[5] L. Lee, R. Romano, and G. Stein, "Monitoring activities from multiple video streams: Establishing a common coordinate frame," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, pp. 758–767, 2000.
[6] P. Spagnolo, T. D'Orazio, M. Leo, and A. Distante, "Advances in background updating and shadow removing for motion detection algorithms," in *Proc. CAIP*, 2005, pp. 398–406.
[7] Y. Benezeth, B. Emile, H. Laurent, and C. Rosenberger, "A real time human detection system based on far infrared vision," in *Proc. CIPS*, 2008, pp. 76–84.
[8] D. Greenhill, J. Renno, J. Orwell, and G. A. Jones, "Learning the semantic landscape: Embedding scene knowledge in object tracking," *Real Time Imag.*, vol. 11, no. 3, pp. 186–203, 2005.
[9] B. Hu, C. Brown, and R. Nelson, Multiple-View 3-D Reconstruction Using a Mirror, Tech. Rep., Comput. Sci. Dept., Univ. Rochester, Rochester, NY, 2005.
[10] Z. Szlávik, L. Havasi, and T. Szirányi, "Video camera registration using accumulated co-motion maps," *ISPRS J. Photogramm. Remote Sens.*, vol. 61, no. 1, pp. 298–306, 2007.
[11] L. Havasi and T. Szirányi, "Estimation of vanishing point in camera-mirror scenes using video," *Opt. Lett.*, pp. 1411–1413, 2006.
[12] L. Havasi and T. Szirányi, "Use of motion statistics for vanishing point estimation in camera-mirror scenes," in *Proc. Int. Conf. Image Process.*, 2006, pp. 2993–2996.
[13] Z. Szlávik, L. Havasi, and T. Szirányi, "Geometrical scene analysis using co-motion statistics," in *Proc. ACIVS*, 2007, pp. 968–979.
[14] S. Keren, I. Shimshoni, and A. Tal, "Placing three-dimensional models in an uncalibrated single image of an architectural scene," in *Proc. ACM Symp. Virtual Reality Software and Technology*, 2002, pp. 186–193.
[15] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts and shadows in video streams," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, pp. 1337–1342, 2003.
[16] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, pp. 747–757, 2000.
[17] Cs. Benedek and T. Sziranyi, "Bayesian foreground and shadow detection in uncertain frame rate surveillance videos'," *IEEE Trans. Image Process.*, vol. 17, no. 4, pp. 608–621, Apr. 2008.
[18] G. Xu and Z. Zhang, *Epipolar Geometry in Stereo, Motion and Object Recognition*. Norwell, MA: Kluwer, 1996.
[19] R. A. Johnson and D. W. Wichern, *Applied Multivariate Statistical Analysis*. Upper Saddle River, NJ: Prentice Hall, 2002.
[20] F. Pernkopf and D. Bouchaffra, "Genetic-based EM algorithm for learning Gaussian mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, pp. 1344–1348, 2005.
[21] W. Feller, "Laws of large numbers," *An Introduction to Probability Theory and Its Applications*, vol. 1, pp. 228–247, 1968.
[22] V. Nguyen, A. Martinelli, N. Tomatis, and R. Siegwart, "A comparison of line extraction algorithms using 2D laser rangefinder for indoor mobile robotics," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2005, pp. 1929–1934.
[23] H. Zabrodsky and D. Weinshall, "Utilizing symmetry in the reconstruction of 3-dimensional shape from noisy images," in *Proc. ECCV*, 1994, pp. 403–410.
[24] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the Nelder-Mead simplex method in low dimensions," *SIAM J. Optim.*, vol. 9, pp. 112–147, 1998.
[25] O. Kallenberg, *Foundations of Modern Probability*. New York: Springer-Verlag, 1997.
[26] Z. Szlavik and T. Sziranyi, "Bayesian estimation of common areas in multi-camera systems," in *Proc. Int. Conf. Image Process.*, 2006, pp. 1045–1048.
[27] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Trans. Robot. Autom.*, vol. 7, pp. 278–288, 1991.
[28] N. Kiryati and A. M. Bruckstein, "Heteroscedastic hough transform (HtHT): An effective method for robust line fitting in the 'errors in the variables' problem," *Comput. Vis. Image Understand.*, vol. 78, pp. 69–83, 2000.
[29] A. Webb, *Statistical Pattern Analysis*. New York: Wiley, 2004.
[30] Cs. Benedek, L. Havasi, Z. Szlávik, and T. Szirányi, "Motion-based flexible camera registration," in *Proc. IEEE AVSS*, 2005, pp. 439–444.
[31] D. Hall, J. Nascimento, P. Ribeiro, E. Andrade, P. Moreno, S. Pesnel, T. List, R. Emonet, R. B. Fisher, J. Santos Victor, and J. L. Crowley, "Comparison of target detection algorithms using adaptive background models," in *Proc. PETS*, 2005, pp. 113–120.

On Color Texture Normalization for Active Appearance Models

Mircea C. Ionita, Peter Corcoran, and Vasile Buzuloiu

***Abstract*—The extension of the standard grayscale *active appearance model* (AAM) techniques to color images is investigated. Prior work in this field has mainly focused on RGB color models which did not demonstrate noticeable benefits over grayscale models from the point of view of convergence accuracy. We improve on previous work by normalizing the color texture vector separately for intensity and chromaticity components. Where an appropriate color space is chosen, we demonstrate improvements in convergence accuracy as well as image synthesization quality for AAMs. Optimal results are achieved when a color space in which the image channels are strongly decorrelated is chosen. Our best results are achieved using the I1I2I3 color space, originally proposed by Ohta.**

***Index Terms*—Active appearance model (AAM), color spaces, PCA.**

I. Introduction

The AAM techniques were first described by Edwards *et al.* [1]. They have been extensively used in applications such as face tracking and analysis and interpretation of medical images.

Originally designed for grayscale images, AAMs have been later extended to color images. Edwards *et al.* [2] first proposed a color AAM based on the RGB color space. This approach involves constructing a color texture vector by merging the concatenated values of each color channel. However, their results did not indicate that benefits in accuracy could be achieved from the additional chromaticity data which were made available. Furthermore, the extra computation required to process these data suggested that color-based AAMs did not justify their use over conventional grayscale AAMs from the point of view of fitting accuracy.

Stegmann *et al.* [3] proposed a value, hue, edge map (VHE) representation of image structure. They used a transformation to HSV (hue, saturation, and value) color space from where they retained only the hue and value (intensity) components; they added to these an edge map component, obtained using numeric differential operators. A color texture vector was created as in [2], using VHE components instead of the RGB ones. In their experiments they compared the convergence accuracy of the VHE model with the grayscale and RGB implementations. Here they obtained unexpected results indicating that the RGB model (as proposed in [2]) was slightly less accurate than the grayscale model. The VHE model outperformed both grayscale and RGB models but only by a modest amount. Yet, some applicability of the VHE model has been shown for the case of directional lighting changes.

We study in this correspondence the way in which a more appropriate extension of AAM techniques to color images could be achieved. Our work has focused on color spaces other than RGB because intensity

Manuscript received May 19, 2007; revised February 03, 2009. First published April 17, 2009; current version published May 13, 2009. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Gabriel Marcu.

M. C. Ionita and P. Corcoran are with the Electronic Engineering Department, National University of Ireland, Galway, Ireland (e-mail: mc.ionita@gmail.com; peter.corcoran@nuigalway.ie).

V. Buzuloiu is with the University Politehnica of Bucharest, Applied Electronics and Information Engineering Department, Bucharest, Romania (e-mail: buzuloiu@alpha.imag.pub.ro).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2009.2017163

and chromaticity information are strongly mixed in each of the RGB channels. By employing color spaces where there is a better separation of the chromaticity and intensity information, we are able to distinguish between intensity-dependent and chromaticity-dependent aspects of a color AAM. This allows for a new approach on normalizing color texture vectors by performing a set of independent normalization operations on the sub-vectors corresponding to each color channel. This approach has further enabled us to train more accurate models.

In Section II, we briefly summarize the basic AAM algorithm for grayscale images. We also present in more detail the grayscale texture normalization method as well as its common extension to color data. We further analyze in Section III, the possibility of generating color texture vectors that retain more specific content by applying the texture normalization separately to each component of the color space. We show that the I1I2I3 color space, which decorrelates the color channels, is well suited for this purpose. In Section IV, we show the experimental results and provide a detailed comparison between the standard grayscale model, the common RGB extension, and the proposed models. We also demonstrate that the proposed method for color texture normalization is not suited to be used directly with the common RGB color space. Finally, in Section V, we present the conclusions of our work.

II. Background

In what follows, we frequently use the term *texture*, which, in AAM terminology, refers to the set of pixel intensities across the modeled object, possibly subsequent to a suitable normalization.

A. Overview of the Grayscale AAM

The image properties modeled by AAMs are shape and texture. The parameters of the model are estimated from the initial scene giving the parametric image. In order to build a statistical model of the appearance of an object, *principal components analysis* (PCA) is used generate i) a shape model, ii) a texture model, and iii) a combined model of appearance. A training data set contains image examples annotated with a fixed set of landmark points.

The sets of 2-D coordinates of the landmark points define the shapes inside the image frame. If N is the number of training examples, each shape is represented as a vector $\mathbf{s} = (x_1, x_2, \ldots, x_L, y_1, y_2, \ldots, y_L)^T$ of concatenated x and y coordinates, where L is the number of landmark points. The shapes are aligned using generalized Procrustes analysis [4], a technique used for normalizing the shapes by removing 2-D translation, rotation and scale differences between them. PCA is then applied to the set of aligned shape vectors and shape variability is linearly modeled as a base (mean) shape plus a linear combination of shape eigenvectors

$$\mathbf{s}_m = \bar{\mathbf{s}} + \Phi_s \mathbf{b}_s \tag{1}$$

where $\mathbf{s}_m$ represents a modeled shape, $\bar{\mathbf{s}}$ is the mean shape, $\Phi_s = (\phi_{s_1}|\phi_{s_2}|\cdots|\phi_{s_p})$ is a matrix whose columns represent the first p $(p < N)$ eigenvectors; p is chosen such that a certain percentage of the total variance is retained; finally, $\mathbf{b}_s$ are the parameters of the shape model.

For building the texture model, the set of texture vector examples is acquired using a reference shape, e.g., the mean shape. A piecewise affine warping based on a triangulated mesh of the reference shape is commonly employed to map the image examples into a shape-normalized representation, performing, thus, an image registration process. The texture vectors are formed by scanning the pixel values across the reference shape as $\mathbf{t}_{im} = (t_{im_1}, t_{im_2}, \ldots, t_{im_P})^T$, where P is the number of texture samples. A photometric normalization is applied on the texture vectors, as it will be detailed in Section II-B, in order to reduce the effect of global illumination variations and to align them as closely as possible to the normalized mean. PCA is again used to generate the texture model in (2)

$$\mathbf{t}_m = \bar{\mathbf{t}} + \Phi_t \mathbf{b}_t \tag{2}$$

where $\mathbf{t}_m$ is a synthesized texture, $\bar{\mathbf{t}}$ is the mean of the training data, $\Phi_t = (\phi_{t_1}|\phi_{t_2}|\cdots|\phi_{t_q})$ is the matrix of q $(q < N)$ column eigenvectors, which retains a certain percentage of total texture variation; $\mathbf{b}_t$ are the texture parameters.

A vector $\mathbf{c}$ is further formed by concatenating the shape and texture parameters that optimally describe each of the training examples, $\mathbf{c} = \begin{bmatrix} \mathbf{W}_s \mathbf{b}_s^{(\mathrm{opt})} \\ \mathbf{b}_t^{(\mathrm{opt})} \end{bmatrix}$; $\mathbf{W}_s$ is a diagonal matrix of weights that compensates for the differences in units between the shape and texture parameters.

A model for which the concatenated shape and texture parameters $\mathbf{c}$ are directly used to describe the appearance variability is called an *independent model of appearance*. A more compact model may be obtained by taking into account the correlation between shape and texture. Thus, a third PCA is applied on the set of vectors $\mathbf{c}$, resulting in a *combined model of appearance*

$$\mathbf{c}_m = \Phi_c \mathbf{b}_c \tag{3}$$

where Φ_c is the matrix of retained eigenvectors and $\mathbf{b}_c$ represents the set of parameters that provide combined control of shape and texture variations.

During the optimization stage of an AAM, i.e., fitting the model to a query image, the parameters to be found are $\mathbf{p} = \begin{bmatrix} \mathbf{g}_s \\ \mathbf{b}_c \end{bmatrix}$, where $\mathbf{g}_s$ are the shape $2-D$ translation (t_x, t_y), rotation $(\theta \in [0, 2\pi))$ and (isotropic) scaling $(s > 0)$ parameters inside the image frame, and $\mathbf{b}_c$ are the combined model parameters. The statistical model is linear in both shape and texture or appearance. Fitting the model to a new image example represents though a nonlinear optimization problem. The fitting algorithm is based on minimizing the error between the query image and the model-synthesized image. The error is evaluated in the coordinate frame of the model, i.e., in the normalized texture reference frame, rather than in the coordinate frame of the image. This choice allows for a fast approximation of a gradient descent optimization algorithm to be used, which will be described as follows:

$$\mathbf{r}(\mathbf{p}) = \mathbf{t} - \mathbf{t}_m \tag{4}$$

while $\|\mathbf{r}(\mathbf{p})\|^2$ gives the reconstruction error, with $\|\cdot\|$ marking the Euclidean norm.

A first order Taylor extension of $\mathbf{r}(\mathbf{p})$ is given by

$$\mathbf{r}(\mathbf{p} + \delta\mathbf{p}) \simeq \mathbf{r}(\mathbf{p}) + \frac{\partial \mathbf{r}}{\partial \mathbf{p}} \delta\mathbf{p}. \tag{5}$$

$\delta\mathbf{p}$ should be chosen so that to minimize $\|\mathbf{r}(\mathbf{p} + \delta\mathbf{p})\|^2$. It follows that

$$\frac{\partial \mathbf{r}}{\partial \mathbf{p}} \delta\mathbf{p} = -\mathbf{r}(\mathbf{p}). \tag{6}$$

Normally, the gradient matrix $(\partial\mathbf{r})/(\partial\mathbf{p})$ should be recomputed at each iteration. Yet, as the error is estimated in a normalized texture frame, this gradient matrix is considered fixed. This enables it to be precomputed from the training data set. Given a training image, each parameter in $\mathbf{p}$ is systematically displaced from its known optimal value producing a set of normalized texture differences. The resulted matrices are then averaged over several displacement amounts and over several training images.

The update direction of the model parameters $\mathbf{p}$ is then given by

$$\delta\mathbf{p} = -\mathbf{R}\mathbf{r}(\mathbf{p}) \tag{7}$$

where $\mathbf{R} = ((\partial \mathbf{r})/(\partial \mathbf{p})^{*T}(\partial \mathbf{r})/(\partial \mathbf{p})^{*})^{-1}(\partial \mathbf{r})/(\partial \mathbf{p})^{*T}$ is the pseudo-inverse of the predetermined gradient matrix $((\partial \mathbf{r})/(\partial \mathbf{p}))^{*}$. The parameters $\mathbf{p}$ are updated iteratively until the error can no longer be reduced and convergence is declared.

B. Texture Normalization Stage

As noted also by Batur *et al.* [5], and confirmed by our experiments, this stage is critical during the optimization process, providing the best chance for predicting the correct update direction of the parameter vector.

Texture normalization is realized by applying a scaling α, and an offset β to the texture vector $\mathbf{t}_{im}$

$$\mathbf{t} = \frac{\mathbf{t}_{im} - \beta \mathbf{1}}{\alpha} \tag{8}$$

where $\mathbf{1}$ is a vector of ones. The values for α and β are chosen so that to best match the current vector to the mean vector of the normalized data. In practice, the mean normalized texture vector is offset and scaled to have zero-mean and unit-variance. If $(1/N)\sum_{i=1}^{N}\mathbf{t}_i$ is the mean vector of the normalized texture data, let $\bar{\mathbf{t}}_{zm,uv}$ be its zero-mean and unit-variance correspondent. Then, the values for α and β required to normalize a texture vector $\mathbf{t}_{im}$, according to (8), are given by

$$\alpha = \mathbf{t}_{im}^{T}\bar{\mathbf{t}}_{zm,uv} \tag{9}$$

$$\beta = \frac{\mathbf{t}_{im}^{T}\mathbf{1}}{P}. \tag{10}$$

Obtaining the mean of the normalized data is, thus, a recursive process. A stable solution can be found by using one texture vector as the first estimate of the mean. Each texture vector is then aligned to zero mean and unit variance mean vector as described in (8)–(10), re-estimating the mean and iteratively repeating these steps until convergence is achieved.

C. Color AAM Extensions. The Global Color Texture Normalization

Given the fact that the existing methods for extending the AAM techniques to color images showed rather unsatisfactory results from the point of view of convergence accuracy over the grayscale model, we decided to have a more thorough look on how the color AAM extensions could be realized. Before investigating this further, we present in this section the main characteristics of the common AAM extension to color images.

RGB is by far the most widely used color space in digital imagery [6]. The extension proposed by Edwards *et al.* [2] is, thus, realized by using an extended texture vector given by concatenated RGB components as in (11)

$$\begin{aligned}\mathbf{t}_{im}^{\mathrm{RGB}} = \Big(&t_{im_1}^{R}, t_{im_2}^{R}, \ldots, t_{im_{P_c}}^{R}\\ &t_{im_1}^{G}, t_{im_2}^{G}, \ldots, t_{im_{P_c}}^{G}\\ &t_{im_1}^{B}, t_{im_2}^{B}, \ldots, t_{im_{P_c}}^{B}\Big)^{T}\end{aligned} \tag{11}$$

where P_c is the number of color texture samples. In order to reduce the effects of global lighting variations, the same normalization method as for the grayscale model, described in Section II-A, is applied on the full color texture vectors. The main purpose of the texture normalization stage is, similar with the grayscale model, to facilitate the use of a fixed gradient matrix during the parameter optimization stage and, consequently, to enable the application of the fast AAM fitting algorithm.

III. Texture Normalization on Channel Sub-Vectors

When a typical multichannel image is represented in a conventional color space such as RGB, there are correlations between its channels. Channel decorrelation refers to the reduction of the cross-correlation between the components of a color image in a certain color space representation. In particular, the RGB color space presents very high interchannel correlations. For natural images, the cross-correlation coefficient between B and R channels is ~ 0.78, between R and G channels is ~ 0.98, and between G and B channels is ~ 0.94 [7]. This implies that, in order to process the appearance of a set of pixels in a consistent way, one must process the color channels as a whole and not independently.

This observation suggests an explanation as to why previous authors [2] obtained rater unsatisfactory results when being compelled to treat all three color components as a single entity. Indeed, if one attempts to normalize individual image channels within a highly correlated color space, such as RGB, the resulting model fails to improve on the model on a model with global texture normalization, and, most probably, it yields much poorer results. However, once we realize that each image channel can be individually normalized once it is substantially decorrelated from the other image channels, then a more suitable color texture normalization can be designed.

We remark that there are several color spaces which were specifically designed to separate color information into intensity and chromaticity components. However, such a separation still does not necessarily guarantee they have a good generality from the point of view of offering strong image channel decorrelation. A color space which meets our requirements has particularly been designed so that to meet exactly this requirement as to offer a good interchannel decorrelation for a wide range of natural images. This color space is described in the following section.

A. Efficiently Decorrelated Color Space

An interesting color space is I1I2I3, proposed by Ohta *et al.* [8], which realizes an efficient minimization of the interchannel correlations (decorrelation of the RGB components) for natural images. The conversion from RGB to I1I2I3 is given by the simple linear transformation in (12)

$$I_1 = \frac{R+G+B}{3} \tag{12a}$$

$$I_2 = \frac{R-B}{2} \tag{12b}$$

$$I_3 = \frac{2G-R-B}{4}. \tag{12c}$$

I1 stands as the achromatic (intensity) component, while I2 and I3 are the chromatic components. We remark that the simple numeric transformation from RGB to I1I2I3 enables for an efficient transformation of data sets between these two color spaces.

I1I2I3 was designed as an approximation for the Karhunen-Loève transform (KLT) of the RGB data to be used for region segmentation on color images. The KLT is optimal in terms of energy compaction and mean squared error minimization for a truncated representation. Note that KLT is very similar to PCA. In a geometric interpretation, KLT can be viewed as a rotation of the coordinate system, while for PCA the rotation of the coordinate system is preceded by a shift of the origin to the mean point [9]. By applying KLT to a color image, it creates image basis vectors which are orthogonal, and it, thus, achieves complete decorrelation of the image channels. As the fixed transformation to I1I2I3 represents a good approximation of the KLT for a large set of natural images, the resulting color channels present a statistically best degree of decorrelation. The I1I2I3 color space can, thus, become useful for applying color image processing operations independently to each image channel.

In the previous work of Ohta *et al.* [8], the discriminating power of 109 linear combinations of R, G, and B were tested on eight different color scenes. The selected linear combinations were gathered such that they could successfully be used for segmenting important (large area) regions of an image, based on a histogram threshold. It was found that

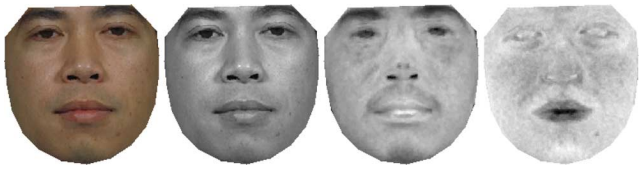

Fig. 1. Color texture example and its decomposition into I1, I2, and I3 representation of Ohta space, respectively.

83 of the linear combinations had all positive weights, corresponding mainly to an intensity component which is best approximated by I1; another 22 showed opposite signs for the weights of R and B, representing the difference between the R and B components which is are best approximated by I2; finally the remaining 4 linear combinations could be approximated by I3. Thus, it was shown that the I1, I2, and I3 components in (12) are performing the best when discriminating between different regions and that they are significant in this order [8].

B. I1I2I3-Based Color AAM

The advantage of the I1I2I3 representation is that the texture alignment method used for grayscale models can now be applied independently on each channel. By considering the band sub-vectors individually, the alignment method described by (8)–(10) can be independently applied to each sub-vector. A color texture example and its decomposition in the I1I2I3 color space are shown in Fig. 1.

The color texture vector is then rebuilt using the separately normalized components into the full normalized texture vector. In this way, the effect of global lighting variation is reduced due to the normalization on the first channel which corresponds to an intensity component. Furthermore, the effect of some global chromaticity variation is reduced due to the normalization operations applied on the other two channels corresponding to the chromatic components. Thus, the AAM search algorithm becomes more robust to variations in lighting levels and color distributions.

This also addresses a further aspect of AAMs which is their dependency on initial training set. Although this effect can sometimes be seen as a feature, it very much restricts its range of applicability to the particular (naturally constrained) environment of the training data set. Color AAMs are in particular much more sensitive to variations of the image acquisition or environmental characteristics, e.g., color illuminant or color balance. For example, if an annotated training set is prepared using a digital camera with a color gamut with extra emphasis on *redness* (some manufacturers do customize their cameras according to market requirements), then the RGB-based AAM will perform poorly on images captured with a camera which has a normal color balance. A model, build using multichannel normalization will be noticeably more tolerant to such variations in image color balance.

As remarked also in [3], the common linear normalization applied on concatenated RGB components is less than optimal. The proposed I1I2I3-based model uses a more powerful normalization method which yields more accurate results, as will be shown in the next section.

Moreover, by employing the I1I2I3 color space, a more efficient compaction of the color texture data is achieved. As the texture sub-vectors correspond to I1, I2, and I3 channels, which are significant in the order of $\sim 76\%$, $\sim 20\%$, and $\sim 4\%$, one can retain a significant amount of the useful information just from the first two texture sub-vectors. Thus, a reduced I1I2 model can be designed with the performance comparable to a full I1I2I3 model in terms of final convergence accuracy. It is, thus, expected that, combined with the proposed normalization method for separate texture sub-vectors, a reduced I1I2 model would preserve much of the full I1I2I3 model characteristics, yet reducing the computational costs.

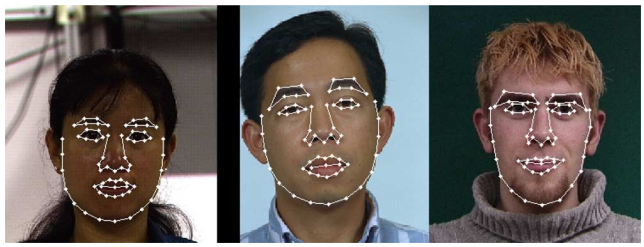

Fig. 2. Annotated image examples from CMU PIE, FERET, and IMM databases, respectively.

IV. Experiments

We analyze the performance of different texture representation/normalization methods in the context of AAM for face modeling. The performance is evaluated in terms of convergence accuracy of the models. The *point-to-curve* (Pt-Crv) boundary error is measured between the optimized shape points in the image frame and the boundary of the ground-truth annotations; it is given by $(1/L)\sum_{i=1}^{L}\min_t \|s_i - r_i(t)\|$, where the border is modeled as a linear spline $r(t) = (r_x(t), r_y(t))$. The texture reconstruction error is measured between the query image and modeled image after texture de-normalization. In order to have a qualitative differentiation between the synthesized images which should be in accordance with the human perception, this error is evaluated as an Euclidean distance in CIELAB color space, giving, thus, the *perceptual texture error* (PTE).

The following model implementations have been considered: standard grayscale model—added for comparative reasons, RGB with no normalization (RGB-none)—added so that to acknowledge the importance the normalization process, RGB with global normalization (RGB-G), RGB with normalization per channel sub-vectors (RGB-Ch), I1I2I3 with normalization per channel sub-vectors (I1I2I3-Ch), and reduced I1I2 representation with separate channel normalization (I1I2-Ch).

We mention that during our previous tests we have also considered the CIELAB color space representation, yet the results have not been eloquent. A limitation of the CIELAB representation has been observed as it could not be efficiently used with texture normalization on separate channel sub-vectors. Although results were improved over the common RGB implementation for many tested databases, in particular for unseen databases, they lacked consistency.

Three standard face image databases have been considered, namely the CMU PIE [10], color FERET [11], and the IMM [12]. We used for our tests only face images with full frontal pose, no glasses, and diffuse lighting. A total of 66 images have been chosen from the whole PIE database, 78 images from FERET database and all 37 images of IMM database. All images have been manually annotated using 65 landmark points as shown in Fig. 2. For CMU PIE as well as for FERET sets, the first 40 images have been used during training; for IMM database the first 20 images have been used for training. The three training, or *seen* data sets will also be referred to in the following as *db1*, *db2*, and *db3*, respectively. The remaining images corresponding to each data set represent the *unseen* set.

To be sure that the measured difference between the different model implementations can only be attributed to differences in texture normalization/modeling, all models have been trained using all shape examples from the three training data sets; thus, the shape model is unique across all model implementations.

All models are initialized using mean shape and mean texture and setting an offset from the optimal pose of the ground-truth shape of -20 and -10 pixels on the x and y coordinates, respectively. Convergence is declared successful when Pt-Crv is less than ten pixels. We

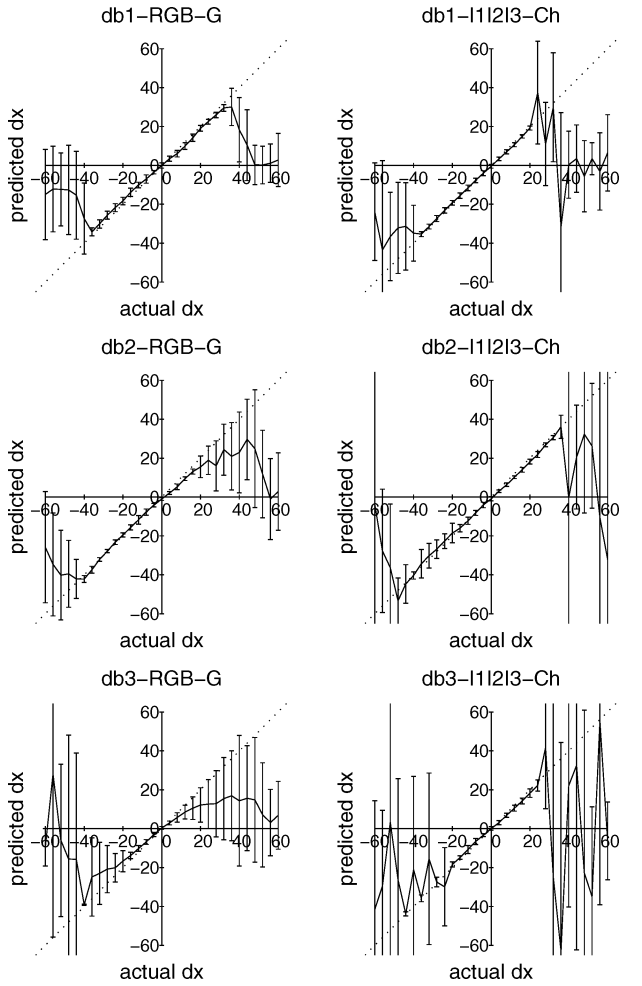

Fig. 3. Actual versus predicted dx displacements for RGB-G and I1I2I3-Ch model implementations on examples from the three databases considered.

TABLE I
CONVERGENCE RESULTS ON *SEEN* IMAGES

Model	Success Rate [%]	Pt-Crv (Mean/Std/Median)			PTE (Mean/Std/Median)		
-	-	15.00	6.75	15.00	-		
db1-Grayscale*	95.00	4.31	2.17	3.58	3.46	2.13	2.54
db1-RGB-none	97.50	2.85	0.96	2.69	3.89	1.59	3.33
db1-RGB-G	97.50	3.48	1.82	2.76	4.37	1.96	3.54
db1-RGB-Ch	92.50	4.11	2.05	3.70	4.90	2.00	4.21
db1-I1I2I3-Ch	**100**	**2.62**	**0.85**	**2.31**	**3.37**	**0.70**	**3.18**
db1-I1I2-Ch	**100**	2.88	1.06	2.52	5.60	1.18	5.41
db2-Grayscale*	92.50	3.60	1.37	3.59	3.22	1.96	2.74
db2-RGB-none	**97.50**	2.85	1.32	2.31	**3.23**	**1.58**	**2.82**
db2-RGB-G	**97.50**	3.05	1.42	2.60	3.35	1.02	3.19
db2-RGB-Ch	77.50	3.50	1.39	3.13	6.11	7.60	3.44
db2-I1I2I3-Ch	92.50	**2.69**	**1.07**	**2.43**	3.36	1.83	2.85
db2-I1I2-Ch	92.50	3.23	1.46	2.89	4.67	1.35	4.30
db3-Grayscale*	100	2.92	1.52	2.30	2.56	0.80	2.23
db3-RGB-none	**100**	2.28	0.71	2.12	3.24	0.56	3.12
db3-RGB-G	**100**	2.45	0.98	2.13	3.27	0.75	3.10
db3-RGB-Ch	**100**	2.82	1.29	2.47	3.29	0.86	3.06
db3-I1I2I3-Ch	**100**	**2.01**	**0.35**	**2.00**	**2.81**	**0.42**	**2.77**
db3-I1I2-Ch	**100**	2.42	0.96	2.07	13.25	2.40	13.49

TABLE II
CONVERGENCE RESULTS ON *UNSEEN* IMAGES FROM THE *SEEN* DATABASE

Model	Success Rate [%]	Pt-Crv (Mean/Std/Median)			PTE (Mean/Std/Median)		
db1-Grayscale*	100	3.75	1.45	3.28	3.23	1.10	3.01
db1-RGB-none	**100**	3.19	1.55	2.65	4.89	1.14	4.80
db1-RGB-G	96.15	2.91	0.85	2.79	5.05	1.27	4.90
db1-RGB-Ch	**100**	4.12	1.69	3.75	5.34	1.41	5.02
db1-I1I2I3-Ch	**100**	2.72	0.77	2.73	**4.69**	**0.94**	**4.43**
db1-I1I2-Ch	**100**	**2.57**	**0.78**	**2.37**	5.85	1.19	5.97
db2-Grayscale*	89.47	3.56	1.21	3.14	3.36	0.89	3.37
db2-RGB-none	**97.37**	3.28	1.41	2.97	4.51	1.83	4.15
db2-RGB-G	**97.37**	3.13	0.96	2.89	4.38	0.78	4.26
db2-RGB-Ch	71.05	3.61	1.31	3.01	6.60	6.78	4.22
db2-I1I2I3-Ch	**97.37**	**2.70**	**0.66**	**2.77**	**3.99**	**0.63**	**3.97**
db2-I1I2-Ch	**97.37**	2.87	0.85	2.81	4.77	0.84	4.69
db3-Grayscale*	100	2.61	0.86	2.34	2.84	0.57	2.64
db3-RGB-none	**100**	2.66	0.85	2.61	5.06	1.16	4.98
db3-RGB-G	**100**	2.77	1.12	2.58	4.63	1.08	4.40
db3-RGB-Ch	**100**	2.76	1.00	2.67	**3.94**	**0.76**	**3.70**
db3-I1I2I3-Ch	**100**	**2.64**	**0.92**	**2.60**	4.27	0.87	4.22
db3-I1I2-Ch	**100**	**2.64**	**0.99**	**2.55**	12.88	2.93	12.27

note that, due to the characteristics of the face, a particular model can successfully converge from larger displacements on the horizontal axis than the vertical axis, as observed from previous experiments; hence, the choice for our initial displacements.

To provide an indication on the convergence range differences between the models, we also studied convergence accuracy over a wider range of initial displacements on the x coordinate. The tests have been performed on a mixed data set containing images from all three considered databases for the RGB-G and I1I2I3-Ch models. Fig. 3 shows diagrams of actual versus predicted displacements on a range of $[-60, 60]$ pixels from the optimum position. The predicted displacements are averaged with respect to all images in the analyzed data set; the vertical segments represent one unit of standard deviation of each predicted displacement over the considered data set. We note that for these tests the unsuccessful convergence results have also been considered in order to determine which are the extreme dx values from which a particular model has still a high chance to convergence. It can be seen that the convergence range, represented by the linear part of the diagram, is rather similar for both model implementations for $db1$-trained and $db2$-trained models; I1I2I3-Ch implementation presents a more accurate convergence range over the RGB-G counterpart for the $db3$-trained models.

Four sets of experiments have been performed in order to analyze i) the capacity of the models to *memorize* the set of learning examples, ii) their capability of generalizing to new examples from the training database, as well as iii) their capability of generalizing to new, *unseen* databases; finally, iv) we want to acknowledge the degree in which the PCA texture modeling is actually responsible for the fitting accuracy in a particular implementation—for this test, the entire texture variation has been removed. The full sets of results are summarized in Tables I–IV. Both Pt-Crv and PTE measures are shown in terms of their mean, standard deviation (Std) and median values over the tested data sets. We remark that the outliers, i.e., images that recorded unsuccessful convergence, have this time been removed from the Pt-Crv and PTE statistics.

The convergence tests for the *seen* data sets in Table I show that the RGB model with no texture normalization performs actually better than the RGB model with global normalization, which shows the inefficiency of this type of normalization.

Consistent results have been obtained for I1I2I3-Ch and I1I2-Ch models, where the convergence accuracy is improved over the RGB-G implementation for all studied data sets and experiments, both in terms of Pt-Crv and PTE. It can also be noticed that the proposed normalization cannot be successfully used with a RGB color space representation.

The results for the case of retaining no texture variation, i.e., when only the mean texture vector information is preserved, clearly favored the proposed per-channel I1I2I3 texture normalization method. It can be observed, by comparing the figures in Tables III and IV, that the importance of the PCA-based texture modeling decreases with

TABLE III
CONVERGENCE RESULTS ON *UNSEEN* DATABASES

Model	Success Rate [%]	Pt-Crv (Mean/Std/Median)			PTE (Mean/Std/Median)		
db1-Grayscale*	92.17	5.10	1.66	4.90	4.28	1.03	4.21
db1-RGB-none	**99.13**	4.94	1.37	4.82	10.09	1.58	9.93
db1-RGB-G	98.26	4.98	1.44	4.65	7.49	1.98	7.02
db1-RGB-Ch	87.83	5.32	1.65	5.08	6.33	1.40	5.95
db1-I1I2I3-Ch	**99.13**	**3.60**	**1.32**	**3.32**	**5.10**	**1.01**	**4.85**
db1-I1I2-Ch	**99.13**	4.25	1.65	3.79	8.26	4.11	6.10
db2-Grayscale*	75.73	4.17	1.44	3.67	5.12	4.24	4.03
db2-RGB-none	84.47	4.02	1.40	3.69	12.43	3.43	12.41
db2-RGB-G	**94.17**	3.74	1.45	3.23	9.04	1.83	8.97
db2-RGB-Ch	62.14	4.01	1.60	3.46	7.70	4.26	6.06
db2-I1I2I3-Ch	88.35	**3.31**	**1.26**	**2.98**	**6.16**	**2.28**	**5.73**
db2-I1I2-Ch	87.38	3.60	1.55	3.04	10.00	3.41	8.94
db3-Grayscale*	63.89	4.85	2.12	4.26	4.90	3.44	3.98
db3-RGB-none	72.22	4.44	1.79	3.99	14.23	4.79	13.34
db3-RGB-G	65.28	4.55	2.03	4.01	9.68	2.81	9.27
db3-RGB-Ch	59.72	5.02	2.04	4.26	7.16	4.91	5.74
db3-I1I2I3-Ch	**86.81**	**3.53**	**1.49**	**3.15**	**6.04**	**2.56**	**5.20**
db3-I1I2-Ch	**86.81**	3.90	1.66	3.41	6.60	1.94	6.30

TABLE IV
CONVERGENCE RESULTS ON *UNSEEN* DATABASES WITH NO TEXTURE VARIATION MODELING

Model	Success Rate [%]	Pt-Crv (Mean/Std/Median)			PTE (Mean/Std/Median)		
db1-Grayscale*	93.04	4.16	1.56	3.86	4.37	0.91	4.17
db1-RGB-none	74.78	6.95	2.05	7.35	24.05	2.10	24.06
db1-RGB-G	**99.13**	4.00	1.50	3.58	7.92	3.33	6.25
db1-RGB-Ch	89.57	4.59	1.79	4.27	6.02	1.49	5.66
db1-I1I2I3-Ch	**99.13**	**3.37**	**1.26**	**3.10**	**5.36**	**1.03**	**5.10**
db1-I1I2-Ch	97.39	4.18	1.55	3.78	8.53	4.03	6.49
db2-Grayscale*	77.67	4.48	1.44	3.98	6.06	5.11	4.93
db2-RGB-none	38.83	4.40	1.76	3.92	19.77	7.26	22.09
db2-RGB-G	80.58	4.75	1.63	4.42	11.41	2.02	11.35
db2-RGB-Ch	65.05	4.44	1.67	3.75	8.39	5.54	6.57
db2-I1I2I3-Ch	**92.23**	**3.36**	**1.13**	**2.98**	**6.68**	**2.04**	**6.20**
db2-I1I2-Ch	89.32	3.84	1.43	3.38	10.64	3.50	9.45
db3-Grayscale*	62.50	4.67	1.96	4.34	5.18	3.53	4.35
db3-RGB-none	42.36	5.01	2.39	4.21	18.98	7.73	20.96
db3-RGB-G	61.11	4.77	2.19	4.20	12.33	3.93	11.06
db3-RGB-Ch	59.03	4.73	2.16	4.21	7.63	6.87	5.59
db3-I1I2I3-Ch	**86.81**	**3.74**	**1.64**	**3.30**	**6.19**	**2.44**	**5.63**
db3-I1I2-Ch	84.03	4.16	1.71	3.82	7.10	2.22	6.66

the level of normalization performed onto the texture vectors. Thus, texture modeling is an essential operation when no normalization is performed, becoming least critical in the case of I1I2I3-Ch or I1I2-Ch implementations.

It is known (see Section III-A) that a great amount of relevant data is actually encapsulated in the I1 and I2 components of the I1I2I3 representation. As expected, the difference between using an AAM derived from a full I1I2I3 color space representation and the one which retains only the first two channels is not significant. Where the speed of convergence is a priority, the reduced I1I2 model may be favored to a full I1I2I3 model due to the lower dimensionality of the overall texture vector and the reduced computational requirements of this two-channel model.

Note that, although some noise amplification effect might be expected when the normalization on channel sub-vectors is employed, we found that this effect is not present. This fact is also reflected in the PTE values (see Tables I–IV), which actually decrease when using the normalization on channel sub-vectors.

V. DISCUSSION AND CONCLUSIONS

Texture normalization was used in the grayscale AAM as an important tool to compensate for global lighting variations between images, as well as to allow and motivate the use of the constant gradient assumption, which further permitted the development of a fast fitting algorithm. When AAM techniques have been extended to color images, the same principle was used on the enlarged texture vector of concatenated RGB components; thus, a rather similar effect of compensating for global intensity variations was obtained. Yet, when color texture is being modeled, one is confronted with a much more complex task, which involves not only intensity variations, but also chrominance variations. Chrominance can have a wide range of variability in digital images, even when considering one object class. It is, thus, much more difficult to model a colored object and it can be easily influenced by external factors like data acquisition devices or changes of environment. Typically, the area of applicability is restrained to a highly constrained environmental setup, in which the intrinsic characteristics of the modeled object can be reliably extracted and interpreted. In the case of face modeling, the resulted model is applicable to images that share similar characteristics, as it is usually the case of a standard database. To a greater extent as for grayscale, the trained color model loses generality when a new color database is considered due to inherent environmental and image acquisition differences. Thus, texture normalization prior to PCA modeling becomes an essential tool to make proper use of the available color information and to increase the generality of a particular model.

We proposed a more powerful color texture normalization technique where each texture sub-vector corresponding to an individual color channel is normalized independently of the other channels. Although this approach cannot be successfully used with the common RGB representation due to the high interchannel correlations, it has been deduced that this is achievable by employing a color space where intensity and chromaticity information are better separated. In particular, it was found that the I1I2I3 color space, which was specifically designed to minimize the correlation coefficients between the color channels, is the best practical choice for this purpose.

By employing the I1I2I3 color space coupled with texture normalization on separate channel sub-vectors, we are able to improve the convergence accuracy and to achieve a more accurate reconstruction of the color image. Note that, by using the proposed I1I2I3 model with texture normalization on separate channel subvectors, the optimization algorithm, which is typically based on a gradient descent approximation, is less susceptible to errors caused by local error function minima. Thus, the algorithm performance is also more robust.

The proposed I1I2I3 color AAM was designed to filter out, by means of a more powerful texture normalization, image content that is difficult or not desirable to be included in the PCA texture modeling. More database-specific color content is filtered out using the normalization performed per each individual channel, while intrinsic general facial content is still retained and modeled within texture PCA. The proposed normalization method reduces as well the importance of the texture modeling stage from the point of view of successful convergence rates. More accurate model convergence, both in terms of shape errors and texture errors, has been demonstrated.

REFERENCES

[1] G. J. Edwards, C. J. Taylor, and T. F. Cootes, "Interpreting face images using active appearance models," in *Proc. 3rd IEEE Int. Conf. Face & Gesture Recognition*, 1998, pp. 300–305.
[2] G. J. Edwards, T. F. Cootes, and C. J. Taylor, "Advances in active appearance models," in *Proc. Int. Conf. Computer Vision*, 1999, pp. 137–142.
[3] M. B. Stegmann and R. Larsen, "Multi-band modelling of appearance," *Image Vis. Comput.*, vol. 21, no. 1, pp. 61–67, Jan. 2003.
[4] C. Goodall, "Procrustes methods in the statistical analysis of shape," *J. Roy. Statist. Soc. B*, vol. 53, no. 2, pp. 285–339, 1991.
[5] A. U. Batur and M. H. Hayes, "Adaptive active appearance models," *IEEE Trans. Image Process.*, vol. 14, no. 11, pp. 1707–1721, Nov. 2005.
[6] G. Sharma and H. J. Trussell, "Digital color imaging," *IEEE Trans. Image Process.*, vol. 6, no. 7, pp. 901–932, Jul. 1997.

[7] M. Tkalčič and J. F. Tasič, "Colour spaces—Perceptual, historical, and applicational background," presented at the IEEE, EUROCON, 2003.
[8] Y. Ohta, T. Kanade, and T. Sakai, "Color information for region segmentation," *Comput. Graph. Image Process.*, no. 13, pp. 222–241, 1980.
[9] J. J. Gerbrands, "On the relationships between SVD, KLT, and PCA," *Pattern Recognit.*, vol. 14, no. 1–6, pp. 375–381, 1981.
[10] T. Sim, S. Baker, and M. Bsat, The CMU Pose, Illumination, and Expression (PIE) Database of Human Faces, Robotics Institute, Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-RI-TR-01-02, Jan. 2001.
[11] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 10, pp. 1090–1104, Oct. 2000.
[12] M. M. Nordstrøm, M. Larsen, J. Sierakowski, and M. B. Stegmann, The IMM Face Database—An Annotated Dataset of 240 Face Images, Informatics and Mathematical Modelling, Technical Univ. Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, May 2004, Tech. Rep.