# A system for multi-camera task recognition and summarization for structured environments

Dimitrios I. Kosmopoulos, *IEEE Member,*Athanasios S. Voulodimos, *IEEE Member,* Anastasios D. Doulamis, *IEEE Member*

***Abstract*— In this paper, we propose a novel system for visual recognition and summarization of pick and place tasks, that may be executed in settings such as an industrial assembly line. Our novel approach is based on the utilization of hidden Markov models for online task recognition as well as on the use of prior knowledge via a Hopfield-based optimization scheme. To facilitate offline analysis we extract summaries of the captured content based on these features. We extract the motion energy using the norms of the Zernike moments, looking for local minima and maxima that indicate distinctive visual events and thus key-frames. The proposed scheme is not threshold-dependent and therefore the number of extracted key-frames varies according to the complexity of motion energy variation. We validate our system by experimenting on two datasets.**

## I. Introduction

Large-scale enterprises like industrial plants have a clear need for supervision services to guarantee quality and safety [1], which need to be implemented by using multi-camera architectures (see e.g., [2]). However, monitoring is often performed manually and thus inefficiently and subjectively. Focusing on monitoring the production in an industrial plant (such as an automobile manufacturer), which is a fairly structured process, makes modeling of the monitored activities more realistic than in unstructured settings, e.g., an airport. The former processes are often hierarchically structured as workflows, that comprise sequential tasks. As opposed to isolated action monitoring, the goal here is to monitor activities that occur continuously.

For forensic investigations, the challenge of browsing large collections of captured video is even more tedious and error prone for a human [3]. Video analysis technologies can be applied to develop smart surveillance systems to aid the human operator. Apart from the straightforward retrieval of events or abnormalities that were recognized by online processing, techniques for efficient browsing, like video summarization, are of importance, to alleviate for the inevitable errors of automated systems in very complex scenes.

Manuscript received March 28, 2011; revised January 26, 2012; accepted July 9, 2012.



D. I. Kosmopoulos is with the Heracleia Laboratory, Computer Science and Engineering Department, University of Texas at Arlignton TX, E-mail: dkosmo@ieee.org

Athanasios S. Voulodimos is with the Department of Electrical and Computer Engineering, National Technical University of Athens, Greece. E-mail: thanosv@mail.ntua.gr

Anastasios D. Doulamis is with the Department of Production and Management, Technical University of Crete, Greece. E-mail: adoulam@dpem.tuc.gr.

Taking these observations into consideration, our work contributes by presenting a novel online system for task recognition in visual workflows, so that real time production monitoring can be achieved. The proposed framework is based on hidden Markov models (HMMs) and holistic features. The prior information about the task sequence is exploited by minimizing an energy function via a Hopfield neural network (HNN). We show how the streams from multiple cameras can be exploited for task segmentation and classification.

Our second contribution is a method for real time summarization of the acquired video, which enables better browsing by reusing the scene representation vectors. Offline manual supervision of industrial processes is long and arduous but also subjective in the sense that human interpretation highly depends on fatigue, attention, cognitive ability etc. On the other hand, storing and processing such huge amount of visual information is challenging and financially inviable. Large scale industries may operate 24/7 producing vast amount of content, which has to be accessed efficiently. Apart from the aforementioned human factors, a log file recording all the abnormal events detected by the system, i.e., only some rare events, would probably not be sufficient because the parameters that affect event detection may change frequently. Moreover, a log-file approach assumes zero or few false negatives; but eliminating false negatives by parameter tuning, can result in more false positives, thus increasing the inspection time.

In the rest of the paper after surveying the related literature in section II, we describe the workflows as well as the system setup in section III. We continue with the feature extraction and task recognition in a multicamera setting in section IV, and the workflow analysis in section V; the summarization is described in section VI. In the experimental results, in section VII, we compare our methods with other popular ones, by using a challenging industrial workflow dataset [4] and by verifying the results in the TUM kitchen dataset [5]. We conclude in section VIII.

## II. Related work

### A. Behavior analysis

Human action recognition has been the focus of interest of computer vision and machine learning, mostly as isolated activities and not as workflows, see, e.g., [6], [7], [8], [9], [10].

A very flexible framework for classification of time series is the HMM (e.g., [11]), which has been used for modeling and extraction of human behavior (e.g., [12]). It can be easily extended to fuse multiple streams (e.g., [13]). It is very efficient for application in previously segmented sequences

(e.g., [1]), however when the boundaries of the sequence that we aim to classify are not known in advance, the search space of all possible beginning and end points make the search inefficient [14]. [15], presents a dynamic programming algorithm of cost proportional to the cube of the duration for segmentation and classification, which is restrictive in real world applications.

To exploit the hierarchical structure of some time series, the hierarchical HMMs (HHMMs) [16] were used. Each state is considered to be a self-contained probabilistic model (an HHMM). Examples of such approaches can be found in [17], where the workflow in a hospital operating room is described. Another approach is the layered HMM (LHMM) (see [18]), which consists of $N$ levels of HMMs where the HMMs on level $N+1$ correspond to observation symbols or probability generators at level $N$. In [19] structure learning in HMMs is addressed to obtain temporal dependencies between high-level events for video segmentation.

In many workflows, such as in industrial production, where a sequence of tasks has to be completed, the execution of a task means that it will not appear again in the same workflow. Therefore the whole history of tasks must be kept in memory to exclude false positives. The Markovian property, which states that the current state depends only on the previous, is obviously not applicable. Thus, the above approaches have an inherent problem to describe such workflows.

The Echo State Network (ESN), (see, e.g., [20]), could be a promising method for online classification of workflow time series, because it does not make any explicit Markovian assumption. However, it was shown in [21] that it effectively behaves as a Markovian classifier, i.e., recent states have a far larger influence on the predicted state.

Of relevance to the preprocessing step of task segmentation that we apply, are the methods that seek to segment and classify video sequences. Unlike other types of videos (e.g., sports, news, movies), where there are discrete shot boundaries or color variations that can be used as visual cues [22], generally in surveillance videos there are not so discrimintative changes. However, there is a source of information that can be used for content classification and it is the object motion, e.g., [23]. Methods for spatiotemporal object segmentation, (e.g., [24] that uses object trajectories) without assumptions about the structure of the videos seem attractive in this context. However, although such methods can be effective for offline processing, they are rather impractical in an online acquisition setting, due to high processing demands and due to the need for availability of the full videos.

In this work we take advantage of the object motion structure to identify the boundaries of the tasks online (semantic segmentation), by employing a learning approach. We also bypass the erroneous Markovian assumption by employing an optimization scheme that penalizes the re-appearance of tasks.

### *B. Video summarization*

The first approaches for automatic video summarization had the goal of extracting key-frames at regular time instances or within a shot [25]. Such selection, however, is far from being representative especially when someone should quickly overview complex industrial processes as in our case. Research was concentrated on specific types of video content like sports [26], instructional videos [27]. Algorithms were proposed maximizing entropy and exploiting information theoretic measures [28], [29], cross correlation [30] and/or perceptual users' centric video summaries [31].

Most of the current algorithms assume a predefined number of key-frames. In industrial settings, however, the exact number of key-frames is not known in advance. Another significant peculiarity is that industrial workflows are often decomposed by periodic movements and tasks which should be handled as different key-frames, since they refer to different time stamps. It is important to detect not only the commencement of a task, but also to capture the time for such execution, especially in relation with other tasks. Most of the current summarization methods process videos offline, but in our case, due to continuous video production, we should apply computationally efficient algorithms that can yield the summarization results at a rate at least as fast as the content production rate.

We propose a novel method that exploits the fluctuation of the feature vector energy to detect efficiently periodic motion, which is very common in industrial settings.

## III. Problem Definition and system setup

The workflow on this assembly line included tasks of picking several parts from racks and placing them on a the welding cell (WC) some meters away. The behaviors are [4]:

1) Pick part #1 from rack #1 and places on WC.
2) Pick part #2a from rack #2 and place it on WC.
3) Pick part #2b from rack #3 and place it WC.
4) Pick parts #3a and #3b from rack #4 and place them on the WC.
5) Pick part #4 from rack #1 and place on WC.
6) Pick part #5 from rack #5 and place it on WC.

Usually a single worker performs tasks 1,3,5 and a pair of them performs the rest ones. Each of these tasks is a class.

An overview of the proposed system architecture is given in Fig.1. The configuration of the cameras and the assembly workspace are depicted in Fig.2. Two cameras with partially overlapping views were used to overcome occlusions.

## IV. Feature extraction and task recognition in a distributed camera setting

The classification of visual tasks requires the definition and extraction of features. We define one such vector per frame and series of those vectors are the input to time series classifiers.

The employment of features directly extracted from the video frames has the significant advantage of obviating the need of detecting and tracking the salient scene objects, a process which is notoriously difficult in cases of occlusions, target deformations and illumination changes such as in the workflow recognition dataset that we deal with [1].

In [32], it was shown that pixel change history (PCH) images can capture the motion duration information with high

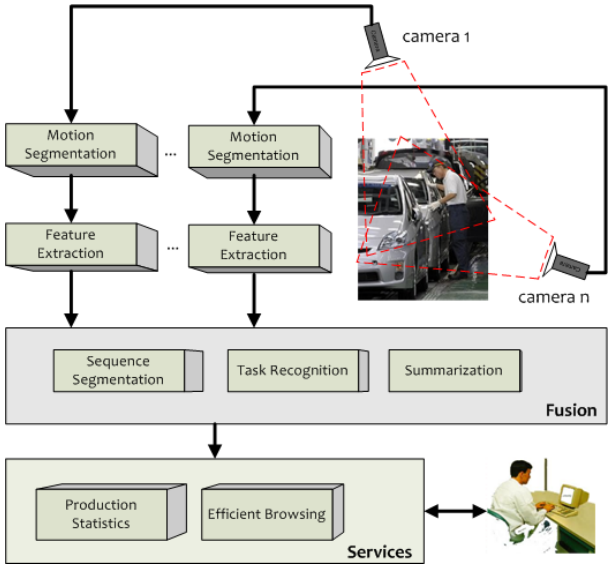



Fig. 1. Architecture of the proposed framework. For each camera stream the motion segmentation is followed by extraction of feature vectors. The feature vectors from multiple cameras are fused to enable task segmentation, task recognition and summarization. Then services such as production statistics and efficient browsing through summaries can become available.

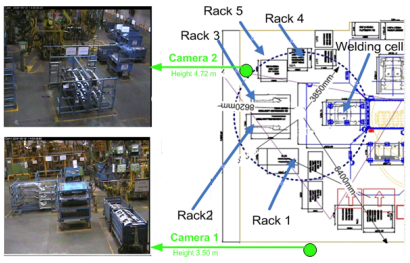



Fig. 2. Depiction of a workcell along with the position of the cameras and the racks #1-5. The recognized behaviors are associated with transferring each part from the respective pallet and putting it on the welding cell.

performance. The PCH of a pixel is defined as:

$$P_{\varsigma,\tau}(x,y,t) = \begin{cases} \min(P_{\varsigma,\tau}(x,y,t-1) + \frac{255}{\varsigma}, 255) \\ \text{if} D(x,y,t) = 1 \\ \max(P_{\varsigma,\tau}(x,y,t-1) - \frac{255}{\tau}, 0) \\ otherwise \end{cases} \quad (1)$$

where $P_{\varsigma,\tau}(x,y,t)$ is the PCH for a pixel at $(x,y)$, $D(x,y,t)$ is the binary image indicating the foreground region (can be extracted in real time by standard foreground segmentation methods e.g., [33]), $\varsigma$ is an accumulation factor and $\tau$ is a decay factor. By setting appropriate values to $\varsigma$ and $\tau$ we are able to capture pixel-level changes over time (see Fig. 3).

We need to mention that the result of the background subtraction process doesn't have to be highly accurate. The only requirements are that (a) different foreground objects give different foreground segments, so that there is high dissimilarity between different patterns and that (b) the results are repeatable, so that there is high similarity between similar patterns. Most baseline methods satisfy (a) and (b).

To represent the PCH images as vectors we use the Zernike moments, which are among the most popular choices as shape

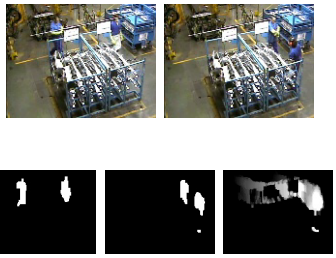


Fig. 3. Two key-frames (first row), the respective background subtraction images and the extracted PCH image (second row)

descriptors (see e.g., [34]), due to noise resiliency, reduced information redundancy, and reconstruction capability.

The order $p$ complex Zernike moments are defined as:

$$Z_{pq} = \frac{p+1}{\pi} \int_0^1 \int_{-\pi}^{\pi} R_{pq}(r) e^{-jq\theta} f(r,\theta) r dr d\theta \quad (2)$$

where $r = \sqrt{x^2+y^2}$, and $\theta = \tan^{-1}(\text{y/x})$ and $-1 < x,y < 1$ ($(x,y)$ are the normalized coordinates in image $f$, with respect to the center of the image and the integration is calculated within a circle of normalized radius equal to 1) and:

$$R_{pq}(r) = \sum_{s=0}^{\frac{p-q}{2}} (-1)^s \frac{(p-s)!}{s!(\frac{p+q}{2}-s)!(\frac{p-q}{2}-s)!} r^{p-2s} \quad (3)$$

where $p-q = even$ and $0 \le q \le p$. Given values for the pair $(p,q)$, subject to the previous constraint, the outcome of (2) is a complex number. To represent a circular image region we use the magnitude and the phase of the complex number returned by (2) for several pairs of $(p,q)$.

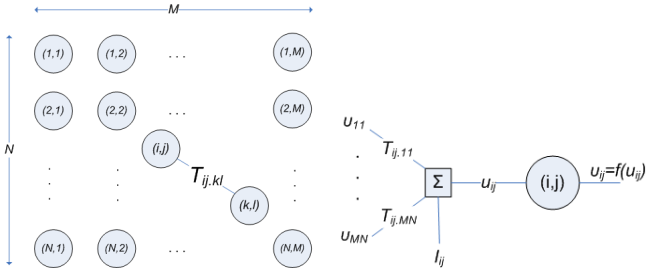
In the following step we extract the most probable task given segmented sequences of vectors coming from several cameras, i.e., determining the start and finish time for each task. To handle the occlusions we fuse the input features from multiple cameras. Our goal is to achieve behavior recognition results better than the results that we could attain by using the information obtained by the individual camera streams independently of each other.

We adopt the multistream fused HMM (MFHMM), which was proposed recently for multistream data modeling [13]. The connections between the component *streamwise* HMMs of this model are chosen based on a probabilistic fusion model, which is optimal according to the maximum entropy principle and a maximum mutual information criterion for selecting dimension-reduction transforms. Here the observations emitted by a camera-specific stream are coupled with the states of the other streams. The adopted fusion scheme is a state-fusion approach in contrast to feature level fusion and decision-level fusion (see e.g., [35] for details on different fusion approaches). In [1] the superiority of this method compared to other popular fusion methods was verified. Here we train one multistream model per task that we seek to recognize. Given the sequence of observations coming from multiple cameras the most probable task is the one for which the associated multistream model gives the highest probability.

The proposed fusion method works with segmented tasks. We thus had to pre-process the observation sequence to identify tasks transitions from the visual data as they are captured online by the cameras. The key observations that enable the task segmentation in our context are: (a) the tasks are sequential but their order may vary, (b) each task is executed only once, (c) the tasks have a variable duration, however, the durations of the same tasks are statistically similar, (d) each task ends with placing a part on the welding cell (e) when there is no task execution, normally, there is no activity.

Based on the above observations we create models to recognize the part placement on the welding cell, since it signifies the tasks boundaries. This is similar to learning models for recognizing the transition between shots in other types of videos where shots exist (e.g., [22]). In particular, let us denote as $x_t$ a binary value indicating whether a task's transition has occurred at frame $t$ or not and as $\mathbf{f_t}$ a feature vector containing all observations at time $t$. $\mathbf{f_t}$ includes data derived from a camera or all cameras observing the scene. Then, our goal is to estimate the binary transition value $x_{t+1}$ for the following future frame at time $t+1$, taking into account the current knowledge $x_t$ and observations $\mathbf{f_t}$. However, to ascertain a reliable estimation of the value $x_{t+1}$, it is more effective to take into account, apart from the current value $x_t$, previous observations taking place at older time instances. Assuming that $r$ previous observations are sufficient to get a reliable prediction for the task's transition value $x_{t+1}$ we can model $x_{t+1}$ through the previous observations $\mathbf{f_t}$ $t = 1, 2, .., r$. Indeed the worker's trajectory towards the welding cell is discriminative and can be modelled. For more complex applications, non-linear regression models have been introduced to relate previous samples under complex non-linear functions and to derive predictions for future transitions.

Here we use neural networks to model the non-linear regressions. To provide a probabilistic output for the transition state, probabilistic networks (PNNs) are employed. A PNN (e.g., [36]) is an implementation of a statistical algorithm called kernel discriminant analysis in which the operations are organized into a multilayered feedforward network.




(a) A configuration of $M \times N$ nodes. $T_{ij,kl}$ is the weight from node $(k,l)$ to node $(i,j)$.

(b) A single node $(i,j)$ with input $u_{i,j}$ and output $v_{ij}$. The input is calculated by (5) and the output by the sigmoid transfer function).

Fig. 4. A 2-D Hopfield neural network.

## V. Workflow recognition

As may happen in many worklow cases, in our application scenario the execution of a task prohibits the re-appearance of the same task until the workflow is over. Additionally, the order of appearance of each task is not random, but is imposed by the assembly process, which requires that some tasks have to be finished before some other ones can start. In such structured applications it is reasonable to take advantage of such information to exclude tasks that are not feasible given their context. In this section we present a method on how to identify the task sequence by using prior information without relying on the Markovian assumption.

Assuming that the workflow is composed of $K$ segmented tasks, our goal is to evaluate the different task permutations and select the most appropriate according to an objective function. However, the number of task permutations is given by $K!$ assuming no task repetitions. For $K = 6$ we may have 720 cases to evaluate, for $K$ = 20 the cases are more than $2.43 \cdot 10^{18}$, so an exhaustive approach is not scalable.

We can formulate the problem of assigning the optimal task label to each of the $K$ segments in such a way that we can solve it efficiently by employing an HNN [37]. The HNN has been employed in the past to solve iteratively optimization problems such as the traveling salesman problem [38]. In our case we have $K$ tasks and we can define a 2-D network of $K \times K$ nodes. If the output $v_{ij}$ of the node $(i,j)$ equals one, then the $i$-th task is executed in the $j$-th order.

The energy function of a 2-D HNN with $N \times M$ nodes (see Fig. 4) is defined as:

$$E_h = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{N}\sum_{l=1}^{M} T_{ij,kl} v_{ij} v_{kl} - \sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij} v_{ij} \quad (4)$$

where $v_{ij}$ is the binary state of the neuron in row $i$ and column $j$, $T_{ij,kl}$ is the interconnection weight between the neuron in row $i$ and column $j$ and the neuron in row $k$ and column $l$. A neuron $(i,j)$ in the network receives weighted inputs $T_{ij,kl}$ from the neuron $(k,l)$ and a bias input $I_{i,j}$ from outside. The total input to neuron $(i,j)$ is given by:

$$u_{ij} = \sum_{k=1}^{N}\sum_{l=1}^{M} T_{ij,kl} v_{kl} + I_{ij} \quad (5)$$

The output of each neuron is computed by using a continuous sigmoid transfer function like $v_{ij} = \tanh(a u_{ij})$, where $a$ is a positive factor.

As the network runs, starting from a random initial output, the operating rule drives the network towards to the direction of minimizing the energy function. We can formulate our problem in such a way that the resulting energy function is minimized by the optimum task sequence. Therefore, we can define an energy function which is composed of the terms that are given in the following. Initially

$$f_1(x) = \sum_{i=1}^{K}(\sum_{j=1}^{K} v_{ij} - 1)^2 \quad (6)$$

expresses the constraint that each order is assigned to a single task. We note that here $N = M = K$. Similarly,

$$f_2(x) = \sum_{j=1}^{K} (\sum_{i=1}^{K} v_{ij} - 1)^2 \quad (7)$$

expresses the constraint that each task is assigned to a single order. For each of the segments a probability is computed with respect to all possible tasks. We seek to maximize the probability of the whole sequence of tasks, so the term that we need to minimize is expressed as:

$$f_3(x) = -\sum_{i=1}^{K}\sum_{j=1}^{K} pc_{ij} v_{ij} \quad (8)$$

where as $pc_{ij}$ we denote the probability that the $i$-th task is executed in the $j$-th order, which is calculated for the current input segment by employing the respective HMM. Finally, the current task sequence has to conform to the prior information (obtained after training) so the associated term is given by:

$$f_4(x) = -\sum_{i=1}^{K}\sum_{j=1}^{K} pp_{ij} v_{ij} \quad (9)$$

where as $pp_{ij}$ we denote the probability that the $i$-th task is executed in the $j$-th order, according to prior information.

The energy function combines the terms as follows:

$$E_h = Af_1 + Bf_2 + Cf_3 + Df_4 \quad (10)$$

where the factors $A, B, C, D$ are related to the weight of each term and are experimentally defined. If we replace (6), (7), (8) and (9) in (10) and compare with (4) we find that the quadratic factor is given by:

$$\begin{array}{c} T_{ij,kl} = -2A\left[\delta(i-k, j-l) + 2(1-\delta(j-l))\right] \\ -2B\left[\delta(i-k, j-l) + 2(1-\delta(i-k))\right] \end{array} \quad (11)$$

where $\delta(x)$, $\delta(x, y)$ denote the 1-D and 2-D Kronecker's functions respectively. The $T_{ij,kl}$ are the values of the weights connecting the output of node $(k, l)$ to the input of node $(i, j)$. Similarly, the linear factor in (10) is the bias to the $(i, j)$ node and is given by:

$$I_{ij} = 2(A + B) + Cpc_{ij} + Dpp_{ij} \quad (12)$$

## VI. Summarization

In this section, we describe our approach to dramatically reduce visual information, without, however, losing important information as far as the meaning of an industrial task and/or workflow is concerned [39]. We claim that the semantic content is expressed in the sense of feature vector complexity as described in section IV. This means that other types of features may yield different summaries.

The sum of the squared coefficients of the Zernike moments can express the motion energy or in other words a measure of motion in the current scene. It is defined as:

$$E = \sum_{p=0}^{Q} \sum_{\substack{q \le p \\ \frac{p-q}{2}:even}} \|Z_{pq}\|^2 \quad (13)$$

where $Q$ is the selected order of the moments and $\|\|$ the $L_2$ norm. In other words the energy of a frame is defined as the sum of the squared $L_2$ norms of the associated Zernike moments up to the order $Q$. The total energy in a distributed camera setting can be defined as the weighted sum of the energies of the individual streams, while the energy for each individual stream is given by (13).

The energy can be plotted for each video frame forming a trajectory, which expresses the temporal variation of the energy shape through time. Thus, selection of the most representative frames within a shot is equivalent to selection of appropriate curve points, able to represent the corresponding trajectory. In our case, the second derivative of the shape energy for all frames within a shot with respect to time is used as a curvature measure. Local maxima correspond to time instances of peak variation of the object shape. In addition, local minima indicate low variation of the object shape.

Let us also denote as $E(n)$ the energy of shape coefficients to the $n$-th frame of the examined shot. Initially, the first derivative of signal $E(n)$, say $E'(n)$, is evaluated with respect to time index $n$. A weighted average of the first derivative, say $E'_w$, over a time window, is used to make smoother the fluctuation for the magnitude of the frame feature vectors.

Since frames are discrete time instances, we can model the derivative via difference equations and thus we can estimate the weighted second derivative $E''_w$ , for the $n$-th frame as:

$$E''_w(n) = \sum_{l=n-N_w}^{l=n+N_w} w_{l-n} E''(l) \quad (14)$$

where $2N_w+1$ is the window length and $E''(n) = E'(n+1) - E'(n)$ and $w_{l-n}$ the weight of the $E''(l)$. The local maxima and minima of $E''$ are considered as appropriate curve points, i.e., as time instances for the selected key-frames.

Note that this algorithm is extremely fast since calculations are independent from frame to frame and the time required for applying the frame difference is minimal. Also the Zernike feature vectors are computed anyway for task recognition purposes.

## VII. Experimental Evaluation

### A. Setup

For our experiments, we have used a dataset containing 20 sequences representing full assembly cycles (the dataset will be referred to as WR - Workflow Recognition - dataset) [4]. Each cycle included all six behaviors (one occurrence of each). The total number of frames was approximately 40,000 per camera. In the dataset, the assembly process was rather well structured and was performed by maximum two persons. Other persons or vehicles were sometimes present. The annotation of the datasets has been done manually.

To produce the PCH images we used the blobs calculated from background subtraction. Therefore, we assumed that the motion signatures for each task could be well represented by sequences of holistic features (one feature vector per frame): we used the area, the center of gravity, and the moments (norm and phase) up to 6th order. The moments were calculated in down-scaled rectangular regions of interest (approximately

15,000 pixels) to allow for real-time performance (50-60 fps). From that set, we removed the constant values of four phases; this way, a 31-dimensional vector representation was eventually obtained for each frame. The first 10 sequences were used for training and the rest ten for testing.

### B. Task recognition

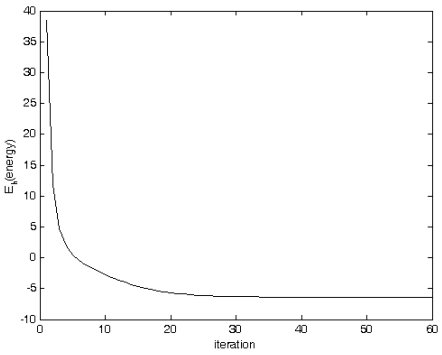
Before applying the time series classifier we had to pre-process the test sequences for recognizing the task transitions. We have employed the PNN (see section IV). We sought to identify two classes (transition and no transition). The PNN included 20 nodes in the hidden layer, using a Gaussian distance function and two nodes in the pattern layer. We used $k$-means to find the centers and generalized inverse for the weights from the hidden to pattern layer. We trained with 60 positive sequences and 300 negative, not overlapping with the positive ones. The number $r$ of previous observations taken into account was set to 25. We experimented with both streams and with stream fusion by concatenating the two streams' feature vectors. The network's output was median filtered with size 5 to avoid high jitter. The segmenation accuracy for the two streams and the fused estimate was measured as $14.1 \pm 10.3$, $10.4 \pm 9.1$ and $10.8 \pm 8.9$ respectively. These correspond to the mean error and standard deviation of the segmentation point estimates from the ground truth (expressed in frames). The measures indicate that the proposed method gave transitions that were very close to the ground truth.

To recognize the tasks a distinct HMM was trained for each separate class/visual task. We used three-state HMMs with one Gaussian per state to model each of the six tasks. Again, we experimented on two individual streamwise HMMs, as well as the MFHMM model, for which the stream weights $r_c$ were selected according to the reliability of the individual streams.

For each stream-wise HMM, as well as for the MFHMM we have done optimization via a $6 \times 6$ HNN as explained in section V. To this end we combined the task probabilities $pc_{ij}$ and the task order priors $pp_{ij}$ with the related uniqueness constraints. The HNN that was employed was a 2-D network of $6 \times 6$ nodes. The parameters ($a$=10, $A$=0.01, $B$=0.01, $C$=1 and $D$=1) were experimentally defined so that the input of the nodes would be in the linear and not the saturated region of the sigmoid transfer function. The maximum number of iterations was 100 and the initial output values were assigned random positive or negative values close to zero. The convergence of the network is displayed in Fig. 5.

Furthermore, to verify the competitiveness of our method we firstly compared it to the ESN, using two instances corresponding to the two camera streams. Each of them had a linear regression reservoir of 500 plain nodes, which was efficient for real time execution, small enough to avoid overfitting but also effective. Increasing the number of nodes would result in high memory requirements without actual benefit. We had six output nodes, corresponding to six tasks. The median of the last 51 estimations was taken for lower output jitter. We have used the Matlab toolbox provided by the authors [20] using spectral radius 0.60, input scaling 0.3 and smoothing of noise level 0.0003 after some experimentation for optimal results. We trained the ESNs with the entire workflows.

(a) Values of the nodes for iterations 1, 5, 10, 20, 40, 60




(b) The energy convergence (see (4) )

Fig. 5. The convergence of the $6 \times 6$ Hopfield neural network for a scenario, for which the correct order of tasks is given by the sequence (1,2,3,5,4,6).

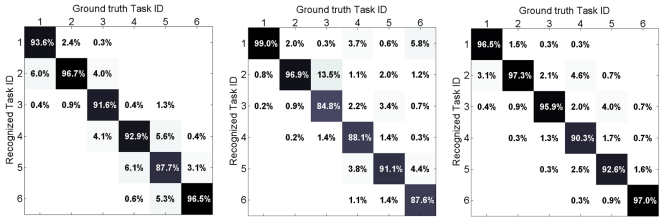
Then we used a method of different rationale, the Dynamic Time Warping (DTW) [40], and we selected as representative for each task the training sample that minimized the sum of the warping distances between itself and all other training samples. During the test, the task representative that minimized the warping distance from the test sequence would determine the label of that sequence. We experimented with both cameras.

Finally to show the effect of the invalidity of the Markovian assumption we also used an HMM in a hierarchical fashion (instead of the HNN) on top of the MFHMM. The HMM states were the detected segments, the emitted observations were the probabilities for each task as provided by the MFHMM, the transition matrix was given by the transition probabilities of the tasks, and the prior was given by the tasks' priors. Then we run the Viterbi algorithm to find the sequence of states-tasks for the HMM. This scheme will be hereafter referred to as MFHMM+HMM, as opposed to our proposed MFHMM+HNN scheme.

Table I shows the comparative precision and recall rates. The HMM+HNNs outperformed the ESN for single streams; this can be largely attributed to the effective Markovian behavior of the ESN, while the sequences are clearly non-Markovian. The HMM+HNNs also performed better compared to DTW on the same segmented sequences, which can be explained by our exploitation of prior knowledge. As for the MFHMM+HMM method we saw that in many cases the Viterbi algorithm did not return the correct sequence of tasks, although it used the same input with the proposed HNN, obviously due to the erroneous Markovian assumption.

We also noticed that the fusion by the MFHMM provided considerable added value leading to precision and recall of 95.2% and 95.0% respectively. The confusion matrices in

Fig. 6 display the impact of the complementarity of the views on the results as well as the successful exploitation of this fact in the case of MFHMM. For example, camera 2 offers a more favorable viewpoint for discerning task 1 from task 5, whereas camera 1 provides a better angle for recognizing task 6.




(a) HMM 1 (b) HMM 2 (c) MFHMM

Fig. 6. Confusion matrices for task classification (WR dataset) using our proposed HMM+HNN approach.

### C. Summarization

We evaluated the proposed summarization algorithm in terms of effectiveness, i.e., how relevant the extracted key-frames were with respect to the overall meaning of an industrial task, but also in terms of computational efficiency. For the latter issue, we stress that the adopted algorithm was implemented to run in real-time and during the frame acquisition. Thus, the proposed approach imposed minimal computational burden allowing online processing.

The goal was to detect a small number of key-frames that were able to represent the whole content visual complexity, without the necessity to browse the whole sequence. The experiments were conducted for all the data. In Fig. 7 we present a typical example that concerns task 2, in scenario 1. In Fig. 7d high fluctuation of E is noticed among frames 250 and 300, which is explained by the high level of activity. The zero-crossings of the 2nd derivative (after filtering) gave the key-frames. The number of key-frames was not predefined but was automatically estimated.

We have also objectively evaluated the proposed summarization algorithm over the whole dataset. For each task a small number of key-frames was extracted, and displayed to two industrial engineers. Then, they identified which was the task that has been executed by observing only this small number of key-frames. For all cases, the tasks have been correctly recognized by the expert users. This reveals that the proposed summarization scheme represented sufficiently the complexity and periodicity of the industrial visual content.

In Table III, we compared the performance of the proposed video summarization method with the methods in [29], [30] and [28]. To perform the comparisons, we initially defined the ground truth of all sequences using six students from two different universities, one in Europe and one in USA. The annotation was performed by forcing the users to set a small range of frames (time intervals), within which the key-frames should belong. No student was aware of the experiments, thus, they were not biased. Only the annotations which were marked by the majority of students were kept as consistent, while the rest were ignored. In case of sequences with no majority, we repeated the annotation.

To objectively compare the results, we used the precision/recall criteria. All key-frames that fell within the time interval set by the users were considered as relevant, while the remaining ones were irrelevant. The comparison was carried out with respect to the number of key-frames needed to be extracted. In our method, the number of key-frames was not fixed but varied according to content complexity. This is an advantage of the proposed technique compared to the other approaches. As observed, the other techniques demonstrated lower performance compared to the proposed method. This was also due to the fact that the proposed technique was suitable for detecting periodic motion.

### D. The TUM kitchen dataset case

The proposed methodology for online task recognition has been developed having in mind the automobile assembly line application. However, it can be applied in several cases where the observations mentioned in section IV are present.

Such an application, which resembles an industrial workflow, is presented by the TUM kitchen dataset [5]. It contains instances of a table-setting workflow performed by different subjects, involving the manipulation of objects. We have used the views from two cameras (cam0 and cam1) to recognize the following sequential tasks (permutations are allowed):

1) Take tray and put it on the table.
2) Take a napkin and put it on the table.
3) Open a drawer, take a fork and put it on the table.
4) Open a drawer (the same as in 3), take a knife and put it on the table.
5) Open a drawer (the same as in 3, 4), take a spoon and put it on the table.
6) Open a shelf, take out a plate and put it on the table.
7) Open a shelf, take out a cup and put it on the table.

We have used workflows/episodes with IDs: 0_0, 0_1, 0_3, 0_6, 0_8, 0_10, 1_0, 1_2, 1_4, 1_7 for training and workflows/episodes with IDs: 0_4, 0_7, 0_9, 0_11, 1_1, 1_3, 1_6 for testing. The ground truth was based on the annotation provided in the dataset. As soon an object was arranged on the table and the subject started heading away from it we marked that point as the end of the current segment and the beginning of a new one. Similarly to the previous experiment, to detect the task transitions we used the same method. The segmenation accuracy for stream 1, stream 2 and the fused estimate was measured as $16.3 \pm 9.2$, $13.2 \pm 10.2$ and $13.5 \pm 8.8$ frames respectively (error mean$\pm$ standard deviation). As can be seen, the described task segmentation method yields good results in this dataset as well, facilitated by the fact that all tasks end in a similar way (placing an object on the table).

For the classification of the tasks we have used HMMs with four states and three mixture components per state. The HNN used the same parameters as in the previous application. The recognition results in terms of precision and recall for the methods examined are shown in Table II. The superiority of our proposed HMM+HNN/MFHMM+HNN approaches over

TABLE I

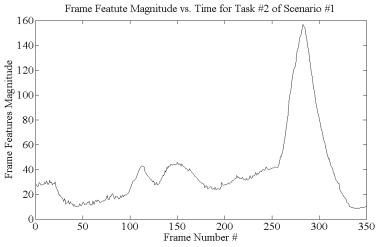
TASK RECOGNITION RESULTS (WR DATASET)

| | HMM1 +HNN | HMM2 +HNN | MFHMM +HNN | ESN 1 | ESN 2 | DTW 1 | DTW 2 | MFHMM +HMM |
|---|---|---|---|---|---|---|---|---|
| Precision | 93.8% | 91.6% | 95.2% | 82.9% | 86.3% | 73.1% | 84.2% | 87.4% |
| Recall | 93.2% | 90.9% | 95.0% | 82.0% | 85.8% | 72.2% | 83.8% | 83.1% |

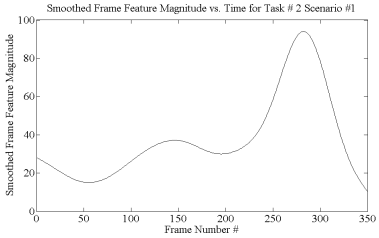
(a) frames 1-20-40-60-80-100

(b) frames 120-140-160-180-200-220

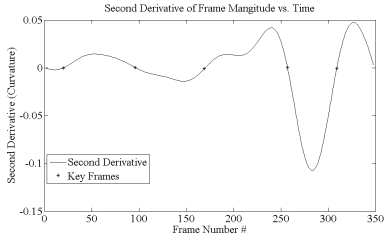
(c) frames 240-260-280-300-320-340




(d) First derivative of $E$




(e) Filtered first Derivative of $E$




(f) Zero crossings of 2nd Derivative

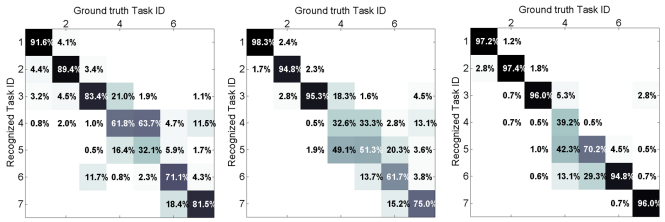
(g) Selected key-frames: 20, 96, 169, 257, 309

Fig. 7. Extraction of key-frames for scenario 1, task 2, camera 1

TABLE II

TASK RECOGNITION RESULTS (TUM KITCHEN DATASET)

| | HMM1 +HNN | HMM2 +HNN | MFHMM +HNN | ESN 1 | ESN 2 | DTW 1 | DTW 2 | MFHMM +HMM |
|---|---|---|---|---|---|---|---|---|
| Precision | 73.1% | 72.7% | 86.1% | 67.1% | 69.5% | 55.1% | 67.1% | 83.5% |
| Recall | 73.3% | 72.4% | 83.9% | 66.5% | 69.7% | 49.4% | 47.6% | 57.8% |




(a) HMM 0 (b) HMM 1 (c) MFHMM

Fig. 8. Confusion matrices for task classification (TUM kitchen dataset) using our proposed HMM+HNN approach.

the other methods (ESN, DTW, MFHMM+HMM) is apparent by the corresponding recognition rates achieved, while similar observations as in the WR dataset case hold. The MFHMM contributes to improvement in precision by a significant 13% in comparison to the single stream case, thus yielding a 86.1% precision rate. The recognition results can be observed in more detail through the confusion matrices in Fig. 8.

It is noted that tasks 4, 5 and 6 bear a great resemblance since they consist in opening the same drawer, picking a similar object (fork, knife, or spoon) and placing it on the table, thus it is quite difficult for a classifier to differentiate among them; that is confirmed by the misclassification rates among tasks 4, 5, 6 as displayed in Fig. 8. If we chose to consider these tasks as a common task (which would then have a triple occurrence in every workflow), then the rates attained by our approach using camera 0, camera 1 and fusion would rise to 86.5%, 89.6% and 96.7% respectively in terms of precision, and 86.1%, 89.3% and 95.9% in terms of recall.

For the summarization task, similar methodology as in sub-section VII-C was applied. The comparisons with other methods are given in Table IV and verify the superiority of the proposed method.

## VIII. Conclusion

We have demonstrated the theoretical and practical issues associated with the implementation of a system for visual monitoring and summarization in the assembly line of a major automobile manufacturer and we further validated our results by using the TUM kitchen dataset. The system recognized tasks in workflows with high accuracy despite the challenging setting. The prior information was effectively exploited by a fast converging HNN. Comparison to the ESN and DTW was favorable to our method. Substitution of the HNN with an HMM deteriorated the results and verified the invalidity of the Markovian assumption. The application of a distributed camera network enhanced the classification results via appropriate fusion techniques. The summarization method used was able to work in real time, to extract periodic motion and was not dependent on a predefined number of key-frames. Furthermore, it outperformed several state of the art methods.

## References


[1] D. Kosmopoulos and S. Chatzis, "Robust visual behavior recognition," *Signal Processing Magazine, IEEE*, vol. 27, no. 5, pp. 34 –45, sep. 2010.

[2] G. Wang, L. Tao, H. Di, X. Ye, and Y. Shi, "A scalable distributed architecture for intelligent vision system," *Industrial Informatics, IEEE Transactions on*, vol. 8, no. 1, pp. 91 –99, Feb. 2012.

[3] A. Hampapur, L. Brown, J. Connell, A. Ekin, N. Haas, M. Lu, H. Merkl, and S. Pankanti, "Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking," *Signal Processing Magazine, IEEE*, vol. 22, no. 2, pp. 38 – 51, march 2005.

[4] A. Voulodimos, D. Kosmopoulos, G. Vasileiou, E. Sardis, A. Doulamis, V. Anagnostopoulos, C. Lalos, and T. Varvarigou, "A dataset for workflow recognition in industrial scenes," in *IEEE Int. Conference on Image Processing*, 2011, pp. 3310–3313.

[5] M. Tenorth, J. Bandouch, and M. Beetz, "The TUM Kitchen Data Set of Everyday Manipulation Activities for Motion Tracking and Action Recognition," in *IEEE Int. Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS). In conjunction with ICCV2009*, 2009.

[6] J. K. Aggarwal and Q. Cai, "Human motion analysis: A review," *Computer Vision and Image Understanding*, vol. 73, no. 3, pp. 428 – 440, 1999.

[7] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man and Cybernetics, Part C*, vol. 34, no. 3, pp. 334–352, 2004.

[8] H. Liu, S. Chen, and N. Kubota, "Guest editorial special section on intelligent video systems and analytics," *Industrial Informatics, IEEE Transactions on*, vol. 8, no. 1, p. 90, 2012.

[9] X. Cao, B. Ning, P. Yan, and X. Li, "Selecting key poses on manifold for pairwise action recognition," *Industrial Informatics, IEEE Transactions on*, vol. 8, no. 1, pp. 168 –177, 2012.

[10] C. Tran and M. Trivedi, "3-d posture and gesture recognition for interactivity in smart spaces," *Industrial Informatics, IEEE Transactions on*, vol. 8, no. 1, pp. 178 –187, feb. 2012.

[11] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[12] D. Bruckner and R. Velik, "Behavior learning in dwelling environments with hidden markov models," *Industrial Electronics, IEEE Transactions on*, vol. 57, no. 11, pp. 3653 –3660, Nov 2010.

[13] Z. Zeng, J. Tu, B. Pianfetti, and T. Huang, "Audio-visual affective expression recognition through multistream fused HMM," *IEEE Transactions on Multimedia*, vol. 10, no. 4, pp. 570–577, June 2008.

[14] S. Eickeler, A. Kosmala, and G. Rigoll, "Hidden markov model based continuous online gesture recognition," in *In Int. Conference on Pattern Recognition (ICPR*, 1998, pp. 1206–1208.

[15] F. Lv and R. Nevatia, "Recognition and segmentation of 3-d human action using hmm and multi-class adaboost," in *ECCV06*, 2006, pp. IV: 359–372.

[16] S. Fine, Y. Singer, and N. Tishby, "The hierarchical hidden markov model: Analysis and applications," *Machine Learning*, vol. 32, no. 1, pp. 41–62, 1998.

[17] N. Padoy, D. Mateus, D. Weinland, M.-O. Berger, and N. Navab, "Workflow Monitoring based on 3D Motion Features," in *Workshop on Video-Oriented Object and Event Classification in Conjunction with ICCV 2009*. Kyoto Japan: IEEE, 2009, pp. 585–592.

[18] N. Oliver, A. Garg, and E. Horvitz, "Layered representations for learning and inferring office activity from multiple sensory channels," *Comput. Vis. Image Underst.*, vol. 96, no. 2, pp. 163–180, 2004.

[19] T. Xiang and S. Gong, "Optimising dynamic graphical models for video content analysis," *Comput. Vis. Image Underst.*, vol. 112, pp. 310–323, December 2008.

[20] H. Jaeger, W. Maass, and J. Principe, "Special issue on echo state networks and liquid state machines," *Neural Networks*, vol. 20, no. 3, pp. 287 – 289, 2007.

[21] C. Gallicchio and A. Micheli, "Architectural and markovian factors of echo state networks," *Neural Networks*, vol. 24, no. 5, pp. 440 – 456, 2011.

[22] C. Cotsaces, N. Nikolaidis, and I. Pitas, "Video shot detection and condensed representation. a review," *Signal Processing Magazine, IEEE*, vol. 23, no. 2, pp. 28 –37, March 2006.

[23] Y. Su, R. Qian, and Z. Ji, "Surveillance video sequence segmentation based on moving object detection," in *Computer Science and Engineering, 2009. WCSE '09. Second International Workshop on*, vol. 1, Oct. 2009, pp. 534 –537.

[24] L. S. da Silva and J. Scharcanski, "Video segmentation based on motion coherence of particles in a video sequence," *IEEE Transactions on Image Processing*, vol. 19, no. 4, pp. 1036–1049, 2010.

TABLE III
COMPARISON OF VIDEO SUMMARIZATION ALGORITHMS - WR DATASET

| **Method** | **Comparison of Video Summarization Algorithms** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | $K$=5 | | $K$=10 | | $K$=15 | | Variable $K$ | |
| | Precision | Recall | Precision | Recall | Precision | Recall | Precision | Recall |
| Panagiotakis et al [29] | 58.1% | 58.9% | 79.4% | 76.5% | 81.0% | 86.2% | - | - |
| Doulamis and Doulamis [30] | 43.0% | 48.7% | 63.4% | 68.7% | 74.9% | 80.4% | - | - |
| Li at al [28] | 47.6% | 53.4% | 74.3% | 82.5% | 78.5% | 85.5% | - | - |
| The proposed method | - | - | - | - | - | - | 82.5% | 92.4% |

TABLE IV
COMPARISON OF VIDEO SUMMARIZATION ALGORITHMS - TUM KITCHEN DATASET

| **Method** | **Comparison of Video Summarization Algorithms** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | $K$=5 | | $K$=10 | | $K$=15 | | Variable $K$ | |
| | Precision | Recall | Precision | Recall | Precision | Recall | Precision | Recall |
| Panagiotakis et al [29] | 65.8% | 74.0% | 71.9% | 78.2% | 88.2% | 93.7% | - | - |
| Doulamis and Doulamis [30] | 60.4% | 68.0% | 65.3% | 72.8% | 82.1% | 87.6% | - | - |
| Li et al [28] | 72.1% | 75.4% | 75.2% | 85.3% | 84.7% | 90.3% | - | - |
| The proposed method | - | - | - | - | - | - | 90.3% | 94.6% |


[25] A. Doulamis, N. Doulamis, and S. Kollias, "Non-sequential video content representation using temporal variation of feature vectors," *Consumer Electronics, IEEE Transactions on*, vol. 46, no. 3, pp. 758 –768, Aug 2000.

[26] F. Chen and C. De Vleeschouwer, "Formulating team-sport video summarization as a resource allocation problem," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 21, no. 2, pp. 193 –205, Feb. 2011.

[27] C. Choudary and T. Liu, "Summarization of visual content in instructional videos," *Multimedia, IEEE Transactions on*, vol. 9, no. 7, pp. 1443 –1455, Nov. 2007.

[28] Z. Li, G. Schuster, and A. Katsaggelos, "Minmax optimal video summarization," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 15, no. 10, pp. 1245 – 1256, Oct. 2005.

[29] C. Panagiotakis, A. Doulamis, and G. Tziritas, "Equivalent key frames selection based on iso-content principles," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 19, no. 3, pp. 447 –451, 2009.

[30] A. Doulamis and N. Doulamis, "Optimal content-based video decomposition for interactive video navigation," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 14, no. 6, pp. 757–775, 2004.

[31] D. Liu, G. Hua, and T. Chen, "A hierarchical visual model for video object summarization," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 12, pp. 2178 –2190, Dec. 2010.

[32] T. Xiang and S. Gong, "Beyond tracking: modelling activity and understanding behaviour," *International Journal of Computer Vision*, vol. 67, pp. 21–51, 2006.

[33] S. Chen, J. Zhang, Y. Li, and J. Zhang, "A hierarchical model incorporating segmented regions and pixel descriptors for video background subtraction," *Industrial Informatics, IEEE Transactions on*, vol. 8, no. 1, pp. 118–127, 2012.

[34] J. Flusser, B. Zitova, and T. Suk, *Moment Functions in Image Analysis: Theory and Applications.* Wiley, 2009.

[35] R. Luo and C.-C. Chang, "Multisensor fusion and integration: A review on approaches and its applications in mechatronics," *Industrial Informatics, IEEE Transactions on*, vol. 8, no. 1, pp. 49 –60, Feb. 2012.

[36] D. Specht, "Enhancements to probabilistic neural networks," in *Neural Networks, 1992. IJCNN., International Joint Conference on*, vol. 1, jun 1992, pp. 761 –768 vol.1.

[37] S. Haykin, *Neural Networks: A Comprehensive Foundation (2nd Edition)*. Prentice Hall, 1998.

[38] S. Abe, "Theories on the Hopfield neural networks," in *International Joint Conference on Neural Networks*, Jun 1989, pp. 557 –564 vol.1.

[39] Y. Pritch, A. Rav-Acha, and S. Peleg, "Nonchronological video synopsis and indexing," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 11, pp. 1971 –1984, Nov. 2008.

[40] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.


**Dimitrios Kosmopoulos** Dimitrios Kosmopoulos received the BEng degree in Electrical and Computer Engineering from the National Technical University of Athens (NTUA) in 1997 and a PhD from the same institution in 2002. He is with the CBIM Lab at Rutgers University. Previously he was with the University of Texas at Arlignton, NCSR Demokritos, and NTUA (Greece). His research interests are in the field of computer vision, robotics and signal processing. He has published more than 50 papers and has participated in several projects.

**Athanasios Voulodimos** Athanasios Voulodimos received his Dipl.-Ing. degree from the School of Electrical and Computer Engineering of NTUA in 2007, ranking among the top 2% of his class. He also holds an MSc and a PhD from NTUA in the area of computer vision and machine learning focusing on behavior recognition from video. He has participated in several European research projects and has published more than 35 papers. His main research interests include computer vision, machine learning, as well as ubiquitous and cloud computing.

**Anastasios Doulamis** Anastasios Doulamis received the Dipl.-Ing. in Elec. and Comp. Eng. from NTUA in 1995 with the highest honor, and a PhD from the same school in 2000. Since 2006 he is tenured Assistant professor in the Tech. Univ. of Crete in the area of multimedia systems. He has served as prog. committee in several international conferences and as reviewer of IEEE journals and conferences. His research interests include non-linear analysis and multimedia content description; he is the author of more than 200 papers.