Elektrotehniški vestnik 76(1-2): 38–44, 2009
Electrotechnical Review, Ljubljana, Slovenija

Hierarchical Feature Scheme for Object Recognition in Visual Sensor Networks

Vildana Sulić, Janez Perš, Matej Kristan, Stanislav Kovačič

Univerza v Ljubljani, Fakulteta za elektrotehniko, Tržaška 25, 1000 Ljubljana, Slovenija
E-pošta: vildana.sulic@fe.uni-lj.si

Abstract. Visual sensor networks are the meeting point of two significantly different technologies: one is image processing with high computation and storage demands, and the other is distributed sensor approach with low power, low computational and storage capabilities. We propose a framework for hierarchical feature encoding scheme for a frequent computer vision task – object recognition. The key of our approach is the principle that individual nodes in the network hold only a small amount of information about objects seen by the network. However, this information is sufficient to efficiently route network query, when a new, unknown object is encountered. A set of criteria has to be fulfilled by the object recognition method to qualify for use in our framework. The paper provides examples of three widely known object recognition approaches that can be easily adapted for use in such hierarchical feature encoding scheme.

Key words: visual sensor networks, computer vision, object recognition

Hierarhična shema značilnic za razpoznavanje objektov v omrežju vizualnih senzorjev

Povzetek. Članek obravnava problematiko uporabe algoritmov računalniškega vida v omrežju vizualnih senzorjev. Tehnologija senzorskih sistemov postavlja resne omejitve v zmogljivosti tako samih senzorjev kot tudi prenosnih poti. Kot rešitev predstavljamo metodo hierarhičnega zapisa značilnic za eno najpogostejših nalog računalniškega vida – razpoznavanje objektov. Hierarhična shema zagotavlja, da ima vsako vozlišče na voljo le majhno količino informacije o opaženih objektih, vendar pa je ta informacija dovolj za učinkovito usmerjanje omrežnih poizvedb, ko je opažen nov, neznan objekt. Članek definira pogoje, ki jih morajo izpolnjevati metode razpoznavanja objektov, da jih je mogoče uporabiti v takšni hierarhični shemi. Kot primer podajamo prilagoditev treh splošno znanih metod razpoznavanja objektov v predstavljeno hierarhično shemo.

Ključne besede: omrežje vizualnih senzorjev, računalniški vid, prepoznava objektov

1 Introduction

Computer vision deals with extracting useful information from images and video sequences. One of the main characteristics of computer vision algorithms is a large amount of data that has to be processed, stored or transmitted. Usually, a state-of-the-art hardware is employed to deal with requirements that go along with processing of digital images and digital video streams. Such approach represents essentially a *star* network topology - a single powerful processing unit and a large number of sensors. In the majority of applications, these sensors are simply ordinary digital cameras, using IEEE1394 or USB interface for example.

Many applications, such as video surveillance, traffic or environment monitoring need a large number of visual sensors. It is not unusual that the number of cameras for modern application goes into hundreds, and the requirements for processing such huge amounts of data are correspondingly large. For such systems, a distributed architecture is not a matter of choice, but a matter of necessity.

Transition to distributed network topology of visual sensors is not straightforward. Due to high processing, transmission and storage requirements of computer vision algorithms, the distributed network would be put under unbearable strain if such algorithms are directly mapped to the network. For example, in detection and recognition of objects that have been previously seen by any of the sensors, one of the following two scenarios would appear in a distributed system:

- Captured visual information (images or extracted features) from each sensor is distributed to all nodes for local storage. Future detection of similar objects is then performed locally by each sensor as new images are acquired.
- Captured visual information is stored locally.

Received 17 July 2008
Accepted 6 December 2008

However, each task of finding an object results in a query, which is broadcasted across the network, for comparison with the locally stored information on each node, for each unknown object encountered.

In both scenarios, an enormous amount of data would have to be transmitted across the network. In fact, it is obvious that each node would need to have processing and storage capabilities similar to the central unit in star topology.

In this paper we propose a framework for hierarchical feature encoding, where each node stores only a small amount of information about an object previously seen by the network. However, this information is sufficient to efficiently route the network query, when a new, unknown object is encountered.

In the remainder of the paper we define a set of criteria which has to be fulfilled by the object recognition method to qualify for use in the proposed framework. We provide examples of three widely known object recognition approaches that can be easily adapted for use in such hierarchical feature encoding framework and show that they fulfill those criteria.

1.1 Related work

Visual sensor networks (VSNs) are the meeting point of two significantly different technologies. On one side there is a distributed sensor approach, which puts significant constraints on available processing power and network transmission capabilities. On the other side there are image processing and computer vision, which are both computationally and data intensive. Therefore, the main issues in VSNs revolve around the task of achieving maximum performance on hardware with limited capabilities.

The first major issue in VSNs is obviously the efficient data transmission between the nodes. Transmission of visual information usually requires a large bandwidth and therefore specifically tailored optimization to the distributed sensor topologies is highly desirable.

Data transmission techniques in VSNs can be roughly categorized into three categories [1]. In the first category, there are efficient image and video transmission methods for transmission over a single hop. As an example in [2], an energy efficient JPEG-2000 image transmission system over VSNs that minimizes the overall processing and transmission energy consumption is proposed.

In the second category, there are techniques that consider the multihop transmission strategy based on a hop-by-hop, such as [3], where authors show that their scheme can greatly improve the image quality at the destination in case of link impairments and node failure.

The third category includes end-to-end multi-path transmission techniques in multihop networks, where multi-path transmission is used to increase reliability. One of many is [4], where a fast algorithm to trade-off between the end-to-end energy cost and reliability requirements of multi-path data transmission is proposed.

The next major issue in VSNs is their distributed nature and the problems that arise from the distributed sensor concept. In [5], decentralized methods for obtaining the vision graph for a distributed camera network are discussed. Authors propose a novel framework to determine image relationships in large networks of cameras. In [6], a fully distributed calibration approach for 3D camera networks, using belief propagation, is proposed. In [7], a novel distributed approach to protect dense VSNs against eavesdropping attacks is proposed.

There is a large body of research concerning implementation of various computer vision tasks on embedded platforms. In [8], an object recognition system with real time capabilities for deployment on a DSP-based embedded platform is proposed. An embedded platform, capable of performing high-level computer vision tasks such as vehicle and license plate detection in real–time is presented in [9]. In [10], a stereo image for object detection on an embedded system is used. In [11], authors proposed a system that uses background subtraction for target detection, 2-D integer-lifting wavelet transform for feature extraction, support vector machine for target classification and auto-regressive moving-average model for target tracking. Those tasks are performed in each sensor node, while multi-view localization algorithm is implemented with collaboration between wireless sensor nodes in a distributed P2P signal processing framework. In [12], authors applied a multi-agent framework to the management of a surveillance system using a VSN. A software agent is embedded in each camera to control the capture parameters. In [13], authors proposed a technique for tracking objects across spatially separated, uncalibrated, non-overlapping cameras. An autonomous multicamera tracking method on distributed embedded smart cameras is presented in [14]. In [15], authors present a distributed network of smart cameras for real-time tracking.

Our approach aims to facilitate efficient distribution of features to a large number of visual sensors and efficient routing of queries to those nodes, which have enough information to perform object recognition. Differently from many other studies, we do not deal with implementation details or low level issues in data transmission. Our approach aims to reduce the total amount of data that has to be transmitted across the network or stored in individual nodes. We intend to achieve this by encoding features in a hierarchical way, where less descriptive features are derived from the original object features. Those less descriptive features are used for efficient routing, while consuming

significantly less network resources than full distribution of original object features.

2 Hierarchical feature scheme for object recognition

Major challenges in VSNs are related to the discrepancy between computationally intensive image processing in computer vision algorithms and relatively low processing capabilities of nodes in a typical VSN. However, properly designed VSN may provide significant storage and computing capabilities as well, but in a highly distributed manner. The algorithms that run on such network have to be aware of this distributed nature to take advantage of these capabilities.

Let us take a look at one of the examples of computer vision – object recognition. The basic approach to this task is as follows:

- Learning phase: a compact representation (model) of the object is extracted from one or more images and stored. In our case, such compact representation consists of a number of *features* and is represented as feature vector.
- Recognition phase: the same compact representation of an object (feature vector in our case) is extracted from the newly acquired image. This vector is compared to the feature vectors stored during the learning phase to obtain a correspondence with one of the learned objects.

A network of visual sensors is expected to encounter a large number of objects. The purpose of learning in such distributed environment is to provide the ability of the network to recognize objects that have already been seen by any of its nodes. Therefore, after a new image is acquired, a node proceeds as follows:

1. Extraction of the relevant feature vector from the visual representation of the object.
2. Distribution of the feature vector to other nodes that comprise VSN.

As soon as a sufficient amount of knowledge (e.g. sufficient number of feature vectors) has been extracted by the network, object recognition can be performed simultaneously with learning. In our case, we do not explicitly deal with the concept of incremental learning; as new images are acquired, network *knowledge* increases, but only due to the increase in the number of stored feature vectors. The already acquired feature vectors remain unchanged.

It is evident that an efficient mechanism to disseminate knowledge (e.g. mechanism for distribution of feature vectors) across the network is needed, as the complexity of this problem increases non-linearly with the number of nodes.

2.1 Hierarchical encoding structure

Hierarchy is the basic principle on which our feature encoding is based. We require that the *primary node* (the visual sensor that has originally seen the unknown object) retains complete information about the object. Its neighbors receive less detailed, more abstract information, which, in general, requires less storage space and less transmission capacity. In this way the amount of data transmitted across or stored in the network can be significantly reduced.

Specifically, for an object recognition task such structure requires that the feature vectors x, which are passed across the network, fulfill the four major requirements.

Requirement 1: There exists a mapping $f : x^n \mapsto x^{n+1}$, which translates level n feature vector x^n into higher, more abstract level $(n+1)$ feature vector x^{n+1}, without access to the original visual data.

This requirement assumes that the primary node extracts level 0 feature vectors, x^0, directly from the acquired image. Its direct neighbors receive level 1 feature vectors, x^1, their neighbors receive level 2 feature vectors, x^2, and so on. Mapping $f : x^n \mapsto x^{n+1}$ is done on each of the nodes before transmitting the feature vectors x^{n+1} to its neighbors, until the maximum level of abstraction is reached. From this point on, feature vectors are forwarded unchanged.

Requirement 2: If $I(x)$ is the storage space required for the feature vector x in bits, then it should hold that: $I(x^n) \geq I(x^{n+1})$.

Requirement 3: There exists a metric $d^n(x_1^n, x_2^n)$ which provides a measure of *similarity* between two feature vectors x_1^n and x_2^n of the same level n.

The existence of the metric is essential both for the object recognition itself and for the hierarchical feature encoding scheme. The distance $d^n(x_1^n, x_2^n)$, when compared to the threshold T, determines if the objects are *similar*, $d^n(x_1^n, x_2^n) \leq T$, or not, $d^n(x_1^n, x_2^n) > T$.

Requirement 4: Given two vectors x_1^n and x_2^n which are similar, $d^n(x_1^n, x_2^n) \leq T$, the corresponding vectors on the next level $n+1$ should be at least as similar as the vectors on the previous level, $d^{n+1}(x_1^{n+1}, x_2^{n+1}) \leq d^n(x_1^n, x_2^n)$.

2.2 Propagation of the feature vectors

For a moment, let us assume that the primary node has acquired an image of a new object and has already discovered that the observed object had not been seen before.

First, the feature vector is extracted and the random identification number (ID) is generated and attached to the feature vector. The actual procedure of feature extraction depends on the recognition method used. The extracted

vector is then stored locally and marked as being level 0 vector, x^0. Then, using a mapping $f : x^n \mapsto x^{n+1}$, the next level feature vector x^1 is prepared, assigned the same ID and broadcasted to all direct neighbors of the primary node. Each receiving node attaches a tag to the received vector, which uniquely determines the origin of the feature vector. Due to Requirement 2, the level 1 feature vectors x^1 require less storage space than vectors x^0.

The process of applying the mapping $f : x^n \mapsto x^{n+1}$ is repeated on each node, until every node has at least *some* information about the object seen by the primary node. Communication between nodes and unique IDs ensures that the nodes refuse to accept any duplicated feature vectors.

2.3 Object recognition

The task of object recognition is performed for any new image that any of the nodes (cameras) acquire and can reasonably believe that it contains object(s) of interest. It is described in Algorithm 1. Again, we call the node that has acquired the image the *primary node*.

Algorithm 1 : Object recognition

Input: Image
Output: Object correspondence

1: Extract object features.
2: // Local search
3: **for** All levels in local storage **do**
4: Apply the mapping $f : x^n \mapsto x^{n+1}$ and calculate next level feature x^{n+1} from x^n.
5: Compare x^{n+1} with all the vectors of level $n + 1$ from the local storage.
6: **if** No match is found **then**
7: Terminate the search, object is unknown. Optionally, proceed with learning.
8: **else if** Match is found on the level 0 **then**
9: Object has been seen locally.
10: **else**
11: // Some other node might have seen the object.
12: // Proceed with network search.
13: **for** All matching vectors **do**
14: Examine tags, attached to the locally stored matching feature vector.
15: Forward level 0 features of the unknown object to the neighbor, who provided locally stored matching feature vector.
16: // Upon receiving forwarded features, neighboring nodes start from Line 2 of the Algorithm 1.
17: **end for**
18: **end if**
19: **end for**

As it can be seen, Algorithm 1 is recursive in its nature. In the recognition phase, the primary node generates the network *query* packet, if the object has not been seen locally, but it has been seen by any of the other nodes. The query packet contains unmodified level 0 features of an unknown object, and is transmitted to those primary node neighbors, which provided matching features. Upon receiving the query packet, every node runs Algorithm 1 from the Line 2 on.

The neighboring nodes process the forwarded features of an unknown object in exactly the same way as if they have acquired the image of an unknown object by themselves. On each node the result of the processing is either:

- The object is unknown (if there is no local match). This result is not reported to the primary node.
- The object is known (match on level 0 is found). This result is reported to the primary node.
- The object might have been seen by the network (match found on one of the higher levels).

The effect of this algorithm is that, during the recognition phase, features of an unknown object in the unmodified form (level 0 features) are forwarded from node to node along the trail, left by the propagated feature vectors in the learning phase. If the primary node does not receive any replies from the other nodes, the object is unknown to the network. The efficiency of this approach stems from the fact that features are forwarded only in the direction, where there is a possibility that the object has been seen.

3 Examples

Our hierarchical feature encoding framework does not rely on any particular object recognition method. It only provides the requirements that enable any recognition method to be implemented in a distributed way. To illustrate practical usage of the encoding framework, we present application of three very common and widely known recognition methods. The first two methods, template matching and histogram matching, are trivial and provided for illustrative purposes only. The third one, principal component analysis, is widely used in many object recognition tasks, such as face recognition [16].

3.1 Template matching

Let us limit our discussion to the simplest form of template matching – direct comparison of two images of the same dimensions. In this case, the feature vector, representing the object, simply contains pixel values of the original image. Following Requirement 1, we can define the mapping $f : x^n \mapsto x^{n+1}$ as

a simple subsampling operation, which reduces image dimensions by calculating 2×2 pixel averages. It is obvious that Requirement 2 is fulfilled, as the resulting image dimensions are halved and both the image and the corresponding feature vector require only a quarter of the original storage space.

The metric, $d^n(x_1^n, x_2^n)$, can simply be the Euclidean distance between the feature vectors. Since the vector components are pixel values, it is easy to show that the Requirement 4 holds. The situation is presented in Figure 1.

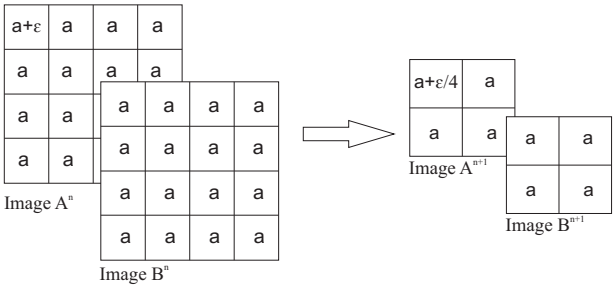

Figure 1. Mapping $f : x^n \mapsto x^{n+1}$ in the case of template matching.

Let us use metric d^n to compare two images that are nearly identical (for the sake of simplicity, let us assume that every pixel has the value of a), except for the one pixel in image A, which has value $a + \epsilon$. The original images A^n and B^n have dimensions $m \times m$ and the resized images A^{n+1} and B^{n+1} have the dimensions $m/2 \times m/2$. The metric d^n is then

$$\begin{aligned} d^n(x_A^n, x_B^n) &= d^n(A^n, B^n) = \\ &= \sqrt{\textstyle\sum_i^m \sum_j^m (A^n(i,j) - B^n(i,j))^2} \end{aligned} \tag{1}$$

and the distances are:

$$\begin{aligned} d^0(A^0, B^0) &= \epsilon \\ d^1(A^1, B^1) &= \tfrac{\epsilon}{4} \\ &\cdots \\ d^n(A^n, B^n) &= \tfrac{\epsilon}{4^n}. \end{aligned} \tag{2}$$

3.2 Histogram matching

Intensity and color histograms are a compact representation of an object. Although in practice [17] high dimensional histograms are used, we will limit ourselves to the usage of one-dimensional intensity histograms.

The intensity histogram for an 8-bit image may contain up to 256 bins. Each bin contains a normalized count of image pixels within a certain range of grey levels. Accordingly, the elements of the feature vectors are simply normalized histogram bin counts. Following Requirement 1, we can define mapping $f : x^n \mapsto x^{n+1}$ as an operation that combines adjoining bins, giving the coarser representation of the original image. Again, it is obvious that Requirement 2 is fulfilled, since the lower number of bins requires less storage space.

The metric, $d^n(x_1^n, x_2^n)$, can simply be the Hellinger distance [17] (which is related to Bhattacharyya coefficient) between the histograms:

$$d^n(x_A^n, x_B^n) = \sqrt{1 - \rho(h_A^n, h_B^n)}, \tag{3}$$

where $d^n(x_A^n, x_B^n)$ is the distance between the feature vectors, x_A^n and x_B^n, $\rho(h_A^n, h_B^n)$ is Bhattacharyya coefficient, $\rho(h_A^n, h_B^n) = \sum_{i=1}^p \sqrt{h_{iA}^n h_{iB}^n}$, p is number of bins, which is the same for both histograms, and h_{iA}^n and h_{iB}^n are the normalized bin values for i-th bin in histograms h_A^n and h_B^n, respectively.

Given the same example as in the previous section, the corresponding histograms of images A and B are shown in the first column of Figure 2. The histogram for image A contains two non-zero bins, a and $a + \epsilon$, whereas the histogram for image B contains only one, a.

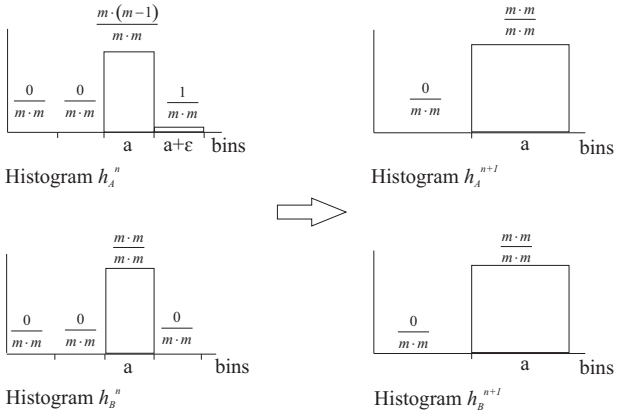

Figure 2. Mapping $f : x^n \mapsto x^{n+1}$ in the case of histogram matching.

The distance $d^n(x_A^n, x_B^n)$ for each level n can be calculated as follows. For level 0, where the 256-bin histogram is used, the distance is:

$$d^0(x_A^0, x_B^0) = \sqrt{1 - \rho(h_A^0, h_B^0)} =$$

$$= \sqrt{1 - (\sqrt{\tfrac{m\cdot(m-1)}{m\cdot m} \cdot \tfrac{m\cdot m}{m\cdot m}} + \sqrt{\tfrac{1}{m\cdot m} \cdot \tfrac{0}{m\cdot m}} + \cdots)}$$

$$d^0(x_A^0, x_B^0) = \sqrt{1 - \sqrt{\tfrac{m^2-1}{m^2}}}, \tag{4}$$

where m is the dimension of the original (square) images, x_A^0, x_B^0 and h_A^0, h_B^0 are level 0 feature vectors and level 0

histograms, respectively. It can be seen that the distance d^0 is greater than 0. After performing mapping f, we have two possible scenarios. If the bins a and $a+\epsilon$ are the adjoining bins, the newly calculated histogram h_A^{n+1} will contain only one non-zero bin with the value 1 and it will be the same as histogram h_B^{n+1}, so the distance $\sqrt{1-\rho(h_A^{n+1},h_B^{n+1})}$ will be 0. In the other scenario, the bins are not adjoining and the mapping f will have no effect on the distance $\sqrt{1-\rho(h_A^{n+1},h_B^{n+1})}$, which remains greater than 0. It can be seen that Requirement 4 is fulfilled as well.

3.3 Principal component analysis

Principal component analysis (PCA, also called the Karhunen-Loéve transform) is a vector space transform for reducing the multidimensional data sets to lower dimensions without significant loss of information. PCA transforms the data to a new coordinate system in which the basis vectors follow modes of greatest variance in data [18]. In essence, properly constructed feature vectors contain feature values, which are already ordered by decreasing importance in terms of reconstruction of the original data. In our case it is assumed that the PCA transformation coefficients are obtained in advance and that they remain fixed during the network operation.

This opens up a possibility of mapping function $f : x^n \mapsto x^{n+1}$, which can be defined as dropping a certain number of features of the lowest importance from the feature vector. Again, it is obvious that Requirement 2 holds.

Considering the metric $d^n(x_1^n, x_2^n)$, Euclidean distance is commonly used when comparing the PCA-based feature vectors. It is easy to show that Requirement 4 holds as well.

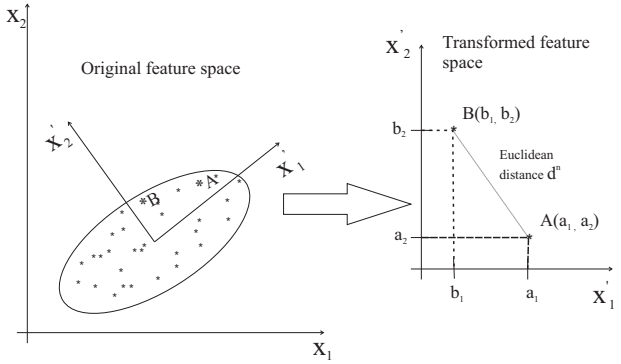

Figure 3. Mapping $f : x^n \mapsto x^{n+1}$ in the case of the PCA-based feature vectors. Images A and B in the original feature space (left) and images A and B in the transformed feature space (right).

Ignoring one of the dimensions from the Euclidean space never increases the distance between the two points, as illustrated in Figure 3. At most, the distance remains the same. The distance d^n is always equal to or larger than any of the distances $|b_1-a_1|$ or $|b_2-a_2|$. This holds also for high-dimensional cases. That means, regardless of the order of features, the distance will always decrease with decreased dimensionality of the feature vectors, and Requirement 4 is fulfilled.

4 Conclusion

The paper focuses on solving an important conceptual problem of mapping computer vision methods to a visual sensor network, specifically on the issue of knowledge propagation. Image processing and computer vision methods operate on large amounts of data, which can easily overload the communication-constrained distributed network. We propose hierarchical feature encoding that guarantees accessibility of any feature vector from any node of the network. The approach works without transmitting complete images or complete feature vectors to every node whenever a new visual information is obtained. We focus on a common task in computer vision – object recognition, however, the outlined designed principles have a wider applicability. The proposition is accompanied with examples of simple object recognition algorithms to show how they fit into our framework. In future, we plan to simulate and measure network behavior by applying the proposed strategy to different algorithms. Adaptation of the state-of-the-art computer vision methods to our framework is planned, too.

5 References

[1] Y. Charfi, N. Wakamiya, and M. Murata. Challenging issues in visual sensor networks. *http://www.nal.ics.es.osaka-u.ac.jp/ charfi/*.

[2] W. Yu, Z. Sahinoglu, and A. Vetro. Energy efficient jpeg 2000 image transmission over point-to-point wireless networks. Technical report, MERL- A Mitsubishi Electric Research Laboratory, 2004.

[3] H. Wu and A.A. Abouzeid. Error resilient image transport in wireless sensor networks. *Computer Networks*, 50(15):2873–2887, 2005.

[4] Y. Charfi, N. Wakamiya, and M. Murata. Trade-off between reliabilty and energy cost for content–rich data transmission in wireless sensor networks. In *Proceedings of the Broadnets*, pages 1–8, 2006.

[5] Z. Cheng, D. Devarajan, and R.J. Radke. Determining vision graph for distributed camera networks using feature digests. *Journal on Advances in Signal Processing*, 2007. doi:10.1155/2007/57034.

[6] D. Devarajan and R. J. Radke. Calibrating distributed camera networks using belief propagation. *EURASIP Journal on Applied Signal Processing*, 2007. doi:10.1155/2007/60696.

[7] W. Luh, D. Kundur, and T. Zourntos. A novel distributed privacy paradigm for visual sensor networks based on sharing dynamical systems. *EURASIP Journal on Advances in Signal Processing*, 2007. doi:10.1155/2007/21646.

[8] C. Arth and H. Bischof. Real–time object recognition using local features on a dsp–based embedded system. *Journal of Real–time Image Processing*, pages 233–253, 2008.

[9] C. Arth, H. Bischof, and C. Leistner. Tricam-an embedded platform for remote traffic surveillance. In *Embedded Computer Vision Workshop (held in conjunction with CVPR)*, 2006. doi:10.1109/CVPRW.2006.208.

[10] C.-H. Moon and D.-Y. Jang. *Intelligent Computing in Signal Processing and Pattern Recognition*, chapter Embedded System Implementation for an Object Detection Using Stereo Image, pages 230–240. Springer Berlin / Heidelberg, 2006.

[11] X. Wang, S. Wang, D.-W. Bi, and J.-J. Ma. Distributed peer-to-peer target tracking in wireless sensor networks. *Sensors*, pages 1001–1027, 2007.

[12] M. A. Patricio, J. Carbo, O. Perez, J. Garcia, and J. M. Molina. Multi–agent framework in visual sensor networks. *EURASIP Journal on Advances in Signal Processing*, 2007. doi:10.1155/2007/98639.

[13] A. Gilbert and R. Bowden. Tracking objects across cameras by incrementally learning inter-camera colour calibration and patterns of activity. In *Proceedings ECCV 2006*, pages 125–136, 2006.

[14] M. Quaritsch, M. Kreuzthaler, B. Rinner, H. Bischof, and B. Strobl. Autonomous multicamera tracking on embedded smart cameras. *EURASIP Journal on Embedded Systems*, 2007. doi:10.1155/2007/92827.

[15] S. Fleck, F. Busch, P. Biber, and W. Strasser. 3d surveillance–a distributed network of smart cameras for real-time tracking and its visualization in 3d. In *Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW06)*, 2006. doi:10.1109/CVPRW.2006.6.

[16] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–591, 1991.

[17] M. Kristan, J. Perš, M. Perše, and S. Kovačič. Closed-world tracking of multiple interacting targets for indoor-sports applications. *Computer Vision and Image Understanding*, 2008. In press.

[18] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. Thomson Learning, third edition, 2008.

Vildana Sulić received her B.Sc. degree from the Faculty of Electrical Engineering of the University of Ljubljana, Slovenia in 2006, in the field of biomedical engineering. Currently she is a Ph.D. student and junior researcher at the Machine Vision Laboratory of the same faculty, with her interests in computer vision, image processing, human motion analysis and visual sensor networks.

Janez Perš received his Ph.D. degree in Electrical Engineering from the Faculty of Electrical Engineering of the University of Ljubljana, Slovenia, in 2004. Currently he works as a researcher at the Machine Vision Laboratory of the same faculty, with his interests in image sequence processing and analysis, object tracking and human motion analysis.

Matej Kristan received his Ph.D. degree in Electrical Engineering from the Faculty of Electrical Engineering of the University of Ljubljana, Slovenia, in 2008. Currently he works as a researcher at the Machine Vision Laboratory of the same faculty and at the Visual Cognitive Systems Laboratory at the Faculty of Computer and Information Science. His research interests focus on statistical pattern recognition, modelling and estimation, tracking, recognition and visual inspection.

Stanislav Kovačič received his Ph.D. degree in Electrical Engineering from the Faculty of Electrical Engineering of the University of Ljubljana, Slovenia, in 1990. Currently he is a vice-dean for Research and professor at the same faculty. His research interests include active vision, image processing and analysis, biomedical and machine vision applications.