Human-centered Video Analysis for Multimedia Postproduction

Costas Cotsaces, Ioannis Marras, Nikolaos Tsapanos, Nikos Nikolaidis and Ioannis Pitas
Informatics and Telematics Institute, Centre for Research and Technology Hellas, Greece
Department of Informatics, Aristotle University of Thessaloniki, Greece
Email: {nikolaid,pitas}@aia.csd.auth.gr

***Abstract*—Semantic analysis of video has witnessed a significant increase of research activities during the last years. Human-centered video analysis plays a central role in this research since humans are the most frequently encountered entities in a video. Results of human-centered video analysis can be of use in numerous applications, one of them being multimedia postproduction. Three recently devised semantic analysis algorithms are reviewed in this paper.**

I. Introduction

During the last two decades, an increasing research interest occurred towards what one could call anthropocentric or human centered video analysis, namely, algorithms that aim to extract, describe, and organize information regarding the basic element of most videos: humans. This diverse group of algorithms processes videos from various sources and extracts a wealth of useful information related to the state (presence, identity, body posture, emotional state, etc.) and state transitions (body parts movements, activities, etc.) of individuals, interactions or communication modes between two or more humans (dialogues, social signals, etc.) and physical characteristics of humans, such as 3D body models. Results of human-centered video analysis can be combined with other semantic analysis and description tools such as object detection/localization or recognition algorithms in order to provide a more complete semantic description of a scene.

The interest of the research community for anthropocentric video analysis stems from the fact that the extracted information can be used in various important applications such as video surveillance, human-computer interfaces, ambient intelligence environments etc. One such application domain is film and games postproduction where anthropocentric video analysis results can be used in tasks such as indexing, retrieval, summarization and organization of videos, automatic semantic annotation, semantic extraction of keyframes for animation purposes, detection of humans for the initialization of matting or background/foreground segmentation algorithms etc. In this paper we will review recent research results in three areas namely frontal facial pose recognition, human head detection and object recognition.

II. Frontal Facial Pose Recognition using Features Extracted Through Discriminant Splitting

Facial image analysis tasks such as face detection and tracking, facial features detection, face recognition or verification and facial expression recognition have attracted the interest of computer vision and pattern recognition communities over the past years. In some of these facial tasks such as face recognition and facial expressions recognition, the majority of developed techniques have been designed to operate on frontal or nearly frontal face images [1], [2], [3]. Due to this fact, a face or facial expression classifier trained on frontal facial images will not be able to operate successfully on non-frontal images. As a result, techniques that recognize frontal facial poses need to be developed, so that frontal facial images can be selected among all available facial images and used as input in face recognition or facial expression recognition systems. The same problem arises in cases where multiple view video data, acquired through a convergent multi-camera setup, are available. In this case, a frontal facial pose recognition algorithm can be applied on the available video streams to identify the view that is closer to a frontal one. By doing so, frontal images of the person can be acquired and fed to a face or facial expression recognition technique that requires frontal faces.

Frontal face pose recognition is essentially a two-class classification problem (frontal vs non-frontal). However, since the non-frontal class is much richer, as it contains all possible head orientations except for the frontal one, it can be split into a number of classes, each containing non-frontal images where the head orientation lies within a range of angle values. Obviously, in such a case all facial images classified to one of the non-frontal classes are labelled as non-frontal.

The frontal facial pose recognition technique described in this section segments the facial images to discriminant regions. The main idea is the creation of a set of regions that is discriminative for each class of facial images in the sense that a subset of these discriminant and homogeneous regions will provide adequate information in order to distinguish this class from another one. The entire set is necessary in order to distinguish this class from the rest of the other classes. The region segmentation is based on the classical image splitting technique. The features that this method uses are the mean intensities of the produced regions. Details about the method will be provided in the following subsections.

A. Feature Extraction Using Discriminant Splitting

Let us assume that there exist n facial image classes namely one class containing frontal facial views (including small

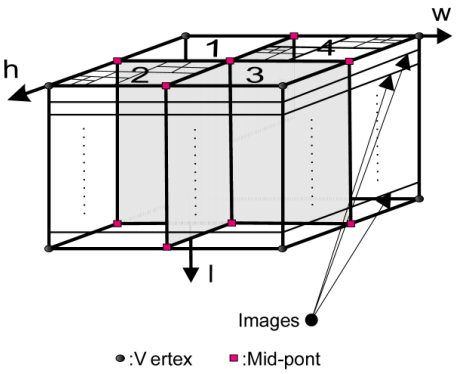

Fig. 1. Class representation as a stack of images.

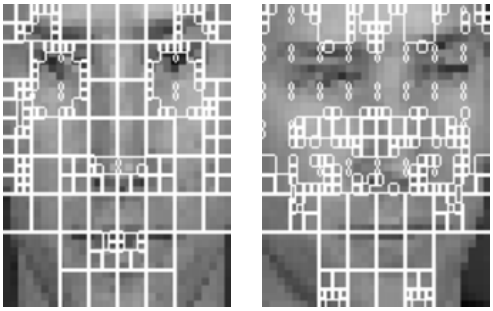

Fig. 2. Facial images from the XM2VTS database that belong to the frontal class and one of the non-frontal classes, along with the corresponding region patterns.

deviations from the fully frontal view) and $n-1$ classes corresponding to non-frontal views. Each class contains l different, equally sized, training facial images. Each of the non-frontal classes contains images where the head rotation angle with respect to the vertical axis (yaw) lies within a certain interval. Thus, the dataset D is divided into n sets $D = \bigcup_{i=1}^{n} \mathcal{U}_i$. The main goal is to find homogeneous regions that are discriminant between the classes. In this way, for each class, a unique regions pattern, i.e. a set of regions, is created. This procedure, that is based on a splitting approach, will be described bellow.

Let two classes a, b each containing l samples (images) of the corresponding facial class, in sets $\mathcal{U}_a$, $\mathcal{U}_b$. If each image is of dimensions $h \times w$, these l images can be considered as a stack of slices (volume) with dimensions $l \times h \times w$. Thus for our purpose, a certain region B can be considered as being a parallelepiped volume comprising of the parts of every image in the class that fall within the region, as illustrated in Figure 1. We assume that an image I is divided into R regions. For a region B defined as above and for a class a we define its discriminant power, with respect to class b, using the Fisher's discriminant ratio $F_{a,b}(B)$ [4]. A region B_1 is more discriminant than a region B_2, for a particular pair a, b of facial classes, when $F_{a,b}(B_1) > F_{a,b}(B_2)$. As mentioned above, except from the discriminant power of a region the method exploits also its homogeneity. As in the case of region discriminant power calculation, the homogeneity of a region is judged based on the pixels intensity values of the parts of *all* the class's training images that fall within the region's boundaries, i.e. on all pixels of the corresponding volume.

In order for the discriminant and homogeneous regions to be determined for each class a, the classical splitting approach is applied to the l images of this class. The corresponding stack of images is recursively split into four quadrants or regions (Figure 1), until 2D discriminant and non-homogeneous regions are encountered. The splitting is performed by bisecting the rectangular regions (in the entire image stack) in the vertical and horizontal directions. In short, if a region is very discriminant for a class it is being split, whereas if it is not discriminant enough, it is split if it is inhomogeneous.

The above procedure is performed for each facial class separately. In the end of the training procedure, a region pattern for each class is created. Two facial images that belong to different classes along with the corresponding region patterns are shown in Figure 2.

Finally, each training image I_k within a class a is characterized by the vector $\underline{\mu}_{ak}$ that contains the mean intensity values for each of the regions $r_{a,j}$, $j = 1 \ldots n_a$, n_a being the number of regions in the pattern of class a.

The rationale behind the splitting procedure outlined above for the creation of the region pattern for each class is that since each region is finally represented by its mean value, large regions are represented in a very coarse way, since they are represented by a single value, whereas an area split into many small regions is represented in a more refined and detailed way, as every such small region is represented by its own mean value. Thus, the algorithm splits regions that are discriminant for a certain class into smaller regions, in order to represent these regions with finer detail, which is important for classification due to their discriminant power. The fact that the method also splits non-discriminant regions that are inhomogeneous helps the fine-tuning of the region placement in the testing (classification) procedure.

B. Image Classification

The algorithm's testing (classification) procedure is as follows. The n_a discriminant regions r_{aj} of class a are selected upon an image I depicting a face in an unknown orientation. In order to solve small alignment problems, the regions boundaries are translated locally by small amounts until they fall on as much as possible homogeneous regions. For a class a, the intensity means $\mu_{Ir_{aj}}$ of every region r_{aj} of class a are computed in I, providing the image I means vector $\underline{\mu}_{Ia}$. The image means vector $\underline{\mu}_{Ia}$ is then compared with the (pre-computed) means vectors $\underline{\mu}_{ak}$ for all training images I_k ($k = 1 \ldots l$) of facial class a, resulting in distances $d_{Ia_k} = \|\underline{\mu}_{Ia} - \underline{\mu}_{ak}\|$ for every training image k that belongs to class a. Thus, l distances are computed for each class. This is repeated for all n_{Total} classes resulting into $l \cdot n_{Total}$ distances. The facial image is classified to the class α^*, that contains the training image κ^* whose means vector is closest to the test

image means vector,

$$(\alpha^*, \kappa^*) = \arg\min_{\alpha,\kappa} d_{Ia_k}. \quad (1)$$

C. Experimental Performance Evaluation

The proposed method was evaluated on data obtained from the XM2VTS face database [5]. Face tracking was applied on the head rotation shot videos, depicting people that start from a frontal pose, turn their heads to their right extreme, back to frontal pose then to the left extreme (Figure 3). The resulting facial images, that depict the face bounding box (Figure 4), were then rescaled. 6862 facial images were obtained, 2486 being frontal and 4376 non-frontal. Images where the head rotation is in the range $[-10^0 \dots 10^0]$, zero degrees being the frontal orientation, were considered as frontal. The non-frontal images were split into four classes. We then randomly split the images in half for all five classes to form the training and test sets. The proposed algorithm was found to be able to classify facial images to frontal and non-frontal with very satisfactory accuracy. Indeed the correct classification percentage achieved by the proposed method was 98%.

Fig. 3. A frame from the XM2VTS database.

Fig. 4. Frontal (top row) and non-frontal (bottom row) facial images from the XM2VTS database.

III. Head Detection Using Template Matching and Histograms of Oriented Gradients

Human head detection is a crucial building block for many algorithms, such as face recognition, facial expression recognition, human detection etc. An algorithm that detects human heads in images or video frames has been developed. The method combines a fast shape matching technique with a strong object (head in our case) detector/classifier in order to achieve improved performance in both tasks.

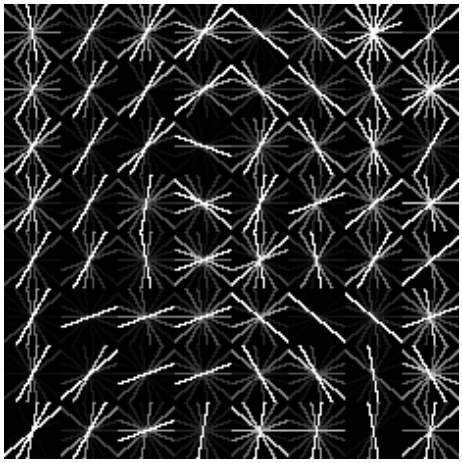

Fig. 5. The visualization of the edge orientation histograms for the image in Figure 6. There are 9 bins for every cell. Each bin is represented by a line of the appropriate orientation and the luminance of the line depends on the normalized value of the bin.

The shape matching technique uses image edges as input and is based on a binary search tree [6] that has been trained with "head and shoulders" shape templates. Each such template consists of a contour that outlines the silhouette of human head and shoulders. The method generates a list of possible template matches on the input image. The input features (image edges) are presented to the root of the tree as input. The root and every subsequent internal node decides whether to direct the search to its left or right child until the search reaches a leaf node. The template corresponding to that leaf node is selected as a candidate and the search continues by reversing the search decision at nodes with weak decisions.

The HOG classifier uses Histograms of Oriented Gradients [7] as features to feed into a Support Vector Machine (SVM) in order to classify the input as head or "not-head". The HOG features are computed by evaluating the image gradients in non-overlapping 8×8 pixel cells and distributing them into a 9-bin histogram according to their orientation and magnitude (Figure 5). Subsequently, the histograms of overlapping blocks consisting of 2×2 cells are normalized and concatenated into a feature vector.

To achieve head detection we scan the image with a sliding window and use the shape tree to obtain possible matches of the head and shoulders contour templates on each window position (Figure 6). Then we collect the blocks that the template contour goes through (Figure 7) into a feature vector. This process is used both on a set of training images in order to obtain the training feature vectors for the SVM as well as on the test images that are to be classified. In order to obtain negative examples, the template matching and feature vector extraction methods were used on images that do not contain any human heads.

By combining these techniques we can improve the shape matching performance by validating a shape match provided by the shape tree with a classification result, thus reducing the

Fig. 6. A head and shoulders contour template matched on an image.

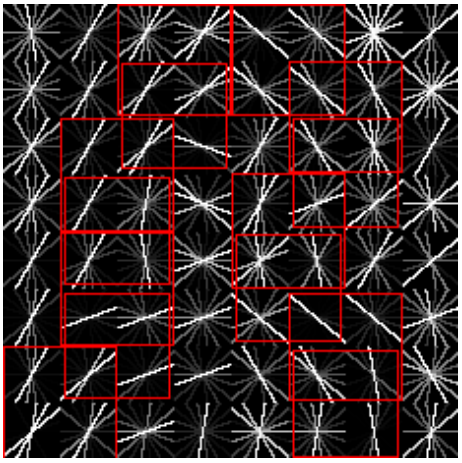

Fig. 7. The blocks selected by the contour to be included in the feature vector.

number of false positive matches. The classification rate of the HOG classifier is also improved since we do not use the entire image window that most probably contains background in it, but rather only the blocks relevant to the object as input.

IV. Multiview Object Recognition Utilizing a Bag of Keypoints Approach

Localization and recognition of objects within a scene provides information that can complement and enrich information related to humans. Object recognition is quite unlike most other tasks in computer vision, for example face recognition, person detection, emotion recognition etc. This is because a class of objects may encompass a vast variety of different entities, both ontologically and visually. Moreover, in the case of objects, the border between recognition ("which one") and categorization ("what type of") is very blurry. This is due to the fact that, unlike humans, objects do not have a distinct identity, and many objects can have multiple nearly identical copies (e.g cars of the same model). Another characteristic of object recognition is that, with the exception of a few classes of radially symmetric objects, most objects exhibit a high visual variety from different views. Due to this fact, the fusion of information from multiple cameras is of great help in object recognition. Thus, we have attempted to design a system that performs object recognition (or categorization) using information from (initially) two cameras. Instead of designing a system from scratch, we have chosen to extend a successful generic single-camera object recognition and categorization system [8] which has been recently shown [9] to have a performance close to the state of the art. The following subsection reviews the single-camera algorithm whereas subsection IV-B describes the proposed multi-view approach

A. Single Camera Framework

The single camera method used [8] was founded on the current trends in computer vision, namely the use of local feature points, and decision using SVMs.

The fundamental steps for the training phase of the single-camera method are the following:

1) Extraction of local feature points from all (labelled) images of a training set. Local feature analysis methods consist of two generally independent components, a feature point detector, and a feature point descriptor. In the present case, the feature point detector that was selected was the Harris affine detector [10]. The advantage of this detector is that is especially robust to transformations. The feature descriptor, respectively, is the classic SIFT descriptor [11] which consists of a set of Gaussian derivatives computed at 8 orientation planes over a 4×4 grid of spatial locations, giving a 128-dimensional vector.
2) Clustering of local feature descriptors into a number of classes. The clustering of local feature descriptors is done in order to abstract the distribution of the feature points. The k-means algorithm is used for this purpose. The number of classes k is decided experimentally, and generally ranges around 1000. The feature that is used for the assignment of a specific feature point to a cluster is the SIFT descriptor. At the end of the clustering procedure, only the cluster centers are retained.
3) Computation of a summary descriptor (feature vector) for each image in the training set. A histogram representing the number of feature points that were assigned to each class center is used as feature vector.
4) Training of a classifier using the feature vectors of all images. Depending on the separation of the training example images into classes, the classification can lead to object recognition, object class recognition (a.k.a. object categorization), object verification etc. In the present case we have chosen to implement object recognition. The classifier that was selected was the classic SVM. Different types were tried, but the simplest linear variety was found to be most effective. Since SVMs are intrinsically a two-class classifier, we use the classic multi-class extension whereby a classifier is trained for each pair of classes and the final recognition of an image is done by a voting procedure, with each classifier contributing a vote to the class it selects.

For recognizing an object in an incoming image (testing stage), a similar procedure is followed: Harris-affine feature points are detected, and their SIFT descriptors are extracted. These descriptors are then assigned into the previously computed cluster centers, and the number of feature points assigned to each center forms a histogram, which is then passed to the previously trained Support Vector Machine which makes the final decision about which object is depicted in the image.

B. Multi-camera Framework

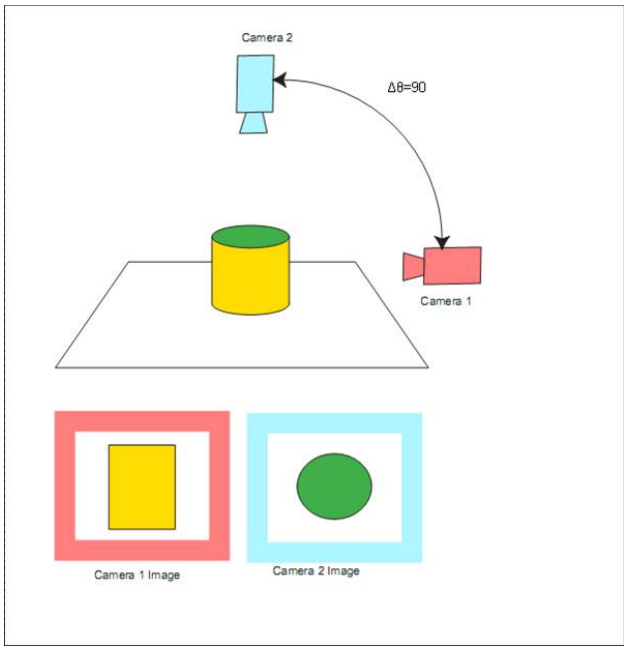

Fig. 8. Schematic of the two-camera configuration used.

We base our approach to multi-camera object recognition on the assumption that the relative spatial configuration of the (static) cameras is known a priori. Here, we are concerned only with the positions of the cameras with respect to the object, i.e. their relative position in a coordinate system rigidly attached to the observed object. Although the proposed approach can be applied to an arbitrary number of cameras, we will limit the discussion to the two-camera problem.

Let us then assume cameras C_1 and C_2 as in Figure 8. Assuming a spherical coordinate system centered on the center of the object, the positions of the two cameras are $\{\rho_1, \phi_1, \theta_1\}$ and $\{\rho_1, \phi_1, \theta_1\}$. Since the keypoint bag recognition method described in the previous subsection is largely scale invariant, the only relevant parameters in this case are the angle differences, normalized to lie between 0 and 2π

$$\Delta\phi = \phi_1 - \phi_2 + 2n\pi, n \in \{0, 1\} \quad (2)$$

and

$$\Delta\theta = \theta_1 - \theta_2 + 2n\pi, n \in \{0, 1\} \quad (3)$$

Thus, in the general case, a different classifier would need to be trained for each combination of $\Delta\phi$ and $\Delta\theta$, i.e. for each spatial configuration of the two cameras.

The classifier that was used is a modification of the basic keypoint bag system described in subsection IV-A. In essence, instead of a single keypoint histogram corresponding to the one camera, two keypoint histograms H_1 and H_2 are created, from the images corresponding to the two cameras. Each histogram is, as before, computed by classifying the SIFT descriptors into pre-computed bins. H_1 and H_2 are then concatenated into a combined histogram H, which is then processed as usual by the SVM stage.

The main difference is in the training stage. Firstly, in the training data, each training image X is labelled not only with the identity of the object in question, but also with the angles ϕ_x and θ_x from which it was imaged. An arbitrary "frontal" pose is chosen as a zero axis. Then, local features are extracted and the k-means centers are established as described in subsection IV-A, without taking camera orientation into account. Likewise, the histograms for each image are computed irrespective of camera orientation, using the class centers generated by the k-means. But in order to compute feature vectors that will form the training data of the SVM, for each image X in the training database, other images Y belonging to the same class are sought such that

$$\phi_y - \phi_x = \Delta\phi + 2n\pi, n \in \{0, 1\} \quad (4)$$

and

$$\theta_y - \theta_x = \Delta\theta + 2n\pi, n \in \{0, 1\} \quad (5)$$

. Then the histograms of the two images H_x and H_y are concatenated into histogram H which is used to train the SVM classifier.

C. Experimental Performance Evaluation

Due to the lack of challenging multi-view object detection/recognition databases we created our own database. We selected to focus on furniture, and particularly on chairs. Chairs are a recurrent object in human environments, and present particular difficulties since they are topologically very diverse, highly concave, and have large gaps in their silhouette.

Fig. 9. Example of the images captured for the chairs database.

For our training set, 8 distinct chairs were selected as shown in Figure 9. For each chair, 16 different camera orientations were defined, by sampling ϕ and θ. In all cases, r was constant. Within each orientation, 6 different photographs were taken, by randomly varying ϕ and θ within $\pm 10^\circ$ and r by $\pm 15cm$. We thus gathered 96 images per object. In general, the objects in question took up approximately 50% of the area of each image (including gaps in the objects).

To achieve a fair comparison we compared the result of the multi-camera recognizer with the result of the merging of two single-camera classifiers, each operating on one of the two images that form the input to the multi-camera algorithm. When the two single camera results are different, the merging is achieved by selecting the one with the greatest SVM margin.

We trained both methods using $2/3$ of the images. Then the other $1/3$ of the images were used for testing and the process was repeated three times. The result of the combination of the two single-camera classifiers was an accuracy of 96%, while the two-camera classifier had an accuracy of 100%

V. Conclusion

Semantic analysis of videos is a very challenging research topic, especially when targeting humans (identity, state, expressions, actions etc). Despite the recent progress and the numerous applications (one of them being multimedia post-production) much remains to be done until the corresponding algorithms reach maturity and good levels or robustness and accuracy. This is especially true in cases where such algorithms need to operate in an unconstrained environment with light variations, cluttered background etc. The use of multiple video feeds, from calibrated, synchronized cameras, or the existence of 3D information is expected to greatly facilitate efforts towards this direction.

Acknowledgment

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 211471 (i3DPost).

References

[1] W. Zhao, R. Chellappa, A. Rosenfeld, and P. J. Phillips, "Face recognition: A literature survey," *ACM Computing Surveys*, pp. 399–458, 2003.

[2] B. Fasel and J. Luettin, "Automatic facial expression analysis: A survey," *Pattern Recognition*, vol. 36, pp. 259–275, 1999.

[3] Z. Zeng, M. Pantic, G. Roisman, and T. Huang, "A survey of affect recognition methods: Audio, visual, and spontaneous expressions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 1, pp. 39–58, January 2009.

[4] K. Fukunaga, *Statistical Pattern Recognition*. Academic Press, San Diego, CA, 1990.

[5] K. Messer, J. Matas, J. Kittler, and K. Jonsson, "XM2VTSDB: The extended M2VTS database," in *2nd International Conference on Audio and Video-based Biometric Person Authentication*, 1999, pp. 72–77.

[6] N. Tsapanos, A. Tefas, and I. Pitas, "Online shape learning using binary search trees," *Image and Vision Computing*, vol. 28, no. 7, pp. 1146–1154, 2010.

[7] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *International Conference on Computer Vision & Pattern Recognition*, vol. 2, June 2005, pp. 886–893.

[8] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *ECCV International Workshop on Statistical Learning in Computer Vision*, 2004.

[9] J. Zhang, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: a comprehensive study," *International Journal of Computer Vision*, vol. 73, no. 2, pp. 213 – 238, June 2007.

[10] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *European Conference on Computer Vision (ECCV)*, 2002, pp. 128–142.

[11] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.