and y of the image intensity (denoted by I_x and I_y). From these derivatives, a gradient field is computed assigning to each pixel a magnitude $mg(x, y)$ and an angle $\Theta(x, y)$ [8]:

$$mg(x, y) = \sqrt{I_x^2(x, y) + I_y^2(x, y)} \tag{1}$$

$$\Theta(x, y) = \arctan(\frac{I_y(x, y)}{I_x(x, y)}). \tag{2}$$

A histogram is formed where each bin is the sum of all magnitudes with the same orientation on in a given region. Histograms are compared using Hellinger distance.

Implementation details An implementation which follows the original publication exactly [9] was used. Experiments were run with default parameters as specified in [6].

Requirements Histograms form a basis of HOG descriptors, therefore, the mapping $f : \mathbf{x}^{(n)} \mapsto \mathbf{x}^{(n+1)}$ and the metric $d^{(n)}(\mathbf{x}_1^{(n)}, \mathbf{x}_2^{(n)})$ can be the same as in the plain histogram matching, as shown in [2]. Such choice also fulfills the four HFD requirements [2].

2.2 Region covariance descriptor

COV descriptor was first presented by Tuzel et al. [6] and was shown to outperform histogram descriptors [8].

Let I be a one dimensional intensity or three dimensional color image. Let F be the W×H×d dimensional feature image extracted from I:

$$F(x, y) = \Phi(I, x, y), \tag{3}$$

where the function Φ can be any mapping such as intensity, color, gradients, filter responses, etc. For a given rectangular region $R \subset F$, let $\{z_k\}_{k=1...n}$ be the d-dimensional feature points inside R. The region R with the $d \times d$ covariance matrix of the feature points is represented as:

$$C_R = \frac{1}{n-1} \sum_{k=1}^{n} (z_k - \mu)(z_k - \mu)^T, \tag{4}$$

where n is the number of points in the region, and μ is the mean of the points.

For the distance calculation on covariance matrices we used the distance measure proposed by Förstner and Moonen in [10]:

$$\rho(C_1, C_2) = \sqrt{\sum_{i=1}^{n} \ln^2 \lambda_i(C_1, C_2)}, \tag{5}$$

where $\lambda_i(C_1, C_2)_{i=1...n}$ are the generalized eigenvalues of C_1 and C_2, computed from

$$\lambda_i C_1 x_i - C_2 x_i = 0, \quad i = 1...d, \tag{6}$$

and $x_i \neq 0$ are the generalized eigenvectors [10]. Using distance formulated in Eq.(5), the dissimilarity of two covariance matrices was measured.

Implementation details First, for each pixel, a nine-dimensional feature vector f_n was extracted.

$$f_n = [x, y, I, I_x, I_y, I_{xx}, I_{yy}, mg, \Theta]^T, \tag{7}$$

where x, y are pixel location, I is the grayscale intensity, I_x, I_y are the norms of the first order derivatives and I_{xx}, I_{yy} are the norms of the second order derivatives. mg and Θ are defined as in Eq.(1) and Eq.(2).

The covariance of a region is computed as shown in Eq.(4). This COV matrix is unwrapped into the feature vector $\mathbf{x}^{(0)}$, and vectors f_n are discarded. From this point on, feature vectors $\mathbf{x}$ are used in HFD in similar way than all other types of features.

In preliminary tests, we evaluated the influence of each of the nine components of the covariance descriptor on recognition performance(Figure 1). This

was done by dropping one or two of the nine components and running recognition tests with the remaining eight or seven components (horizontal and vertical components of the same type were dropped together). This way an approximation of relative importance of each component of covariance descriptor was obtained.

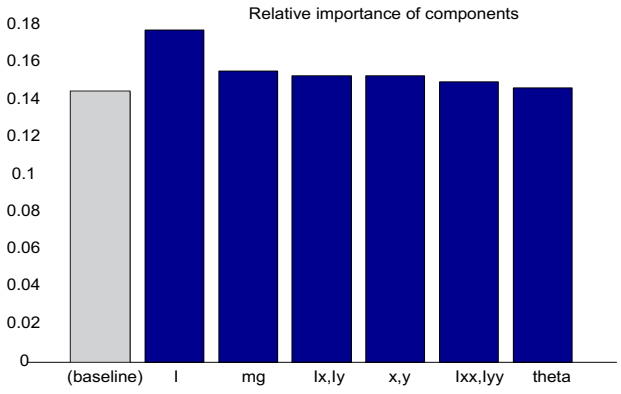

Figure 1: Relative importance of each of the nine components of the covariance descriptor, expressed as the recognition error on removing the component from the descriptor. Leftmost bar shows a baseline value - recognition error, when all components are used. Data was obtained on COIL-100 data set.

Requirements Following the requirements defined in [2] the mapping $f : \mathbf{x}^{(n)} \mapsto \mathbf{x}^{(n+1)}$ and the metric $d^{(n)}(\mathbf{x}_1^{(n)}, \mathbf{x}_2^{(n)})$ can be defined as follows.The mapping $f : \mathbf{x}^{(n)} \mapsto \mathbf{x}^{(n+1)}$ is defined as dropping a feature, which has the lowest effect on recognition performance of the method. This obviously results in the decrease of the required storage space, therefore, Requirement 2 [2] is fulfilled as well. The metric $d^{(n)}(\mathbf{x}_1^{(n)}, \mathbf{x}_2^{(n)})$ can be the distance proposed by Förstner and Moonen (Eq.(5)).

We did not examine the fulfillment of Requirement 4 [2], which guarantees that the recognition performance remains the same when a method is used in HFD scheme. Instead, we rely on the results of experimental testing to see the impact of the HFD on the recognition rate.

3 Experimental setup

We performed a series of experiments using both descriptors. The recognition performance of both methods in terms of percentage of false positives and false negatives on a COIL-100 data set [11] was examined. We used both methods in the VSN simulator [3] in conjunction with three types of feature-distribution methods. Following the same protocol as in [3] the proposed HFD scheme, denoted $\mathbf{M}_{\text{hier}}$ was compared to two flooding based (naive) feature-distribution methods, denoted $\mathbf{M}_{\text{naive1}}$ and $\mathbf{M}_{\text{naive2}}$, respectively.

Simulator To test the performance of both methods in the HFD scheme [2], we used a distributed network simulator. It runs on a standard desktop computer and is written in Matlab. The simulator measures both the amount of traffic transmitted between the nodes and the number of nodes (hops) over which the traffic is transmitted. For the experiments, we used a network consisting of 99 nodes, arranged in a 11×9 rectangular, 4-connected grid.

Experiments Experiment was divided in two phases. The first (learning) phase measured the performance of the network during learning. Twenty nodes, evenly distributed through the network, were injected with images of the 100 different objects from the data set. Those images corresponded to the zero orientation in the COIL-100 data set. Next, the simulation cycle was started, and, after the network traffic stopped, the statistics on the network load (number of hops and the total network traffic per sample) was examined.

The second (recognition) phase measured the performance of the network during recognition. A pseudo-random sequence (same for all tests) was used to choose any image from the data set and any node from the network. The image was injected into the chosen node, and the simulation cycle was started. After the network activity stopped, the result of the recognition was read from the same node, and the statistics on the false positives (FPs) and the false negatives (FNs) were updated. The process of injecting the random image to a random node was repeated 5,000 times, and the statistics on the network load (number of hops and the total network traffic per sample) was recorded. The results for training and recognition for both HOG and COV are shown in Table 1.

Results It can be seen that recognition performance remains the same regardless whether naive or hierarchical feature distribution method is used. This holds both for HOG and COV descriptors. This is not surprising in the case of HOG descriptor, since our implementation of object recognition using HOG descriptor fulfills the Requirement 4 [2]. However, the preservation of recognition rate in the case of COV descriptor indicates that our implementation of COV descriptor fulfills Requirement 4 as well, even though we did not prove this analytically. This property makes both descriptors an appropriate choice for the use in the HFD scheme.

It can also be seen that the HFD outperforms naive methods in terms of network load. In learning stage, where the features are distributed across the network $\mathbf{M}_{\text{hier}}$ drastically lowers the amount of transmitted data per sample in comparison to $\mathbf{M}_{\text{naive2}}$. This holds

Table 1: Experimental results for each combination of recognition methods and feature-distribution methods

Recognition method	Distribution method	Learning Hops $[\frac{1}{sample}]$	Learning Traffic $[\frac{kbytes}{sample}]$	Recognition Hops $[\frac{1}{sample}]$	Recognition Traffic $[\frac{kbytes}{sample}]$	Recognition rate FPs	Recognition rate FNs
HOG	$\mathbf{M}_{\text{naive1}}$	0	0	376	16331	18%	27%
	$\mathbf{M}_{\text{naive2}}$	614	11271	0	0	18%	27%
	$\mathbf{M}_{\text{hier}}$	**614**	**587**	**255**	**8663**	**18**%	**27**%
COV	$\mathbf{M}_{\text{naive1}}$	0	0	323	24	9%	16%
	$\mathbf{M}_{\text{naive2}}$	614	119	0	0	9%	16%
	$\mathbf{M}_{\text{hier}}$	**614**	**15**	**196**	**13**	**9**%	**16**%

for both descriptors. The amount of hops remains unchanged, since features have to reach all the nodes in the learning stage. Essentially, the amount of transmitted data using HFD ($\mathbf{M}_{\text{hier}}$) is negligible in comparison to $\mathbf{M}_{\text{naive2}}$. In recognition stage, the amount of transmitted data is halved when HFD ($\mathbf{M}_{\text{hier}}$) is used. For recognition, $\mathbf{M}_{\text{naive1}}$ is used as a baseline, since it floods the network for each recognition task. It can also be seen that the drop in number of hops during recognition when using HFD is more pronounced in case of COV descriptor. This is, again, expected as the number of hops directly depends on accuracy of the descriptor used.

4 Conclusion

The paper focuses on adaptation of two state-of-the-art computer vision methods to the hierarchical feature-distribution scheme. When image processing and computer vision methods (in our case object recognition) are used in the distributed vision systems, they can easily overload communication-constrained distributed network. For this reason we previously proposed hierarchical feature-distribution scheme (HFD), which utilizes network in a more balanced way than trivial network flooding.

Based on the above observations, we can conclude that COV descriptor is the most appropriate choice for use in the HFD scheme. It is accurate, compact and self-contained (subspace independent). We performed our tests on COIL-100 data set, however, the COV descriptor has been found to be one of the best choices in the more realistic (surveillance) applications as well [8].

References

[1] J. J. Amador. Random projection and orthonormality for lossy image compression. *Image Vision Comput.*, 25(5):754–766, 2007.

[2] V. Sulić, J. Perš, M. Kristan, and S. Kovačič. Hierarchical feature scheme for object recognition in visual sensor networks. *Electrotechnical Review,*, 76(1–2):38–44, 2009. `http://ev.fe.uni-lj.si/1-2-2009/Sulic.pdf`.

[3] V. Sulić, J. Perš, M. Kristan, and S. Kovačič. Efficient feature distribution in visual sensor networks. Technical report, Faculty of Electrical Engneering, University of Ljubljana, 2009. `http://vision.fe.uni-lj.si/docs/danas/TR-FE-LSV-0109.pdf`.

[4] V. Sulić, J. Perš, M. Kristan, and S. Kovačič. Efficient dimensionality reduction using random projection. In *Computer Vision Winter Workshop*, 2010.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2005.

[6] F. Tuzel, O. amd Porikli and P. Meer. Region covariance: A fast descriptor for detection and classification. In *European Conference on Computer Vision (ECCV)*, May 2006.

[7] B. Raluca. Histogram of oriented gradients. Pattern recognition systems – Lab 5-6, 2008.

[8] A. Alahi, P. Vandergheynst, M. Bierlaire, and M. Kunt. Cascade of descriptors to detect and track objects across any network of cameras. *Submitted to Computer Vision and Image Understanding Journal (CVIU)*, 2010.

[9] Hog calculator. `http://hi.baidu.com/timehandle/blog/item/ca6e3cdfab738fe376c638a8.html`, May 2010.

[10] W. Förstner and B. Moonen. A metric for covariance matrices. Technical report, Department of Geodesy and Geoinformatics, Stuttgart University, 1999.

[11] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (coil-100), cucs-006-96. Technical report, Department of Computer Science, Columbia University, 1996.