An Automatic Facial Feature Finding System for Portrait Images

Mark R. Bolin[*] *and Shoupu Chen*[†]
Eastman Kodak Company
Rochester, New York, USA

Abstract

A system is presented that is capable of automatically detecting specific locations of major facial features such as the eyebrows, eyes, nose, mouth, and facial boundary in portrait images. The system uses multiple stages to determine the feature positions. These stages include skin segmentation, eye detection, and facial feature finding. Each successive operation uses the output of the preceding step. Efficiency is achieved by using the least expensive operations to constrain the search area for the more expensive ones. A robust system has been created wherein each stage is tolerant of inaccuracies in the preceding stages.

Introduction

Automatically identifying the positions of facial features is useful for many photofinishing applications. These applications include skin blemish and wrinkle removal, balancing of skin color tones, red-eye detection and removal, and automated zooming and cropping. This ability is also useful for a variety of other applications including face recognition and classification, facial morphing and warping, and expression recognition.

This paper focuses on the problem of detecting facial features in close-up portrait images. This style of picture is commonly used in the applications listed above. The use of this style of images simplifies the detection task. The pictures are assumed to contain a single individual in a frontal pose. The photographs are also assumed to be reasonably high resolution and taken against a non-skin colored background.

A number of difficult challenges must still be overcome to automatically detect the locations of facial features. These difficulties are easily overlooked because of the adeptness of the human visual system. The challenges include variable image capture conditions (exposure, color balance, focus, resolution, etc.), changes in the scale, position, and pose of the head, face shape variety, diversity of facial expressions, differences in hair and skin colors, the presence of occluding objects such as beards and glasses, and misleading facial contours caused by shadows and wrinkles.

The goal of this work was to develop a system that can robustly detect the precise locations of the major facial features. These features include the eyebrows, eyes, nose, mouth, and facial boundary. The method used should both identify semantically meaningful locations and enable the creation of masks of various facial regions. It is important for the system to accurately identify the feature positions and to execute within a short period.

The remainder of the paper is organized as follows. The second section discusses previous work in this area. The third section presents the automatic facial feature finding system. Results are shown in section four, and are followed by concluding remarks in section five.

Background

The first step toward finding the facial features is to identify the coarse position of the face. There have been a number of techniques proposed to do this automatically.[1,2] However, these algorithms tend to be designed to detect multiple faces in a non-restricted class of images. As a result, they tend to be overly expensive for the relatively simple task of detecting faces in portrait images.

Skin segmentation can provide a high-speed alternative to face detection. Many authors have developed approaches for modeling the color distribution of skin.[3,4] These methods typically do not address the problem of overlapping skin and non-skin color distributions. A model is typically created that attempts to account for a wide range of potential skin colors. This can lead to an overly general model that produces a higher false positive rate than is necessary.

The eyes are one of the most consistent and distinctive facial features. Eye detection is therefore a relatively robust mechanism for refining the location of the face. Prior work on eye detection can be grouped into roughly two categories.[5,6] The first class uses template matching to determine locations with appropriate appearance. The second class uses color and geometric reasoning to determine positions with appropriate attributes. The

[*] Email: mark.bolin@kodak.com
[†] Email: shoupu.chen@kodak.com

approaches that use template matching tend to be more robust, but the techniques that use color and geometric reasoning are often quicker. The previous approaches to eye detection are rarely designed to search only within a constrained face region. This limits their ability to exploit color differences between the eyes and the surrounding skin.

A facial feature finding algorithm can be used to locate specific feature points once the position of the face has been identified. Techniques have been suggested to find facial features that are based on template matching[7], edge detection[8], shape models[9], and holistic matching[10]. Techniques that employ shape models appear to offer the most robust solution. These methods can often overcome occlusions and local matching errors by ensuring that the placement of the feature points conform with a global model of their expected relative positions.

Facial Feature Finding System

The facial feature finding system consists of three primary stages. These stages are skin segmentation, eye detection, and facial feature finding. The skin segmentation stage determines the skin colored pixels within an image. These pixels are used to identify the approximate location of the face. The eye detection stage ascertains the placement of the eyes within the face region. This yields a more specific indication of the face position. The facial feature finding stage determines the specific locations of the major facial features. These feature positions are refined from an initial estimate based on their average locations relative to the eyes.

The system is designed to be both efficient and robust. The stages are cascaded so that the least expensive operations constrain the search area for the more expensive ones. Each stage is engineered to tolerate errors in the preceding step.

Skin Segmentation

Skin segmentation is performed on a coarse version of the original image. This expedites the segmentation without adversely affecting the results. Skin segmentation is intended to detect the rough position of the face and does not require the same resolution as feature finding. This enables the image to be downsampled by two in both the horizontal and vertical directions. Minimally framed portrait images typically contain a border around the subject's head and some portion of the neck and shoulders. This allows 15% to be cropped from each side, 10% from the top, and 20% from the bottom without removing significant portions of the face.

The color of skin pixels can be affected by a variety of factors including lighting, exposure, and skin tone. This causes a significant overlap between the skin and non-skin color distributions. A model of the color distribution of skin was created that does not attempt to include all potential skin colors. Instead, a smaller region is modeled that contains the majority of the skin colors. A nonlinear process is applied when needed to shift skin colors from outside the majority region into the detectable area. This technique minimizes the overlap between skin and non-skin color distributions and results in a lower false positive rate.

A test is performed to determine if an image should go through the nonlinear transformation. The mean intensity of each color channel is compared against a threshold. If the mean of any channel is below the threshold, the nonlinear transformation is applied to the image. This threshold was experimentally determined using a large number of sample images.[11]

The color histogram equalization technique developed by Yang and Rodriguez[12] is currently being used for the nonlinear transformation. This method produces a color shift in our classification space. This shift has been shown to compensate for the changes in skin chrominance caused by variation in lighting, exposure, and skin tone.[11]

After the optional nonlinear transformation, the image is converted to a normalized RG space before classifying the pixels. The transformation to this space is $R_N = R / (R + G + B)$ and $G_N = G / (R + G + B)$. This transformation can be computed rapidly, reduces the effects of lighting and image exposure, and uniquely represents normalized colors using two color channels.

A skin color classifier has been constructed in this space. Skin color patches were collected from over a thousand images from various sources. The colors present were projected onto the (R_N, G_N) plane. The distribution density of these colors forms an approximately elliptical region. The ellipse that best fits this region was calculated using a technique based on moments.[13]

A binary classification is performed on each pixel in the coarse image. If the color falls within the elliptical boundary the pixel is determined to portray a skin color. Pixels whose colors fall outside this boundary are deemed non-skin.

Eye Detection

The skin region is used to constrain the area to search for the eyes. An ellipse is fit to the skin pixels to determine the approximate location of the face. A loose area is defined relative to this ellipse that is likely to contain the eyes.

A hybrid system is used to detect the location of the eyes within the search area. Algorithms that are based on color and geometric reasoning are generally fast but less robust. Algorithms that use template matching are more robust, but significantly slower. Our system seeks to combine ideas from both approaches to produce a technique that is both efficient and robust. A flowchart of the system is depicted in Figure 1.

A color classifier is initially used to detect candidate iris locations. The iris pixels typically have a lower red intensity than the surrounding skin pixels encountered within the search area. A Bayesian probability model is used to determine the likelihood that a given red intensity indicates an iris pixel.

The classifier was constructed from a large number of frontal face images. The sample images were manually

segmented into iris and non-iris regions. The non-iris regions were selected to have the same extent as a typical search area.

The probability that a pixel with a particular red intensity (*I*) is an iris pixel is given by Bayes rule:

$$P(iris \mid I) = \frac{P(I \mid iris)P(iris)}{P(I)}. \quad (1)$$

The term *P(I|iris)* indicates the probability that an iris pixel will have color *I*. This value is calculated by counting the number of times the color occurs within the iris pixels and dividing by the total number of iris pixels. The probability that a pixel is an iris pixel (*P(iris)*) is given by dividing the number of iris pixels by the number of both iris and non-iris pixels in the search area. The value of *P(I)* is the probability of occurrence of the color *I* in both the iris and non-iris regions. This is computed by dividing the number of pixels with color *I* by the total number of pixels.

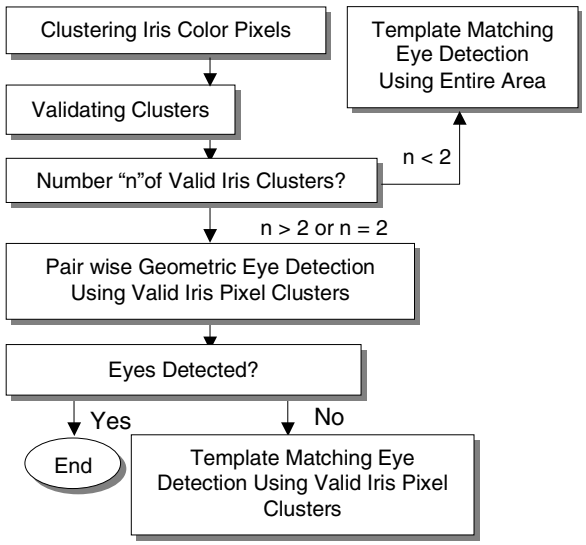

Figure 1. Flowchart of the hybrid eye detection system.

Potential iris pixels are identified as those that exhibit a higher than threshold probability. These pixels are grouped into clusters. Clusters of pixels are defined so that no pixel is further than a fixed distance from another pixel within the same cluster.

The individual clusters are then validated. Clusters with abnormal size or shape are removed from further consideration. If at least two clusters remain then the geometry of pairs of clusters are validated. Pairs with unusual positions relative to the elliptical search area and pairs with inconsistent size are eliminated. If only one pair remains the eyes are deemed successfully located.

If there is more than one pair of valid clusters then template matching is performed at the location of the clusters in order to select the best pair. The eye templates are cropped from an image that is the average of a large number of face images of various races, ages, and genders.[5] Pairs of image patches are removed from the test images that are scaled to have the same extent as the template. The pair that has the smallest sum of squared difference from the templates is selected as the eye locations.

In cases where the system is unable to identify at least two valid clusters, template matching is performed on the entire search area. The search area is resized to have the same extent as the source of the templates. The templates are compared against numerous locations to identify candidate eye positions. The pair with a valid geometric relationship and minimum sum of squared differences from the templates is chosen as the eye locations.

Facial Feature Finding

Eighty-two feature points are detected. These points are organized in a number of connected groups that indicate the outline of the eyebrows, eyes, nose, mouth, and facial boundary. Points are also placed at the center of the pupils and the tip of the nose.

The locations of the features are initially estimated based on their average placement relative to the detected eye positions. The mean shape of the features was determined from approximately 100 manually annotated examples. These examples were aligned and averaged using an iterative least squares technique.[14] The Euclidean transformation is computed that aligns the centers of the pupils of the mean shape with the detected eye points. This transformation is applied to all points in the mean shape to yield the initial estimates.

The positions of the facial features are refined using a variation of the active shape model technique developed by Cootes *et al.*.[9] This algorithm determines the locations of the feature points by performing a series of local searches for positions with matching textural appearance and constraining the results based on a global model of the plausible shapes. This process is repeated a number of times at multiple resolutions to converge upon the final result.

Models of the local appearance of the features were constructed from the annotated examples. At each feature point a 1 by 15 pixel texture window was extracted from each example image. The major axis of this window is oriented normal to the connections between feature points. The extent of the window is scaled to cover a consistent area regardless of the size of the face. The scale factor is derived by using a least squares fit to determine the Euclidean transformation that best aligns the example shape with the mean shape. A gradient profile is calculated for each RGB color channel in the texture window. The values of this profile are given by the difference between neighboring pixels. The gradients are normalized by dividing by the mean intensity and combined into a single vector $\mathbf{t}$. The appearance vectors from all examples are used to compute the mean appearance vector $\bar{\mathbf{t}}$ and the covariance matrix $\mathbf{S}_t$ for the given feature point.

A local search is performed around the estimated location of each feature point to determine the position that

best matches the appearance model for that point. Texture windows are extracted at a 3 by 7 set of points arranged around the feature. The dominant search direction is oriented normal to the feature connections. Both the texture windows and the search intervals are dynamically scaled to cover consistent regions of the face. The scale factor is based on the current estimate of the features and computed using the same technique employed when building the texture models. The dynamic scaling technique is a novel contribution that reduces the sensitivity of the results to the accuracy of the estimated eye locations. The textures within the windows are encoded as before. The Mahalanobis distance is used to measure the similarity of a texture (**t**) to the model. This distance is given by

$$f(\mathbf{t}) = (\mathbf{t} - \bar{\mathbf{t}})^T \mathbf{S}_t (\mathbf{t} - \bar{\mathbf{t}}) . \quad (2)$$

The location of the window with minimum distance is selected as the new feature position.

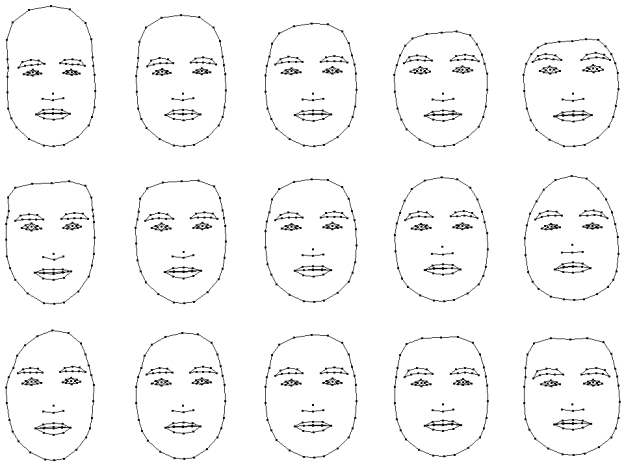

Figure 2. Three most significant axes of variation of the shape model. Modes decrease in significance from top to bottom and shapes range from -2 to +2 standard deviations from left to right.

A model of the global shape of the features was created from a principal components analysis of the annotated examples. The feature locations of the aligned examples were arranged into 1-dimensional coordinate vectors. An ordered list of the most significant axes of shape variation is given by the unit eigenvectors $\mathbf{v}_k$ such that

$$\mathbf{S}\mathbf{v}_k = \lambda_k \mathbf{v}_k , \quad (3)$$

where **S** is the covariance matrix for the coordinate vectors and λ_k is the kth eigenvalue and $\lambda_k \geq \lambda_{k+1}$. The eigenvalues indicate the variance of the examples along the corresponding eigenvector. The three most significant axes of the shape model are illustrated in Figure 2. The majority of the shape variation can typically be explained with relatively few of the primary axes. We retain the most significant M axes that encapsulate 99% of the shape variation. The final model consists of the mean shape ($\bar{\mathbf{x}}$), the primary axes ($\mathbf{v}_k$), and their expected ranges (λ_k).

The shape model is used to constrain the results of the local searches to positions that form a plausible global shape. The current feature locations are converted to the coordinate system of the shape model. This is accomplished using the Euclidean transformation that produces the best least squares fit between the current shape and the mean shape. The feature coordinates are then arranged into a vector (**x**) and projected into the PCA shape space using the expression

$$\mathbf{b} = \mathbf{V}^T (\mathbf{x} - \bar{\mathbf{x}}) , \quad (4)$$

where $\mathbf{V} = (\mathbf{V}_1 \mathbf{V}_2 .. \mathbf{V}_M)$ is the matrix of the first M eigenvectors and $\mathbf{b} = (b_1 b_2 \ldots b_M)^T$ is a vector of shape coefficients for the primary axes. The shape coefficients are constrained to their expected ranges and projected back to feature coordinates using the expression

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{V}\mathbf{b} . \quad (5)$$

Finally, the feature locations are converted back from the coordinate system of the model to image coordinates by inverting the Euclidean transform.

There are a number of approaches that can be used to constrain the shape coefficients to their expected range. Cootes suggests either using box limits and independently cropping each coefficient to a range of three standard deviations or using an elliptical chi-squared limit and scaling the shape toward the mean until it is within range. Neither approach yields the most similar shape within the allowable space. A shape can be considered as a point in the multi-dimensional shape space. The limits of this space can be modeled as a hyper-ellipse with an extent of a uniform number of standard deviations on each axis. The nearest shape can be found by determining the point on this hyper-elliptical boundary that is nearest to the point that represents the current shape. Determining the nearest boundary point requires the solution of a high degree polynomial. This solution can be found numerically using a reasonably straightforward extension of the algorithm for 3-dimensions created by Hart.[15]

A multi-resolution search is used to efficiently converge upon the feature locations. Texture models were created at four different scales. Each successively finer model covers half the extent of the previous model. The search intervals at each finer resolution are also decreased by a factor of two. The searching algorithm initially uses the coarsest texture models and the widest search areas. At each resolution, the process of locally searching for the feature points and constraining the global shape is repeated three times in order to converge upon a stable result. This procedure is repeated at each resolution using successively finer texture models and more narrowly spaced search intervals.

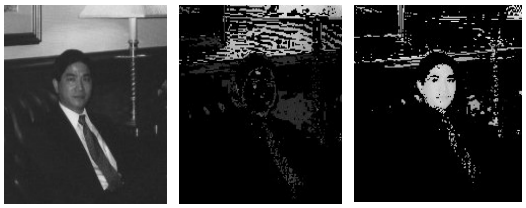

Figure 3. Left: Original image. Middle: Skin segmentation without exposure compensation. Right: Skin segmentation with exposure compensation.

Results

The skin segmentation technique was tested on a variety of images from a number of sources. Forty-seven million pixels were involved in the evaluation. The true positive rate was 88% and the false positive rate was 13%. The average execution time for a 500 by 350 image was 0.57 seconds on a 500 MHz Pentium III. An example is shown in Figure 3. This example illustrates that without exposure compensation, the skin region is not detected. After applying histogram equalization, the color distribution is altered and the skin color moves into the detectable region.

The eye detection algorithm was evaluated using 89 portrait images. The results of the automatic skin segmentation were used as input. The iris locations were identified correctly in 81% of the images. The eye detection stage executed in 0.47 seconds using the same image size and machine mentioned above. The algorithm identified exactly two valid clusters in 45% of the images, template matching was performed at a number of cluster locations in 45% of the cases, and the algorithm had to resort to template matching across the entire search area only 10% of the time.

The accuracy of the facial feature finding stage was determined using a leave-one-out test. The automatically detected eye locations were used as input. The accuracy was measured relative to manually annotated ground truth. The algorithm was able to identify the correct facial features in 96% of the images. The average error of the feature points was 4.0 pixels in the correctly identified images. The average interocular distance in the test set was 82 pixels. The average execution time for the facial feature finding stage was 4.06 seconds. It should be noted that the algorithm was able to correctly find the facial features in a number of cases where the eyes were incorrectly identified. In many of these cases an eyebrow was incorrectly determined to be an eye location. However, these positions were still close enough to enable the correct features to be identified. Examples of the automatically identified feature positions are shown in Figure 4.

Conclusion

A system has been presented that can automatically detect the locations of major facial features in portrait images. The position of the eyebrows, eyes, nose, mouth, and facial boundary are identified by a series of feature points around their border. These points indicate semantically meaningful locations and can be used to create masks of various facial areas. This information is useful for photofinishing and a variety of other applications.

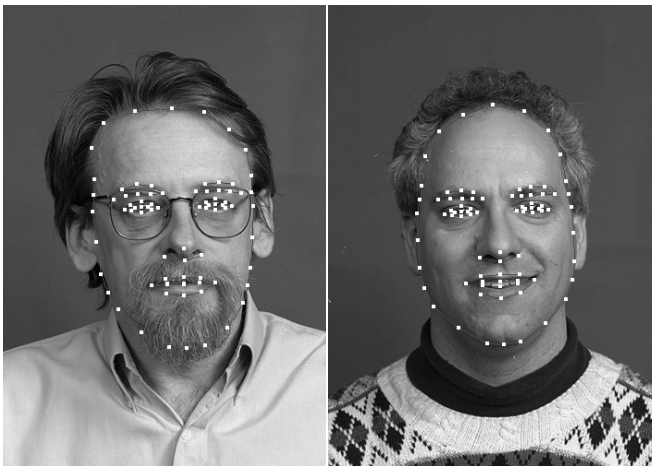

Figure 4. Results of the automatic facial feature finding algorithm.

Skin segmentation is used to identify the coarse position of the face. This technique yields acceptable results for the constrained case of portrait images and executes significantly faster than other more complex face detection methods. The skin segmentation algorithm uses an optional color histogram equalization step to minimize the overlap of the skin and non-skin color distributions. The reduced overlap enables the use of a more specific skin color model that produces a lower false positive rate.

The positions of the eyes are detected to refine the estimated location of the face. The skin region is used to determine the area where the eyes could potentially reside. A hybrid eye detection algorithm was proposed that fuses ideas from techniques based on color and geometric reasoning with those based on template matching. A Bayesian iris pixel detector is combined with geometric reasoning to rapidly detect candidate eye locations. Template matching is used to robustly resolve uncertainties. This provides a solution that retains the best attributes of both techniques.

A facial feature finding algorithm is used to determine the positions of specific feature points. The locations of the features are initially estimated based on their average positions relative to the eyes. The estimates are refined using a variation of the active shape model technique. The multi-resolution algorithm identifies feature points by repeatedly performing a series of local searches and constraining the results based on a global model of the plausible shapes. The sensitivity of the results to the detected eye locations is reduced by dynamically scaling the texture windows and search intervals. The similarity of the constrained feature points to the detected locations is improved by a new method for constraining the shape.

The facial feature finding system is designed so that the least expensive operations are consistently used to constrain the search areas for the more expensive ones. Each of the various stages is also designed to tolerate error in the preceding stage. This yields a system that is both efficient and robust.

References

1. Turk, M. and Pentland, A., "Eigenfaces for Recognition," *J. Cognitive Neurosci.*, **3**, No. 1, 71-86, 1991.
2. Rowley, H.A., Baluja, S., and Kanade, T., "Neural Network-Based Face Detection," *IEEE Trans. Pat. Anal. Mach. Intell.*, **20**, No. 1, 23-38, 1998.
3. Chai, D. and Ngan, K., "Face Sementation Using Skin Color Map in Videophone Applications," *IEEE Trans. Circuits Systems Video Tech.*, **9,** No. 4, 551-564, 1999.
4. Yang, M. and Ahuja, N., "Detecting Human Faces in Color Images," *Proc. 1998 Inter. Conf. Image Proc.*, 127-130, 1998.
5. Luo, J. and Gray, R.T., "Computer Program Product for Locating Object in an Image," U.S. Patent 5,893,837, 1999.
6. Saber, E. and Tekalp, A. M., "Frontal-view Face Detection and Facial Feature Extraction Using Color, Shape and Symmetry Based Cost Functions," *Pat. Recog. Lett.*, **19,** 669-680, 1998.
7. Brunelli, R. and Poggio, T., "Face Recognition: Features versus Templates," *IEEE Trans. Pat. Anal. Mach. Intell.,* **15,** No. 10, 1042-1052, 1993.
8. Kass, M., Witkin, A., and Terzopoulous, D., "Snakes, Active Contour Models," *Proc. IEEE Inter. Conf. Comp. Vis.*, 259-268, 1987.
9. Cootes, T.F., Taylor, C.J., Cooper, D.H., and Graham, J., "Active Shape Models – Their Training and Application," *Comp. Vis. Image Underst.*, **61,** No. 1, 38-59, 1995.
10. Beymer, D., "Vectorizing Face Images by Interleaving Shape and Texture Computations," *MIT AI Memo 1536*, 1995.
11. Chen, S. and Ray, L., "Skin Color Detection and its Application in Bayesian Iris-Based Eye Localization," Submitted to *Comp. Vis. Pat. Recog. Image Proc.*, 2002.
12. Yang, C.C. and Rodriguez, J.J., "Saturation Clipping in the LHS and YIQ Color Spaces," *Proc. SPIE*, **2658**, 297-307, 1996.
13. Sobottka, K. and Pitas, I., "A Novel Method of Automatic Face Segmentation, Facial Feature Extraction and Tracking," *Signal Proc.: Image Comm.*, **12,** No. 3, 263-281, 1998.
14. Cootes, T.F. and Taylor, C.J., "Statistical Models of Appearance for Computer Vision," *WIAU Internal Report*, http://www.wiau.man.ac.uk/~bim/Models/app_model.ps.gz,
15. Hart, J.C., "Distance to an Ellipsoid," *Graphics Gems IV*, Paul S. Heckbert Editor, Academic Press, Boston, MA, 113-119, 1994.

Biography

Mark Bolin received a B.S. in computer engineering from the University of California, San Diego. He received a M.S. and Ph.D. in computer and information science with a specialization in computer graphics from the University of Oregon. He has previously worked for IBM in Tucson, AZ and as a summer intern at Xerox's Webster Research Center in Rochester, NY. He joined Eastman Kodak Company as a Research Scientist in 1999. His current research focuses on computer vision and 3D computer graphics and their applications to digital facial imaging.

Shoupu Chen received a Ph.D. degree in Electrical Engineering in 1992 from the University of Delaware. He joined Eastman Kodak Company's Imaging Science and Technology Laboratory as a research associate in 1998, and has been working on 3D panoramic imaging and facial imaging. He also worked on medical signal analysis at Johns Hopkins University and on image processing, computer vision, and robotics at the University of Delaware. He was an editor of INFORMATION. He served on the technical committee of the Computer Vision, Pattern Recognition, and Image Processing conference in 2000. He was listed in the 1998 edition of American Men and Women of Science. He was a member of Phi Kappa Phi and is a member of IEEE.