Facial Features Matching using a Virtual Structuring Element

Roberto Valenti Nicu Sebe Theo Gevers

Intelligent Systems Lab Amsterdam,
University of Amsterdam,
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands

ABSTRACT

Face analysis in a real-world environment is a complex task as it should deal with challenging problems such as pose variations, illumination changes and complex backgrounds. The use of active appearance models for facial features detection is often successful in restricted environments, but the performance decreases when applied in unconstrained environments. Therefore, in this paper, we introduce a novel method that integrates the knowledge of a face detector inside the shape and the appearance models by using what we call a 'virtual structuring element' (VSE). In this way the possible settings of the active appearance models are constrained in an appearance-driven manner. The use of a virtual structuring element in an active appearance model provides increased performance in both accuracy and robustness over standard active appearance models applied to different environments.

Keywords: Face Analysis, Facial Features Matching, Active Appearance Models, VSE

1. INTRODUCTION

Automatic face analysis has attracted increasing interest in the research community mainly due to its many useful applications. Various approaches to facial feature detection exist in the literature. Some of the most common ones make use of hand-crafted geometric features models,[1] separate face and facial feature detectors,[2] color segmentation,[3] or neural networks.[4] Although these and related methods have been shown to achieve good results, they mainly focus on finding the location of some facial features (e.g., eyes and mouth corners) in restricted environments (e.g., constant lighting, simple background, etc.) and are not always suitable to obtain a complex and accurate system of features.

In recent years deformable model-based approaches for image interpretation have been proven to be successful, especially in images containing objects with large variability such as faces. These approaches are more appropriate to locate a predefined set of features since they make use of a template (e.g., the shape of an object). Among the early deformable template models is the Active Contour Model by Kass et al.[5] in which a correlation structure between shape markers is used to constrain local changes. Cootes et al.[6] proposed a generalized extension, namely Active Shape Models (ASM), where deformation variability is learned using a training set. Active Appearance Models (AAM) were later proposed in[7] and are closely related to the simultaneous formulation of Active Blobs[8] and Morphable Models.[9] AAM can be seen as an extension of ASM which includes the appearance information of an object.

While active appearance models have been shown to be successful, they suffer from important drawbacks such as background handling and initialization. Therefore, in this paper, we want to go one step further and reduce the existing AAM problems by considering the initialization information as part of the shape and appearance models. Our goal is to enhance the AAMs so they can be used in uncontrolled environments. This comes at the cost of adding another constraint, that is, the successful detection of the face by the face detector.

The rest of the paper is structured as follows. The next sections will briefly discuss the theory behind the used face detector and the AAM. After analyzing the problems in the AAM search, we will introduce the VSE as a possible solution. Finally, we will evaluate and discuss the results by using the VSE in a real system for real-time automatic facial expression recognition.

Further author information: (Send correspondence to Roberto Valenti)
Roberto Valenti: E-mail:rvalenti@science.uva.nl, Telephone: +31 (0)20 525 7540
Nicu Sebe: E-mail:nicu@science.uva.nl, Telephone: +31 (0)20 525 7580
Theo Gevers: E-Mail:gevers@science.uva.nl, Telephone: +31 (0)20 525 7516

2. FACE DETECTOR

The used face detector is the one proposed by Viola and Jones[10] and later improved by Lienhart et al.[11] The detector uses a method proposed by Papageorgiou et al.[12] to analyze image features by using a subgroup of Haar-like features, derived from the Haar transforms. The Haar-like features are the input to basic decision–trees classifiers. By means of these features it is possible to account for relations in differences of pixel intensities over specific areas of the input image.

The success of detector is due to the following techniques:[10]

1. A new way to represent an image, called 'Integral Image', easily generated by the cumulative sum of the pixels of the original image and used to efficiently retrieve the Haar-like features in an image.
2. A method to construct a classifier by selecting a small number of relevant features using 'Adaboost',[13,14] based on the Probably Approximately Correct framework (PAC[15]). The algorithm is used to select which of the features in the training set are actually relevant for the sought–after object, and to drastically reduce the number of features to be analyzed in the test set.
3. A method to successively combine more complex classifiers: the first classifier (the most discriminative and less complex) is applied to all the sub-windows of the image and at different scales. The second classifier (more complex than the previous) will be applied only to the sub-windows where the first classifier succeeded. The cascade continues, applying the all sequence of classifiers and discarding the negative sub–windows, concentrating the computational power only on the promising areas.

3. ACTIVE APPEARANCE MODELS

The main idea behind the AAMs approach is to learn the possible variations of facial features exclusively on a probabilistic and statistical basis of the existing observations (*i.e.* which relation holds in all the previously seen instances of facial features). This can be defined as a combination of shape and appearance models.

3.1 Shape Models

We can define a 2D vector representation of the relevant points of a planar shape S as the coordinates of the n points that make up the shape

Since the shapes in a dataset are often sampled with arbitrary translation, rotation and scale, all those attributes should be removed. A common solution is to normalize the shapes using *Procrustes Analysis* and to apply *Principal Component Analysis* to the normalized shapes. In statistics, procrustes analysis is a technique used to analyze the statistical distribution of shapes. This consists of four steps: Compute the centroid of each shape, re-scale each shape to have equal size, align the shapes position w.r.t. their centroids and finally rotate the shapes to align them w.r.t. their orientation.

After all the shapes are aligned it is possible to give an estimate of the prototype shape as a point-wise mean of all the shape points.

Aligned shapes of the same objects will still have some inherent shape variance. A statistical method of dealing with this redundancy is Principal Component Analysis (or PCA[16,17]). In this specific case, PCA is performed as an Eigen analysis of the dispersion matrix (the covariance matrix) of the aligned shapes, estimated with respect to the mean shape $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$, by the maximum likelihood

$$\Sigma_s = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(x_i - \bar{x})^T \tag{1}$$

After performing PCA, an arbitrary shape S can be expressed as a base shape $\bar{s}$ plus a linear combination of shape vectors s_i

$$S = \bar{s} + \sum_{i=1}^{n} p_i s_i \tag{2}$$

where p_i are the shape vector parameters.

3.2 Appearance Models

The appearance of an object is defined as the pixel values that represent the object. If we are interested in the appearance of an object contained in a shape, its appearance is then defined as the pixel values contained by the convex hull of shape. As we did for the shapes, the appearances are normalized by warping them to a reference shape (*i.e.* the mean shape $\bar{s}$). This is done by a piece-wise affine warping of the Delaunay triangulation of the shape. Once all the appearances of the object are warped to the mean shape we should remove the influence from the global linear changes in pixel intensities by applying photometric normalization.

A compact PCA representation is derived to deform the texture in a manner similar to what is observed in the training set. After all of the N shape appearances are normalized the mean shape appearance can be computed as

$$\bar{a} = \frac{1}{N}\sum_{i=1}^{N} a_i \tag{3}$$

while the estimate of the covariance matrix is

$$\Sigma_a = \frac{1}{N}\sum_{i=1}^{N}(a_i - \bar{a})(a_i - \bar{a})^T \tag{4}$$

After a PCA is performed and appearance vectors found, every shape's appearance A can be expressed as a base appearance $\bar{a}$ plus a linear combination of n appearance vectors a

$$A = \bar{a} + \sum_{i=1}^{n} p_i a_i \tag{5}$$

where p_i are the appearance parameters. Stegmann[18] shows a way to reduce the computational load while calculating the eigenvectors.

3.3 Combined Models

We can now generate an instance of a shape starting from a mean shape and a linear combination of shape vectors in the same way that we can generate an appearance from a mean appearance and a linear combination of appearance vectors (see equations 2 and 5).

Any training sample can be summarized by the parameter vectors p_s and p_a. The shape parameters have units of distance, while the appearance parameters have units of intensity. This makes p_s and p_a not directly comparable, unless we estimate the effects of using the shape parameters p_s over the appearance A and consider a weighting factor of the shape parameters to correct the differences.

As shown in,[7] this can be done by systematically displacing each element of p_s from its optimum value on each training example, and consequently sample the image under the displaced shape. The Root Mean Square change in the appearance per unit change in shape gives the sought-after weighting vector. Using this vector we can concatenate the shape and the appearance parameters in a single vector c

$$c = \begin{pmatrix} \mathbf{W}_s p_s \\ p_a \end{pmatrix} \tag{6}$$

where $\mathbf{W}_s$ is the diagonal matrix of the obtained weights.

A third and final PCA is performed on the concatenated shape and texture parameters to remove the correlation between the shape and the appearance parameters and to obtain the combined model parameters m.

$$c = \mathbf{\Phi} m \tag{7}$$

The shape and appearance parameters are linearly re-parameterized in terms of the new eigenvectors of the combined PCA.

Using linear algebra, a combined shape–appearance model is generated by

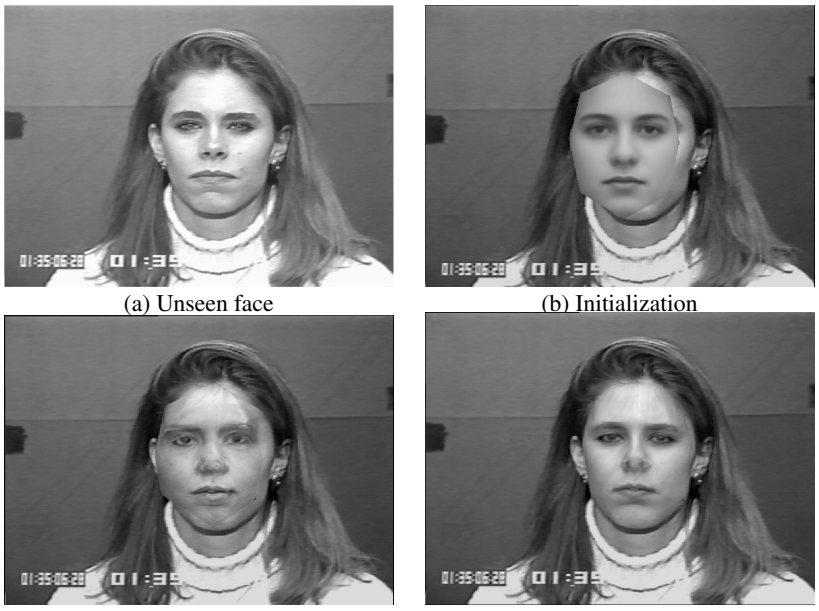
$$S = \bar{s} + \sum_{i=1}^{n} m_i s_i \quad A = \bar{a} + \sum_{i=1}^{n} m_i a_i \tag{8}$$

Note the use of the same model parameters m in both the equations. Another technique to instantiate a combined shape–appearance model consists in warping the appearance obtained with p_a to the shape obtained with p_s. This technique will not merge the models, but instead treats the shape and the appearance parameters independently. This technique is known in literature as 'Independent AMMs'.

3.4 Basic Active Appearance Models

The basis of AAM search is to treat the fitting procedure of a combined shape-appearance model as an optimization problem by minimizing the difference vector between image $\mathbf{I}$ and the generated model $\mathbf{M}$ of shape and appearance: $\delta\mathbf{I} = \mathbf{I} - \mathbf{M}$. Cootes et al.[7] observed that each search corresponds to a similar class of problems where the initial and the final model parameters are the same. This class can be learned offline (when we create the model), saving high-dimensional computations during the search phase.Learning the class of problems means that we have to assume a relation $\mathbf{R}$ between the current error image $\delta\mathbf{I}$ and the needed adjustments in the model parameters m. The common assumption is to use a linear relation: $\delta m = \mathbf{R}\delta\mathbf{I}$. Despite the fact that more accurate models are proposed,[19] the assumption of linearity was shown to be sufficiently accurate to obtain good results.[7] To find $\mathbf{R}$ we can conduct a series of experiments on the training set, where the optimal parameters m are known. Each experiment consists of displacing a set of parameters by a know amount and then measuring the difference between the generated model and the image under it. Note that when we displace the model from its optimal position and calculate the error image $\delta\mathbf{I}$, the image will surely contain parts of the background, which will have high variance for uncontrolled environments.

For the iterative optimization procedure using the found predictions the following computational steps are taken: The first step is to initialize the mean model in an initial position and the parameters within the reach of the parameter prediction range (which depends on the perturbation used during training). Iteratively, a sample of the image under the initialization is taken and compared with the model instance. The differences between the two appearances are used to predict the set of parameters that would perhaps improve the similarity. In case a prediction fails to improve the similarity, it is possible to damp or amplify the prediction several times and maintain the one with the best result.



(a) Unseen face (b) Initialization

(c) Model Optimizations (d) Converged Model

Figure 1. Results of an AAM search on an unseen face

For an overview of some possible variations to the original AAMs algorithm refer to.[20] An example of the AAM search is shown in Fig. 1 where a model is fitted to a previously unseen face.

4. VIRTUAL STRUCTURING ELEMENT

One of the main drawbacks of the AAM is that, when the algorithm learns how to solve the optimization offline, the perturbation applied to the model inevitably takes parts of the background into account. This means that instead of learning how to generally solve the class of problems, the algorithm actually learns how to solve it only for the same or similar background. This makes AMMs domain-specific, that is, the AAM trained for a shape in a predefined environment has difficulties when used on the same shape immersed in a different environment. To solve this drawback an uniform background is often placed behind the test subjects, which resembles the one used during the training phase of the AAM. Another often used idea is to constrain the shape deformation within predefined boundaries. Note that a shape constraint does not adjust the deformation, but will only limit it when it is found to be invalid.

To overcome these deficiencies of AAMs, we propose a novel method to visually integrate the information obtained by a face detector inside the AAM. This method is based on the observation that an object with a specific and recognizable feature would ease the successful alignment of its appearance model.

Since faces have many highly relevant features, erroneously located ones could lead the optimization process to converge to local minima. The novel idea is to **add a virtual artifact in each of the appearances in the training and the test sets**, which will inherently prohibit some deformations. We call this artifact a virtual structuring element (or **VSE**) since it adds structure to the data that was not available otherwise.

To be able to solve the aforementioned AAM problems, we should choose a VSE that: (1) Is big enough to steer the optimization process; (2) Does not create additional uncertainty by covering relevant features (e.g., the eyes or nose); (3) Scales accordingly to the dimension of the detected face; and (4) Completely or partially removes the areas with high variance in the model, by replacing them with uniform ones (0 variance).

In our specific case, namely facial features detection, this element should add visual information about the position of the face as obtained from the used face detector. If we assume that the face detector successfully detects a face, we can use this additional information to build our artifact by using the following procedure: At first an AAM of the faces detected in training set is built. The approximate shape of the VSE is then generated by considering the areas of high variance in the model which are not covered by the average shape model (for uncontrolled environments, this will approximately correspond to the background around the faces). After the rough VSE is found, it can be refined by using morphological operators or by hand-crafting it. The refined VSE is then placed on the training data and the new AAM is trained.

In the used VSE, a black frame with width equal to 20% of the size of the detected face is built around the face itself. Besides the regular markers that capture the facial features (see Fig. 2 and[21] for details) four new markers are added in the corners to stretch the convex hull of the shape to take in consideration the virtual structuring element. Around each of those four points, a black circle with the radius of one third of the size of the face is added. The resulting annotation, shape, and appearance variance are displayed in Fig. 2. Note that in the variance map the initialization variance of the face detector is automatically included in the model (*i.e.* the thick white border delimitating the VSE).

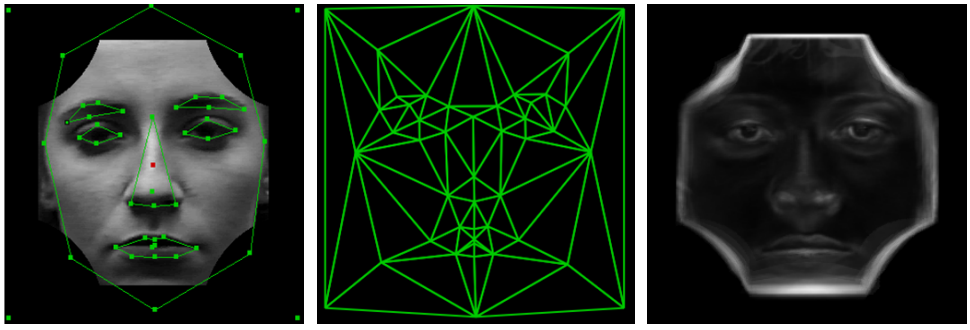


Figure 2. The effect of a virtual structuring element to the annotation, appearance, and variance (white indicates a larger variance)

This virtual structuring element, included as a feature in the training samples and automatically superimposed by the face detector to the test samples, visually passes information between the face detector and the AAM. In the experiments we show that the VSE helps the basic AAMs in the model generalization and fitting performances.

5. EXPERIMENTS

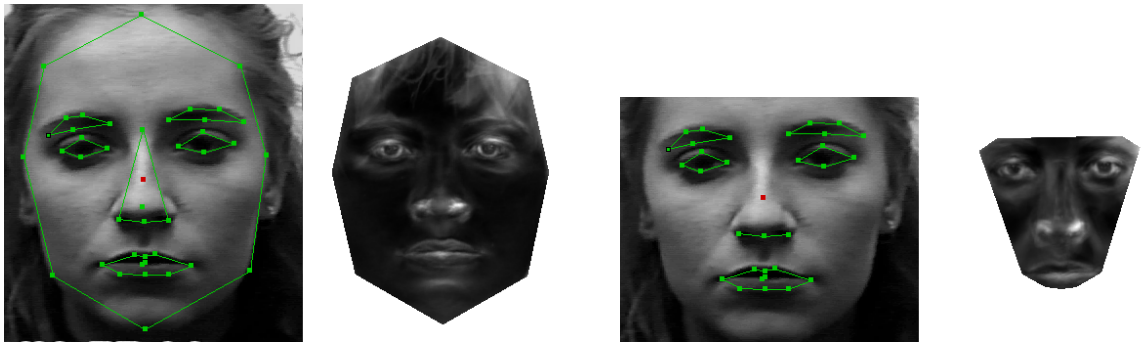
5.1 Application

The proposed technique can be used in various applications. In our case, it was used as a component in a system for real-time automatic facial expression recognition. The system involving such an analysis assumes that the face can be accurately detected and tracked, the facial features can be precisely identified, and that the facial expressions, if any, can be classified and interpreted.

The system developed in our previous work,[21,22] based on the face tracker proposed in,[23] constructs an explicit 3D wireframe model of the face starting from a generic face model consisting of 16 surface patches embedded in Bezier volumes, which are warped to fit the user's facial features. Once the model is constructed and fitted on the image of a face, head motion and local deformations of the facial features such as the eyebrows, eyelids and mouth are tracked. Then, the tracking measurements are fed into a Bayesian Network classifier that provides an estimation of the displayed expression. While the tracker has been shown to be robust to head movements, pose, partial occlusion, and it was successfully used to accurately classify facial expressions, the possibility to use the system in real applications is precluded by the requirement of manual annotation of the user's facial features in the first frame of the image sequence. As a case study, we use the proposed technique to automate or at least minimize the human intervention during the initialization phase of the system.

5.2 Datasets and Measures

Two datasets are used for evaluation: (1) a part of the Cohn-Kanade[24] dataset consisting of 53 male and female subjects, showing neutral frontal faces in a controlled environment; (2) the Unilever dataset consisting of 50 females, showing natural poses in indoor and outdoor uncontrolled environments. The idea is to investigate the influence of the VSE when the background is unchanged (Cohn-Kanade) and when more difficult conditions are present (Unilever).



(a) Relevant (b) Inside

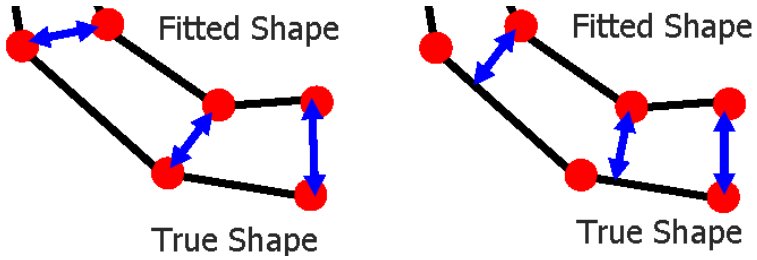
Figure 3. The annotations and their respective variance maps over the datasets

We evaluate two specific annotations: (1) 'relevant' (Fig. 3(a)) describing the facial features that are relevant for the facial expression classifiers including the face contours that are needed for face tracking; (2) 'inside' (Fig. 3(b)) describing the facial features without the face contours. Note that the 'inside' model is surrounded only by a face area so its background variance is lower and the model is more robust. To assess the performance of the AAM we initialize the mean model (*i.e.* the mean shape with the mean appearance) shifted in the Cartesian plane with a predefined amount. This simulates some extremes in the initialization error obtained by the face detector.

The common approach to assess performances of an AAM is to compare the results to a ground truth (*i.e.* the annotations in the training set). As a test tool, we used the AAM-API.[25] The following measures are used:

- **Point to Point Error**(Fig. 4(a)): is the mean Euclidean distance between each point true shape and the corresponding fitted shape:

$$\frac{1}{n}\sum_{i=1}^{n}\sqrt{(x_i - x_{gt,i})^2 + (y_i - y_{gt,i})^2} \tag{9}$$



(a) Point to Point (b) Point to Curve

Figure 4. The graphical representation of the Point to Point and the Point to Curve Errors

- **Point to Curve Error**(Fig. 4(b)): is the Euclidean distance between a fitted shape point and the closest point on the linear spline obtained from the true shape points

$$\frac{1}{n}\sum_{i=1}^{n}\min_{t}\sqrt{(x_i - r_x(t))^2 + (y_i - r_y(t))^2} \tag{10}$$

- **Mahalanobis Distance** It is possible to learn how the model's parameters are assuming a plausible configuration by looking at the configurations in the training data: when a configuration is reached, it means that the model's optimization took a specific path. During the optimization iterations, every single step is considered possible but that doesn't guarantee that the model won't assume impossible final configurations as a combination of every step. In order to avoid this behavior, a distance D from an hyper ellipsoid D_{max} could be used to restrict the possibility for the model parameters m_i (and the respective principal component λ_i) to assume certain values. This distance is known as Mahalanobis distance and is defined as follows:

$$D^2 = \sum_{i=1}^{t}\frac{m_i^2}{\lambda_i} \leq D_{max}^2 \tag{11}$$

We can use this distance to an arbitrary threshold D_{max} to assess the deformity of the obtained shapes with respect to what we have seen in the training set. If the parameters configuration exceeds the threshold, then the deformation is considered invalid.

5.3 Evaluation

We perform two types of experiments: person independent and generalized AAM. In the person independent test we perform a leave-one-out cross validation over each dataset. For the second experiment, the generalized AAM test, we merge the two datasets and we create a model which includes all the different lighting conditions, backgrounds, subject features, and annotations (together with their respective errors). The cross validation is performed over the merged dataset. The goal of this experiment is to test whether the generalization problems of AAMs could be solved just by using a larger amount of training data.

5.3.1 Person Independent

	Cohn-Kanade			Unilever		
	Point-Point	Point-Curve	Mahalanobis	Point-Point	Point-Curve	Mahalanobis
Relevant	16.72 (5.53)	9.09 (3.36)	47.93 (4.90)	54.84 (10.58)	29.82 (6.22)	79.41 (6.66)
Relevant VSE	6.73 (0.21)	4.34 (0.15)	26.46 (1.57)	10.14 (2.07)	6.53 (1.30)	24.75 (3.57)
Inside	9.53 (3.48)	6.19 (2.47)	39.55 (3.66)	25.98 (7.29)	17.69 (5.16)	38.20 (4.52)
Inside VSE	5.85 (0.24)	3.76 (0.13)	27.14 (1.77)	8.99 (1.90)	6.37 (1.46)	23.45 (2.81)

Table 1. Mean and standard error in the person independent test for the two datasets

Table 1 shows the results obtained in the two datasets in the person independent experiment. Important to notice is that the results obtained with Cohn-Kanade datasets are in most cases better than the one obtained with the Unilever

dataset since, in the latter, the effect of the uncontrolled lighting condition and background change is more relevant and the model fitting is more difficult. Furthermore, the 'inside model' is always less affected by the VSE. This comes from the property that an inside model is surrounded by the face, which is not so different from one environment to another, thus the background variance problem is reduced. However, in both cases, it can be derived that the use of a VSE significantly improves the results. An important aspect is that the use of VSE is more effective in the case of the uncontrolled Unilever database since its inconsistent background is reduced to a larger extent. Finally, while the use of a VSE does not excessively improve the accuracy of the 'inside' model, the use of the VSE on the 'relevant' model drastically improves its accuracy making it even better than the basic 'inside' model. This result is interesting since, in the 'relevant' model, parts of the markers are covered by the VSE (*i.e.* the forehead and chin markers) and we expected the final model to inherently generate some errors. Instead, it seems that the inner parts of the face might steer the outer markers to the optimal position. This means that there is a proportional relation between the facial countours and the inside features, which is a very interesting property.

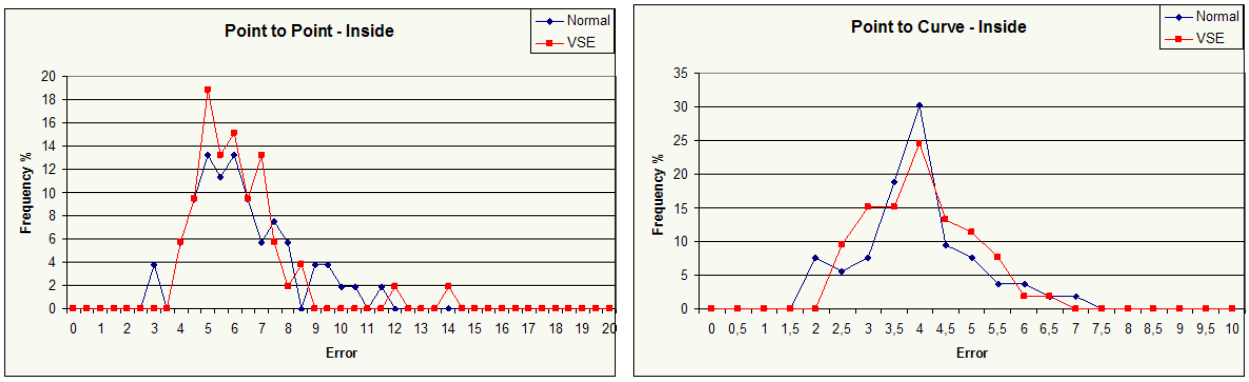


Figure 5. The graphical representation of the error distribution for the 'inside' person independent test

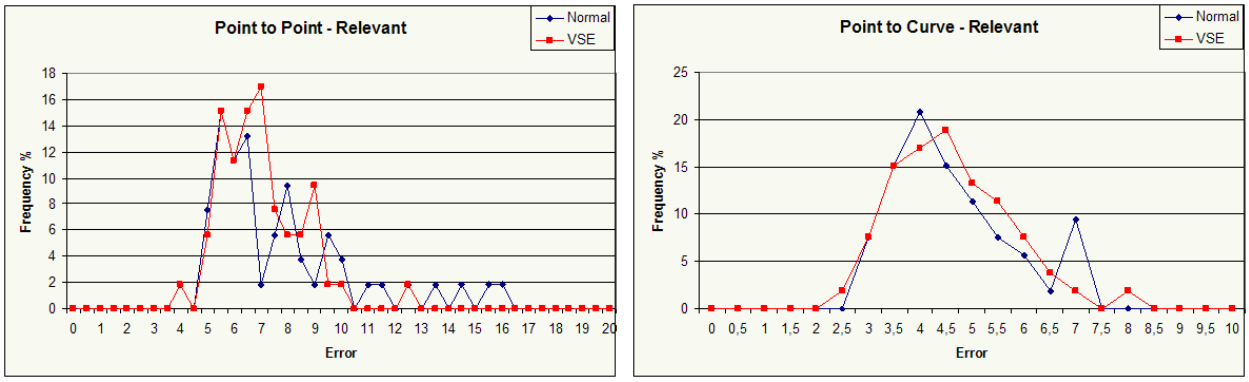


Figure 6. The graphical representation of the error distribution for the 'relevant' person independent test

The graphs in figure 5 and 6 were obtained by grouping the point to point and point to curve distance frequencies for the Cohn-Kanade dataset. In this graphical representation of the results' distribution it is clear that the VSE has the effect of redistributing some of the outliers near a median result.

Besides the robustness obtained by having both good and bad results closer to an average solution point, getting rid of the outliers diminishes the deviation from the mean accuracy. This kind of normalization effect is similar for every measures on both dataset.

	Point-Point	Point-Curve	Mahalanobis
Relevant	21.05 (0.79)	8.45 (0.27)	116.22 (3.57)
Relevant VSE	8.50 (0.20)	5.38 (0.12)	51.11 (0.91)
Inside	8.11 (0.21)	4.77 (0.10)	85.22 (1.98)
Inside VSE	7.22 (0.17)	4.65 (0.09)	52.84 (0.96)

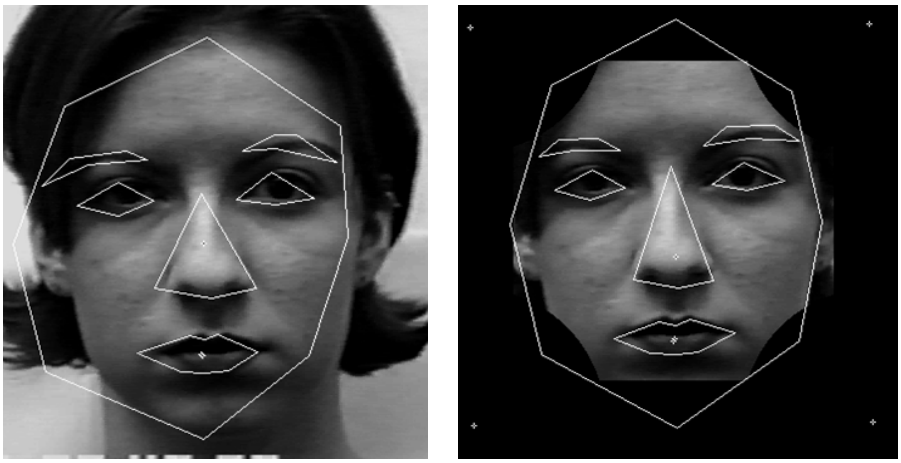
Table 2. Mean and standard error for Generalized AAM

5.3.2 Generalized AAM

In the generalized AAM experiment (see Table 2), it can be observed that the results are generally worse when compared to the person independent results on the 'controlled' Cohn-Kanade dataset, but better when compared to the same experiment on the 'uncontrolled' Unilever dataset. Also in this case the VSE implementation shows remarkable improvements over the basic AAM implementation. Note that the VSE implementation brings the results of the generalized AAM very close to the dataset specific (person independent) results, improving the generalization of basic AAM.

While the 'relevant VSE' model is better than the normal 'inside' model, the 'inside VSE' is the model of choice to obtain the best overall results on facial features detection. In our specific task (described in section 5.1), we could use the 'inside VSE', but we would additionally need some heuristics to correctly position the other markers which are not included in the model. These missing markers are relevant for robust face tracking and implicitly for facial expression classification so their accurate positioning is important. Since in the 'inside VSE' model these markers are not explicitly detected, we indicate the 'relevant VSE' model as the best choice for our purposes.

To better illustrate the effect of using a VSE, Fig. 7 shows an example of the difference in the results when using a 'relevant' model and a 'relevant VSE' model. While the first fails to correctly converge, the second result is optimal for the inner facial features. Empirically, VSE model shows to always overlap to the correct annotation, avoiding the mistakes generated by unsuccessful alignments like the one in Fig. 7(a).



(a) Relevant (b)Relevant VSE

Figure 7. An example of the difference in the results between a 'relevant' and a 'relevant VSE' model

6. CONCLUSIONS

In this paper we introduced a novel method to integrate the knowledge of a face detector inside the active appearance model, using what we call a 'virtual structuring element', which limits the possible settings of the AAM in an appearance-driven manner.

We propose this visual component as a good solution for the background variance problems and respective generalization problems of basic AAMs. We performed a comprehensive analysis of the accuracy of different Active Appearance Models for the task of facial features matching. We also improved and extended the existing real-time emotion recognition system[21] by solving the requirements of human intervention during the initialization phase, allowing a fully automated use of the system.

For future research we plan to allow VSEs to be used with any kind of objects, by automatically learning them from the background variation in the used dataset. Furthermore, while different improvements for basic AAMs are already proposed in literature,[19,20] we are planning to improve the accuracy of the system by adding more shape points (*i.e.* allowing finer refinements of the appearance models where needed).

ACKNOWLEDGMENTS

The work of Roberto Valenti and Nicu Sebe was supported by the MIAUCE European Project (IST-5-0033715-STP).

REFERENCES

1. R. Taagepera, K. Yow, and R. Cipolla, "Feature-based human face detection," *Image and Vision Computing* **15**(9), pp. 713–735, 1997.
2. A. Colmenarez, B. Frey, and T. Huang, "Detection and tracking of faces and facial features," in *ICIP*, pp. 657–661, 1999.
3. H. Graf, E. Casotto, and T. Ezzat, "Face analysis for synthesis of photorealistic talking heads," in *Face & Gesture*, pp. 189–194, 2000.
4. M. Reinders, R. Koch, and J. Gerbrands, "Locating facial features in image sequences using neural networks," in *Face & Gesture*, pp. 230–235, 1997.
5. M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *IJCV* **1**(4), pp. 321–331, 1987.
6. T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models," *CVIU* **61**(1), pp. 38–59, 1995.
7. T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," *PAMI* **23**(6), pp. 681–685, 2001.
8. S. Sclaroff and J. Isidoro, "Active blobs," in *ICCV*, 1998.
9. M. Jones and T. Poggio, "Multidimensional morphable models," in *ICCV*, pp. 683–688, 1998.
10. P. Viola and M. Jones, "Robust real-time face detection," *IJCV* **57**(2), pp. 137–154, 2004.
11. R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in *ICIP*, **1**, pp. 900–903, 2002.
12. C. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *ICCV*, pp. 555–562, 1998.
13. R. Schapire and Y. Freund, "Experiments with a new boosting algorithm," in *ICML*, pp. 148–156, 1996.
14. R. Schapire, "The strenght of weak learnability," *Machine Learning* **5**(1), pp. 197–227, 1990.
15. L. Valiant, "A theory of the learnable," *Communications of the ACM* **27**, pp. 1134–1142, 1984.
16. H. Hotelling, "Some new methods in matrix calculation," *Ann. Math. Statist.* **14**(1), pp. 1–34, 1943.
17. A. Webb, *Statistical Pattern Recognition*, Wiley, 2002.
18. M. B. Stegmann, "Active appearance models: Theory, extensions and cases," Master's thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2000.
19. I. Matthews and S. Baker, "Active appearance models revisited," *IJCV* **60**(2), pp. 135–164, 2004.
20. T. Cootes and P. Kittipanya-ngam, "Comparing variations on the active appearance model algorithm," in *BMVC*, pp. 837–846., 2002.
21. I. Cohen, N. Sebe, A. Garg, L. Chen, and T.S.Huang, "Facial expression recognition from video sequences: Temporal and static modeling," *CVIU* **91**, pp. 160–187, 2003.
22. A.Azcarate, F. Hageloh, K. van de Sande, and R. Valenti, "Automatic facial emotion recognition," tech. rep., University of Amsterdam, 2005.
23. H. Tao and T. Huang, "Connected vibrations: A modal analysis approach to non-rigid motion tracking," in *CVPR*, pp. 735–740, 1998.
24. T. Kanade, J. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," in *Face & Gesture*, pp. 46–53, 2000.
25. M. B. Stegmann, "The 'AAM-API'," 2003.