



Detection and Inpainting of Facial Wrinkles Using Texture Orientation Fields and Markov Random Field Modeling

Nazre Batool, Rama Chellappa

► To cite this version:

Nazre Batool, Rama Chellappa. Detection and Inpainting of Facial Wrinkles Using Texture Orientation Fields and Markov Random Field Modeling. IEEE Transactions on Image Processing, 2014, 23, pp.3773 - 3788. 10.1109/TIP.2014.2332401 . hal-01096624

HAL Id: hal-01096624

<https://inria.hal.science/hal-01096624>

Submitted on 17 Dec 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Detection and Inpainting of Facial Wrinkles using Texture Orientation Fields and Markov Random Field Modeling

Nazre Batool, *Member, IEEE*, and Rama Chellappa, *Life Fellow, IEEE*

Abstract—Facial retouching is widely used in media and entertainment industry. Professional software usually require a minimum level of user expertise to achieve the desirable results. In this paper, we present an algorithm to detect facial wrinkles/imperfection. We believe that any such algorithm would be amenable to facial retouching applications. The detection of wrinkles/imperfections can allow these skin features to be processed differently than the surrounding skin without much user interaction. For detection, Gabor filter responses along with texture orientation field are used as image features. A bimodal Gaussian mixture model (GMM) represents distributions of Gabor features of normal skin vs. skin imperfections. Then a Markov random field model (MRF) is used to incorporate the spatial relationships among neighboring pixels for their GMM distributions and texture orientations. An Expectation-Maximization (EM) algorithm then classifies skin vs. skin wrinkles/imperfections. Once detected automatically, wrinkles/imperfections are removed completely instead of being blended or blurred. We propose an exemplar-based constrained texture synthesis algorithm to inpaint irregularly shaped gaps left by the removal of detected wrinkles/imperfections. We present results conducted on images downloaded from the Internet to show the efficacy of our algorithms.

Index Terms—Facial Wrinkles, Skin Imperfections, Markov Random Field, Gaussian Mixture Model, Gabor Features, Texture Orientation Fields.

I. INTRODUCTION

DIGITAL image inpainting refers to the filling of the gaps of arbitrary shapes in an image so that they seem to be parts of the original image. Several applications of digital inpainting have been reported in the last decade e.g. filling occlusions/gaps, removal of objects, image reconstruction by removing scratches or other degradation [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13]. Here we propose a specific application of digital inpainting to remove facial wrinkles and imperfections. Traditionally, beautification of skin or facial re-touching in images has been done by professionals using high-end software e.g. Adobe PhotoshopTM. Several user friendly smart phone applications (e.g. Visage LabTM [14],

BeautifyTM [15], Perfect365TM [16]) which provide minimum user interaction for facial touch ups have also been introduced. However, both professional and user-friendly software have limitations. Professional software require significant user interactions where results are subjective, depending on user's expertise. Whereas user-friendly applications developed for smart phones, while performing an overall beautification or making up of skin with minimum user interaction, do not target specific skin imperfections e.g. deep wrinkles, acne, scars etc. An example is shown Fig. 1, where overall beautification of skin fades wrinkles and moles but does not remove them completely. The reason may be that these applications seem to process all the skin regions equally and do not make distinction between skin vs. skin imperfections. The results can be improved if skin imperfections are detected as a pre-processing step and then processed differently from the surrounding skin.

The current state-of-the-art approach for the removal of wrinkles is an image *painting* algorithm proposed by Georgiev [17]. The algorithm is based on the widely used Poisson image editing tool [18] and provides improved seamless image cloning through better handling of lighting variations. The algorithm works behind the *Healing Tool* in Adobe PhotoshopTM. Image painting is slightly a different application from image inpainting. The former deals with inclusion (painting) of a smaller image region in a larger image where both source and destination image regions are provided by the user. The latter deals with the automatic filling of a gap/occlusion, mostly provided by the user, in an image based on local and/or global image characteristics and does not require a source image. However, both applications share the requirement of seamless boundaries. Our work is closer to image inpainting than image painting because both source and destination image areas are selected automatically. We make the following observations about the current facial retouching software as a motivation for our proposed work.

- 1) Significant user interaction is required with the Adobe Healing Tool for the selection of source and destination skin patches resulting in subjective results depending on user expertise.
- 2) In the case of more user-friendly applications, facial retouching results in the so-called *flawless* skin. The processing of skin in an image smoothes wrinkles and skin imperfections but does not remove them completely.
- 3) Regarding image inpainting techniques, both structure

N. Batool is currently with the AYIN team at INRIA-Sophia Antipolis, 2004 route des Lucioles, 06902 Sophia Antipolis Cedex, France. R. Chellappa is with the Department Of Electrical and Computer Engineering and the Center for Automation Research, UMIACS, University of Maryland, College Park, MD 20742, USA.

E-mail: nazre.batool@gmail.com, rama@umiacs.umd.edu

The first author would like to acknowledge the support of Fulbright/HEC(Pakistan)/USAID PhD Scholarship. However, the contents of this publication have not been approved by the representing agencies/governments of the USA/Pakistan.

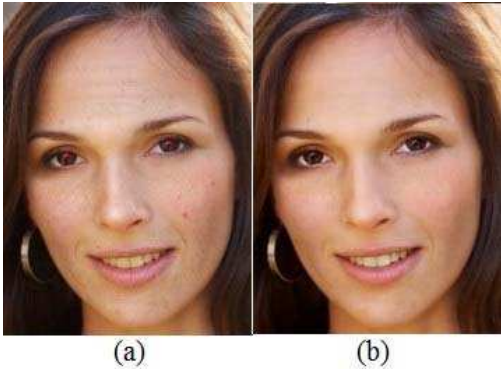


Fig. 1. Typical results of facial retouching for a smart phone application [14]. (a) Original Image. (b) Image after retouching. Note that wrinkles on forehead and brown spots on cheeks are deemphasized due to blending but still visible.

and texture inpainting techniques are not applicable directly to the skin. Wrinkles and skin imperfections do not appear as edges/boundaries and, hence, structural inpainting is not appropriate. Also, as wrinkles are not homogeneous texture patterns, texture inpainting is not effective.

The main contributions of this paper are as follows:

- 1) An algorithm based on the fusion of Gabor features and texture orientation fields in the framework of Markov field modeling (MRF) is proposed to detect wrinkles and other imperfections in the surrounding skin.
- 2) A variation of exemplar-based texture synthesis is proposed to fill the gaps of irregular shapes.
- 3) Both detection and inpainting of wrinkles are unsupervised with minimum user interaction thus minimizing the subjectivity introduced by the user's expertise.
- 4) No 'retouching' or 'beautification' of the rest of the facial skin is done while inpainting skin wrinkles/imperfections.

The organization of this paper is as follows. In section II, we present an overview of some related work. In section III, we present the details of our detection and inpainting algorithms. Experiments and discussion are presented in section IV. Finally, we conclude the paper in section V.

II. RELATED WORK

Image inpainting methods target one or both of the *structure* and *texture* of an image. The difference between the image attributes of structure and texture of an image requires different inpainting methods. A detailed survey of image inpainting methods can be found in [13], [10], [12]. Most texture inpainting methods require user input or some masking function to highlight the gap/occlusion to be filled (e.g. the work by Criminisi [6]). Some examples of automatic filling of scratches, rectangular blocks or random noise can be found in [13], [1], [11]. Shi and Chang introduced a patch-based multi-resolution/multi-layer approach to restore the paintings damaged by red scratches [13]. Their approach involved a mechanism to detect the damaged areas first where the variance in the color of a patch at a specific resolution was used to determine

if a patch had damaged pixels. In contrast, the inpainting methods in [11], [1] do not consider any explicit detection of gaps to be filled. These techniques are based on the analysis of different layers containing low vs. high frequency details. The low frequency layer determines the piecewise smooth regions of the image and the high frequency layer determines the texture. The recovery of these layers automatically fills the gaps without their being detected explicitly. In case where more than one texture is surrounding the gap, sophisticated techniques are used for combining different textures [8], [6]. Once a suitable combination of different textures has been found, the gap is filled by existing texture synthesis techniques. For example, Grossauer [8] used the exemplar-based texture synthesis technique given in [19] and Criminisi et al. [6] used a synthesis method similar to [20].

The specific application of wrinkle removal is different as wrinkles are not artifacts or separate objects to be removed. Wrinkles are an inherent part of the skin and are visible only due to their discontinuous nature in surrounding skin texture. Recently, the detection of wrinkles as sequences of line segments/curves was reported by Batool and Chellappa ([21]). However, this method is not applicable here because of two reasons. First, wrinkles are localized as curves and the surrounding folds of skin due to a wrinkle are not detected. Second, the method reported in [21] is based on line segments and cannot be used to detect other oval like skin imperfections.

Our wrinkle inpainting approach is based on Poisson editing and a variation of exemplar-based texture synthesis. However, we use a novel approach to detect wrinkles and skin imperfections. In the following section we present our approach in detail.

III. APPROACH

An image inpainting technique for textures has three main steps, (a) finding a suitable texture template in the image to fill in the gap with, (b) calculating the seamless warping between the template and the gap and (c) filling the gap via texture synthesis. Since we are proposing unsupervised image inpainting, an additional step is required to detect wrinkles automatically. The process of wrinkling creates deep creases and causes curvature in the surrounding skin. The resulting skin curvature causes specific intensity gradients in skin images which look like discontinuities in surrounding skin textures. An accurate inpainting of wrinkles will require both the wrinkle crease and the surrounding curved skin to be removed. In section III-A, we present our approach for detection. Regarding step (a) of image inpainting mentioned above, we select skin patches surrounding the detected wrinkles. This is due to the fact that the skin texture can vary significantly within a small region of face. The skin patches closest to the wrinkles have the most similar looking skin texture. Regarding steps (b) and (c), we use an exemplar-based texture synthesis method based on the work of Efros and Freeman [22]. The details of our texture synthesis method will be presented section III-B.

A. Automatic Detection of Regions with Wrinkles

We use texture orientation field proposed by Rao and Schunk [23] and Gabor filter responses as image features.

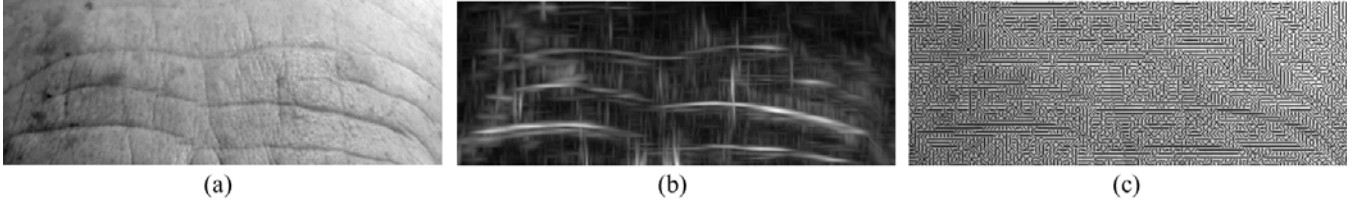


Fig. 2. Image features used for segmentation. (a) Forehead image in gray scale. (b) Maximum Gabor amplitude response (values [4.8, 132] scaled to the gray scale values [0,255]). (c) Texture orientation field.

The orientation field highlights the discontinuities in the normal flow of skin texture whereas the Gabor filter responses highlight the intensity gradients in any directions. The two types of features are fused using Gaussian Mixture Models (GMM) and Markov random field representation. The GMM classifies filter responses as a bimodal distribution for skin vs. skin imperfections. The MRF representation allows us to incorporate spatial relationship among GMM distributions of neighboring pixels and to fuse the orientation fields to reshape the class probabilities.

1) *Computation of Orientation Fields using Gabor Filters:* Several oriented feature detectors have been developed including steerable Gaussian second-derivative filters, line operators and Gabor filters. A comparative study can be found in [24] where the real Gabor filters were assessed to be the best detector of oriented features. The real Gabor filter kernel is given by

$$g(x_1, x_2) = \frac{1}{2\pi\sigma_{x_1}\sigma_{x_2}} \exp \left[\frac{-1}{2} \left(\frac{x_1'^2}{\sigma_{x_1}^2} + \frac{x_2'^2}{\sigma_{x_2}^2} \right) \right] \cos(2\pi f x_1') \quad (1)$$

where

$$\begin{bmatrix} x_1' \\ x_2' \end{bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (2)$$

The parameters α and f denote the orientation angle and frequency of the sinusoidal factor respectively and $\{\sigma_{x_1}, \sigma_{x_2}\}$ denote the standard deviations of the Gaussian envelope in 2D plane. Let $\{g_k(x_1, x_2), k = 0, \dots, K-1\}$ denote the set of real Gabor filters oriented at angles $\alpha_k = -\frac{\pi}{2} + \frac{\pi k}{K}$ where K is the total number of equally spaced filters over the angular range $[-\frac{\pi}{2}, \frac{\pi}{2}]$. Let $\{I(x_1, x_2); x_1 = 1 \dots N_1, x_2 = 1 \dots N_2\}$ denote the input image in gray scale and $I_k^f(x_1, x_2)$ denote the image filtered by the filter $g_k(x_1, x_2)$. Then the orientation field, $\theta_I(x_1, x_2)$ for the image is computed as follows:

$$\theta_I(x_1, x_2) = \arg \max_k I_k^f(x_1, x_2) \quad (3)$$

i.e. at every pixel, the orientation field is equal to the orientation angle of the filter resulting in the maximum filtered response at that pixel. The corresponding maximum amplitude among the filtered responses is given as:

$$I'(x_1, x_2) = \max_k I_k^f(x_1, x_2) \quad (4)$$

The set of the maximum filter response and the orientation angle at every pixel, $\{I'(x_1, x_2), \theta_I(x_1, x_2)\}$, constitutes image features for automatic detection of wrinkle regions. Fig. 2

shows a forehead image with the corresponding maximum responses and the orientation field. The orientation angle is calculated at every pixel, however, the orientation field in Fig. 2(c) is drawn by placing needles at every 3rd pixel. Every needle is of length of 3 pixels and is placed in the direction of the orientation angle.

At high resolution, skin texture appears to be granular resulting in random orientation angles. However, the skin creases of wrinkles and the skin pigments related to other imperfections (e.g. brown spot, moles) smooth out the granular skin texture. As a result, the orientation field depicts two significant properties in wrinkled regions, (a) a dominant angle of zero degrees and (b) pixels with zero orientation angle appear in clusters. Fig. 3 depicts these two properties of orientation field due to wrinkles. We exploit these observations to formulate the GMM-MRF model based, two-class labeling of images into wrinkles and non-wrinkle regions. The next section describes the model in detail.



Fig. 3. (Right) Rectangle 'A' shows the skin texture used as template whereas rectangle 'B' shows change in skin texture due to a wrinkle. (Left) Orientation field at high resolution, note that sites corresponding to the wrinkle have orientation angle of zero degrees.

2) *Gaussian Mixture Model based on Markov Random Field (GMM-MRF):* The motivation behind using the GMM-MRF model is the fact that the Gabor filter responses or the texture orientation field, when used exclusively, are important but insufficient features to detect the wrinkled regions. For example, Fig. 4(a) shows the result of thresholding Gabor amplitude responses in the range [4.8, 132] with the threshold value of 35 and Fig. 4(b) shows image sites with orientation angles lying in the range $[-5, +5]$. Fig. 4(c) shows the product of both results and resembles more closely to the actual wrinkles by reducing false positive in either of the Fig. 4(a) and Fig. 4(b). We make the following observations to justify GMM-MRF modeling.

GMM: Histograms of Gabor response amplitude $I'(x_1, x_2)$ typically follow the Beta distribution with heavy tails. For example Fig. 5 shows a histogram of the Gabor amplitude response for the image in Fig. 2(a). An intelligent thresholding of the Beta distribution can provide a good starting point for any segmenta-

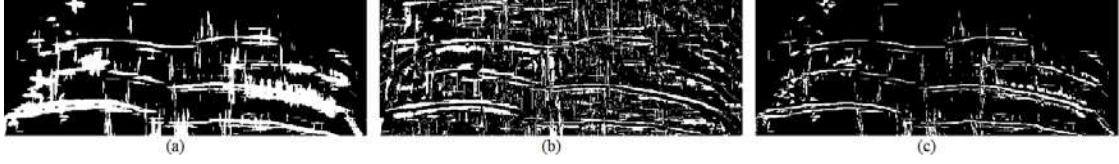


Fig. 4. Results of thresholding. (a) Thresholding maximum Gabor amplitude at value 35. (b) Thresholding orientation field at absolute angle values of less than 5 degrees. (c) Product of images in (a) and (b).

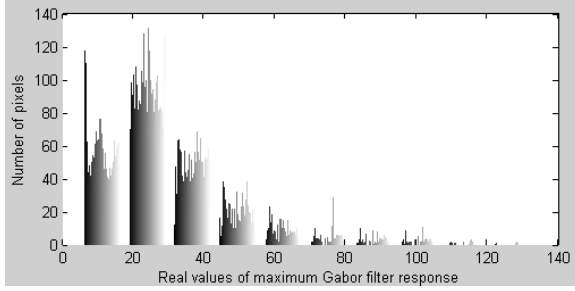


Fig. 5. Histogram of the Gabor features in Fig. 2(b).

tion technique. Modeling of Gabor responses as Beta distribution may seem an obvious choice. However, we take the simpler approach of Gaussian mixture models for its more developed theory. A similar approach can be found in [25] where the authors used the GMM to model the Beta distribution for segmentation of SAR images.

MRF: Since class labels do not depend solely on the Gabor response amplitude, a simple thresholding of Beta distribution does not work. There is always some under segmentation or over segmentation present. Texture orientation field has to be incorporated to aid thresholding by reshaping the probability of each class. An MRF framework enables not only the incorporation of spatial dependencies among neighboring pixels but also the fusion of texture orientation fields and Gabor amplitude responses.

We first present the GMM and MRF models and then discuss how the MRF model is used to fuse the orientation field with Gabor amplitude responses.

Under GMM, the density function for the observation at pixel (x_1, x_2) is given as:

$$f(I'(x_1, x_2)|\Pi, \Theta) = \sum_{j=1}^J \pi_{x_1, x_2}^j \Phi(I'(x_1, x_2)|\mu^j, \sigma^j) \quad (5)$$

where $\Phi(I'(x_1, x_2)|\mu^j, \sigma^j)$ is the standard Gaussian distribution with mean μ^j and standard deviation σ^j and $\Theta = \{(\mu^j, \sigma^j); j = 1, \dots, J\}$ is the parameter set of Gaussian mixture distributions. The set of mixing proportions, $\Pi = \{\pi_{x_1, x_2}^j; x_1 = 1, \dots, N_1; x_2 = 1, \dots, N_2; j = 1, \dots, J\}$ satisfies the following constraints:

$$0 \leq \pi_{x_1, x_2}^j \leq 1 \text{ and } \sum_j \pi_{x_1, x_2}^j = 1 \quad (6)$$

Let \mathcal{I}' denote the ensemble of random variables $I'(x_1, x_2)$ as follows:

$$\mathcal{I}' = \{I'(x_1, x_2); x_1 = 1, \dots, N_2; x_2 = 1, \dots, N_2\} \quad (7)$$

Then, assuming statistical independence of individual pixel sites, the joint conditional density function of the whole image can be written as:

$$p(\mathcal{I}'|\Pi, \Theta) = \prod_{x_1=1}^{N_1} \prod_{x_2=1}^{N_2} \sum_{j=1}^J \pi_{x_1, x_2}^j \Phi(I'(x_1, x_2)|\mu^j, \sigma^j) \quad (8)$$

According to Bayes' theorem, the posterior probability can be written as follows:

$$p(\Pi|\mathcal{I}', \Theta) \propto p(\mathcal{I}'|\Pi, \Theta) \times p(\Pi) \quad (9)$$

Gaussian mixture models based on MRF (GMM-MRF) are proposed to impose spatial smoothness constraints between neighboring pixels [26]. Under MRF models, the prior distribution of the mixing proportion of a pixel (x_1, x_2) , denoted by π_{x_1, x_2}^j , depends on those of its neighboring pixels. The prior joint distribution of π_{x_1, x_2}^j for all pixels is given by the Gibbs distribution:

$$p(\Pi) = \frac{1}{\mathbf{Z}} \exp\left(-\frac{\mathbf{U}(\Pi)}{T}\right) \quad (10)$$

where \mathbf{Z} is the normalization constant, $\mathbf{U}(\Pi)$ is the Gibbs energy function and T is a constant called temperature. According to (8), (9) and (10), the posteriori log-density function can be derived as:

$$\begin{aligned} L(\Pi|\mathcal{I}', \Theta) &= \log p(\Pi|\mathcal{I}', \Theta) \\ &= \sum_{x_1=1}^{N_1} \sum_{x_2=1}^{N_2} \log \left\{ \sum_{j=1}^J \pi_{x_1, x_2}^j \Phi(I'(x_1, x_2)|\mu^j, \sigma^j) \right\} \\ &\quad + \log p(\Pi) \\ &= \sum_{x_1=1}^{N_1} \sum_{x_2=1}^{N_2} \log \left\{ \sum_{j=1}^J \pi_{x_1, x_2}^j \Phi(I'(x_1, x_2)|\mu^j, \sigma^j) \right\} \\ &\quad - \log \mathbf{Z} - \frac{\mathbf{U}(\Pi)}{T} \end{aligned} \quad (11)$$

The expectation maximization (EM) algorithm is usually used to estimate the parameters of a GMM distribution. However, the inclusion of prior distribution to GMM via an MRF introduces additional complexity and the M-step of the EM algorithm cannot be directly applied to estimate the model parameters from the observations. Various approximations have been introduced to tackle this problem. Recently, Nguyen et al.[26] introduced a novel way of incorporating spatial correlations in MRF model which allows a close form solution

at the Maximization step. They introduced a factor G_{x_1, x_2}^j as follows:

$$G_{x_1, x_2}^j = \exp \left\{ \frac{\beta}{2|\mathcal{N}_{x_1, x_2}|} \sum_{i \in \{\mathcal{N}_{x_1, x_2}\}} (z_i^j + \pi_i^j) \right\} \quad (12)$$

where \mathcal{N}_{x_1, x_2} is the neighborhood of the pixel (x_1, x_2) and z_i^j is the posterior probability (later defined in equation 21). The factor G_{x_1, x_2}^j is proportional to the product of both posterior probabilities and prior distributions of the neighboring pixels and causes smoothing of prior probabilities due to the averaging process. In regards with EM algorithm Nguyen et. al. also proposed a new Gibbs energy function update at an EM iteration t based on G_{x_1, x_2}^j as follows:

$$\mathbf{U}^{(t+1)}(\Pi|\Theta) = - \sum_{x_1=1}^{N_1} \sum_{x_2=1}^{N_2} \sum_{j=1}^J G_{x_1, x_2}^{j, (t)} \log \pi_{x_1, x_2}^{j, (t+1)} \quad (13)$$

The factor $G_{x_1, x_2}^{j, (t)}$ is dependent only on the value of the priors and posteriors at the previous step in the EM algorithm. This allows a simpler, closed-form solution for the update of mixing proportions π_{x_1, x_2}^j as will be shown in section III-A4.

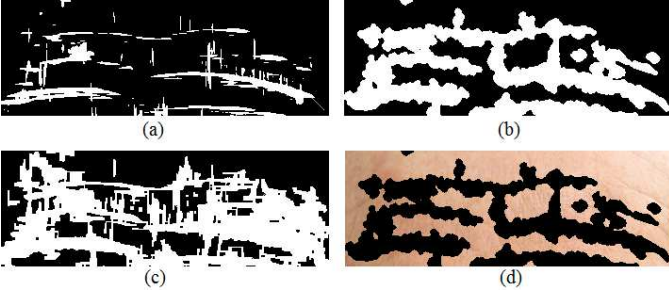


Fig. 6. Binary labeling results. (a) Initial labels obtained after using GMM functions in MatlabTM. (b) labeling results after 15 iterations of our GMM-MRF method (c) labeling results using GMM-MRF method in [26] (d) Resulting gaps in the skin image.

3) Fusion of Gabor Features and Texture Orientation Field:

Under the GMM-MRF model, each pixel shares a global set of parameters of Gaussian distributions denoted by Θ . However, each pixel has a different set of mixing proportions, π_{x_1, x_2}^j , instead of a global set. In this section, we describe our modifications to the GMM-MRF model of Nguyen et. al. [26] for the fusion of texture orientation field and Gabor features. These modifications provide better detection results as compared to the original GMM-MRF model. Fig. 6 depicts an example of differences in detection by both models.

The texture orientation field, $\theta(x_1, x_2)$, is considered a priori field affecting the prior probabilities of mixing proportions. Let $\Omega = \{\theta_{x_1, x_2}; x_1 = 1, \dots, N_1; x_2 = 1, \dots, N_2\}$ denote the set of orientation angles of all pixels. Then, incorporating Ω , equation 9 can be re-written as:

$$\begin{aligned} p(\Pi, \Theta|\mathcal{I}', \Omega) &\propto p(\mathcal{I}'|\Pi, \Theta, \Omega) \times p(\Pi|\Omega) \\ &\propto p(\mathcal{I}'|\Pi, \Theta) \times p(\Pi|\Omega) \end{aligned} \quad (14)$$

where:

$$p(\Pi|\Omega) = \frac{1}{\mathbf{Z}} \exp\left(-\frac{\mathbf{U}(\Pi|\Omega)}{T}\right) \quad (15)$$

$$\mathbf{U}(\Pi|\Omega) = - \sum_{x_1=1}^{N_1} \sum_{x_2=1}^{N_2} \sum_{j=1}^J G_{x_1, x_2}^j(\Omega) \log \pi_{x_1, x_2}^j \quad (16)$$

We introduce the factor $G_{x_1, x_2}^j(\Omega)$ as a function of the orientation field Ω and is given as follows:

$$G_{x_1, x_2}^j(\Omega) = \exp \left\{ \sum_{i \in \{\mathcal{N}_{x_1, x_2}\}} h(j, \theta_i) (z_i^j + \pi_i^j) \right\} \quad (17)$$

The factor $h(j, \theta_i)$ controls the mixing proportions of a pixel based on the orientation field angles in its neighborhood. In the binary case, where $j = 0$ denotes the distribution representing background skin and $j = 1$ denotes the distribution of wrinkled skin, the factor is defined as:

$$h(j, \theta_i) = \begin{cases} 1 & \text{for } j = 0 \\ \beta \cos \theta_i & \text{for } j = 1 \end{cases} \quad (18)$$

The parameter β has a value greater than 1. So whenever a pixel and its surrounding neighbors have orientation angles closer to zero, the value of the factor $h(j, \theta_i)$ increases. This increases the mixing proportion π_i^j and that pixel's probability of being included in the wrinkled area.

4) *Expectation Maximization (EM) Algorithm:* We use the EM algorithm to find mixing proportions and to maximize the posterior distribution. At iteration t , the E-step of the EM algorithm requires the formulation of the following optimization function[27]:

$$\begin{aligned} \mathbf{Q}_{MAP}(\Theta, \Pi|\Theta^{(t)}, \Pi^{(t)}) \\ = \mathbf{E} \left[\log p(\mathcal{I}'|\Theta, \Pi) | \mathcal{I}', \Theta^{(t)}, \Pi^{(t)} \right] + \log p(\Pi) \end{aligned} \quad (19)$$

$$\begin{aligned} \mathbf{Q}_{MAP}(\Theta, \Pi|\Theta^{(t)}, \Pi^{(t)}) \\ = \sum_{x_1=1}^{N_1} \sum_{x_2=1}^{N_2} \sum_{j=1}^J z_{x_1, x_2}^{j, (t)} (\log \pi_{x_1, x_2}^{j, (t+1)} \\ + \log \Phi(I'(x_1, x_2) | \Theta^{(t)}, \Pi^{(t)})) \\ - \log \mathbf{Z} + \frac{1}{\mathbf{T}} \sum_{x_1=1}^{N_1} \sum_{x_2=1}^{N_2} \sum_{j=1}^J G_{x_1, x_2}^{j, (t)}(\Omega) \log \pi_{x_1, x_2}^{j, (t+1)} \end{aligned} \quad (20)$$

where $z_{x_1, x_2}^{j, (t)}$ is given as follows:

$$z_{x_1, x_2}^{j, (t)} = \frac{\pi_{x_1, x_2}^{j, (t)} \Phi(I'(x_1, x_2) | \mu^{j, (t)}, \sigma^{j, (t)})}{\sum_{\tilde{j}=1}^J \pi_{x_1, x_2}^{\tilde{j}, (t)} \Phi(I'(x_1, x_2) | \mu^{\tilde{j}, (t)}, \sigma^{\tilde{j}, (t)})} \quad (21)$$

The constants \mathbf{Z} and \mathbf{T} do not effect the optimization function are set equal to 1. The modified optimization function is then

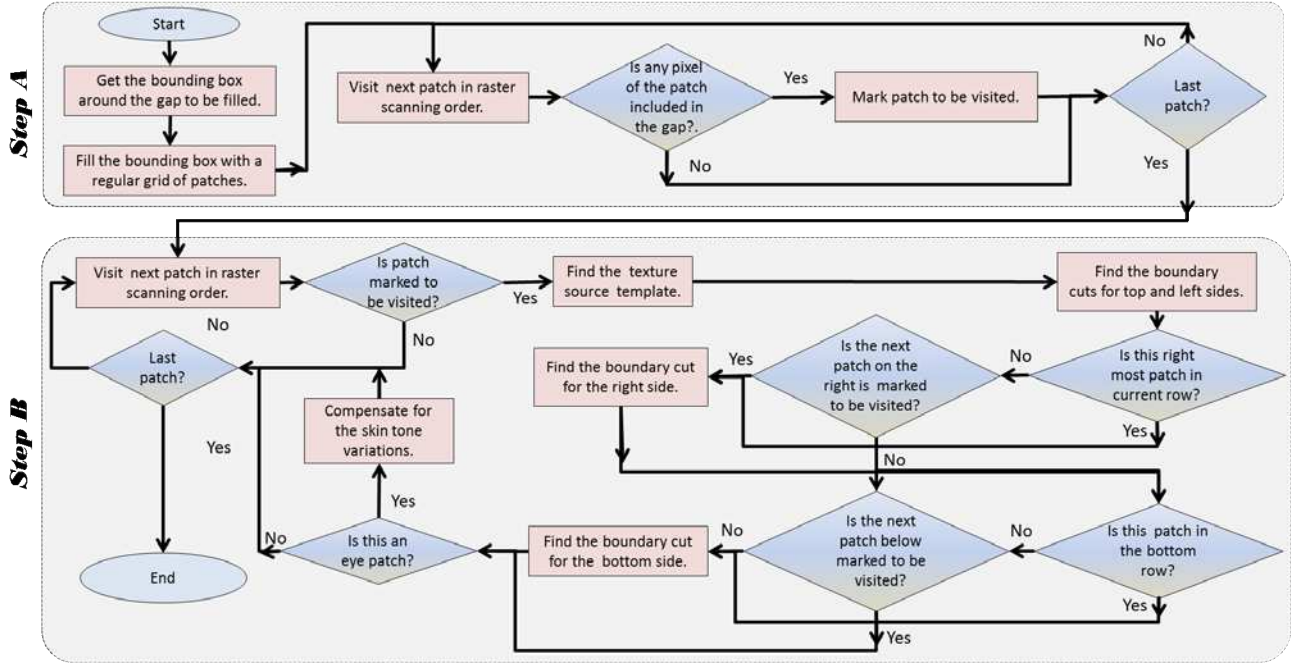


Fig. 7. Flow chart for our Constrained Texture Synthesis algorithm to fill image gaps.

given as:

$$\begin{aligned}
 Q_{MAP}(\Theta, \Pi | \Theta^{(t)}, \Pi^{(t)}) &= \sum_{x_1=1}^{N_1} \sum_{x_2=1}^{N_2} \sum_{j=1}^J z_{x_1, x_2}^{j, (t)} (\log \pi_{x_1, x_2}^{j, (t+1)} \\
 &+ \log \Phi(I'(x_1, x_2) | \Theta^{(t)}, \Pi^{(t)})) \\
 &+ \sum_{x_1=1}^{N_1} \sum_{x_2=1}^{N_2} \sum_{j=1}^J G_{x_1, x_2}^{j, (t)}(\Omega) \log \pi_{x_1, x_2}^{j, (t+1)} \quad (22)
 \end{aligned}$$

$$G_{x_1, x_2}^{j, (t)}(\Omega) = \exp \left\{ \sum_{i \in \{N_{x_1, x_2}\}} (z_i^{j, (t)} + h(j, \theta_i) \pi_i^{j, (t)}) \right\} \quad (23)$$

In the M-step of EM algorithm, the function $Q_{MAP}(\Theta, \Pi | \Theta^{(t)}, \Pi^{(t)})$ is maximized to obtain the updated values of the parameters $\{\Theta, \Pi\}$. After setting the derivative of $Q_{MAP}(\Theta, \Pi | \Theta^{(t)}, \Pi^{(t)})$ to zero, we obtain the following expressions for updating parameters Θ :

$$\mu^{j, (t+1)} = \frac{\sum_{x_1=1}^{N_1} \sum_{x_2=1}^{N_2} z_{x_1, x_2}^{j, (t)} I'(x_1, x_2)}{\sum_{x_1=1}^{N_1} \sum_{x_2=1}^{N_2} z_{x_1, x_2}^{j, (t)}} \quad (24)$$

$$\sigma^{j, (t+1)} = \left[\frac{\sum_{x_1=1}^{N_1} \sum_{x_2=1}^{N_2} z_{x_1, x_2}^{j, (t)} [I'(x_1, x_2) - \mu^{j, (t+1)}]^2}{\sum_{x_1=1}^{N_1} \sum_{x_2=1}^{N_2} z_{x_1, x_2}^{j, (t)}} \right]^{\frac{1}{2}} \quad (25)$$

The values of parameters Π are obtained using the method of Lagrange multiplier (See Appendix A for a detailed derivation) and are given as follows:

$$\pi_{x_1, x_2}^{j, (t+1)} = \frac{z_{x_1, x_2}^{j, (t)} + G(\Omega)_{x_1, x_2}^{j, (t)}}{\sum_{j=1}^J [z_{x_1, x_2}^{j, (t)} + G(\Omega)_{x_1, x_2}^{j, (t)}]} \quad (26)$$

For the EM algorithm, values of the parameters Π, Θ are initialized by using the MatlabTM function 'gmdistribution' to train a GMM with two distributions. Then, following the algorithm given in [26], we proceed with the following steps to complete the automatic detection of wrinkled region.

- Evaluate the values z_{x_1, x_2}^j according to (21) using the current parameter values.
- Update the factor $G_{x_1, x_2}^j(\Omega)$ using (23).
- Update the means μ^j using (24) and variances σ^j using (25).
- Update the prior distributions π_{x_1, x_2}^j using (26).
- Evaluate the log-likelihood in (22) and check the convergence of either the log-likelihood function or the parameter values. If the convergence criterion is not satisfied, then go to step 2.

The result of the EM algorithm is a labeled image where every pixel is assigned the label j with the larger posterior probability value z_{x_1, x_2}^j .

B. Automatic Removal of Facial Wrinkles

The detected wrinkled regions are inpainted by surrounding skin texture using texture synthesis. Texture synthesis techniques can be categorized as parametric or exemplar-based. In parametric methods, the parameters of a generative texture model are learned from a sample texture. A texture image can then be synthesized by sampling the learned model. The exemplar-based methods focus on sampling patches from a sample texture and then stitching them seamlessly, incorporating neighborhood details, to synthesize larger texture images. The exemplar-based methods have become popular in recent years to synthesize 2D texture images, [6], [19], [20], [22]. Parametric methods are appropriate for homogeneous (spatially non-varying) textures where a single set of parameters

can represent the whole texture sample completely. Since, skin textures are slowly varying, inhomogeneous natural textures within face, we adopt exemplar-based approach for efficiency and accuracy. In addition, exemplar-based methods are more suited to the ‘constrained texture synthesis’ problem discussed later in Section III-B2.

Our method is based on the exemplar-based texture synthesis method proposed by Efros and Freeman [22]. In their work, Efros and Freeman introduced a novel method called *image quilting* for seamless stitching of small patches of the exemplar texture. We use their method to stitch skin patches together to fill the gaps left by removal of wrinkled regions. Filling of gaps in images using texture synthesis is also called as ‘constrained texture synthesis’ for the reason that the boundaries between the original texture and the synthesized texture have to be invisible. In section III-B1, we briefly describe the method of Efros and Freeman for seamless stitching of two patches. Then, in section III-B2, we present our algorithm of constrained texture synthesis to fill the gaps. Finally, in section III-B3, we present our method of selecting the skin source texture used for inpainting the gaps.

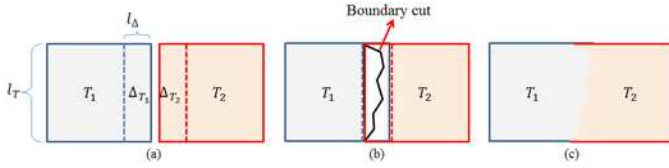


Fig. 8. Seamless stitching of two patches. (a) Patches with side portions to be overlapped. (b) Minimum square distance boundary cut along the overlapped portions. (c) Resulting stitching of the two patches.

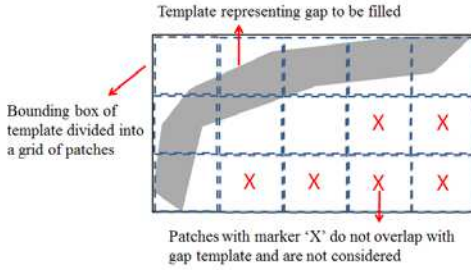


Fig. 9. The constrained texture synthesis algorithm divides an irregular shaped gap into a regular grid of patches. Each patch is then marked to be painted if it overlaps any pixels of the gap.

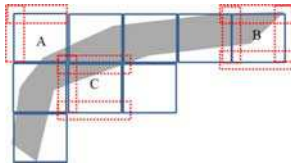


Fig. 10. The sides of a patch used for seamless stitching are based on two factors; the patch's location in the grid and if the surrounding patches are marked to be painted or not.

1) *Patch based Image Quilting*: Let S_T denote a small source texture sample from which the bigger texture image has to be synthesized. Let T_1 and T_2 denote two square patches

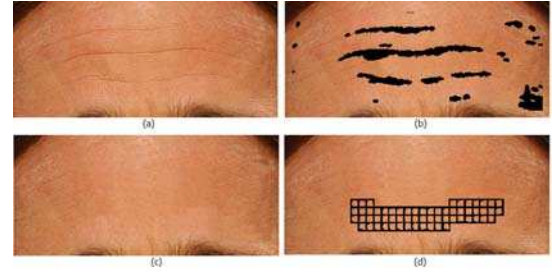


Fig. 11. Wrinkle removal. (a) Original Image (b) Several wrinkled areas detected by GMM-MRF (c) Inpainted image after removal of wrinkles. (d) Patches from regular grid fitted on one gap which were actually inpainted.

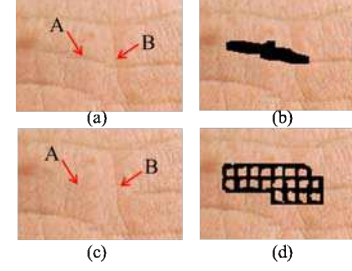


Fig. 12. A close up view of wrinkle removal. (a) Original skin image. (b) Template of the gap to be filled. (c) Skin Image after texture synthesis. Comparing with original skin image in (a), note that wrinkle ‘A’ has been removed since it was included in the gap whereas part of wrinkle ‘B’ is re-painted. (d) Stitching of skin patches to fill the gap.

to be stitched together and l_T denote the length of a side of a patch. Let Δ_{T_1} and Δ_{T_2} denote portions of the patches T_1 and T_2 from any side as is shown in Fig. 8. The portions are set to be of width $l_\Delta < l_T$. The problem of seamless stitching of the two patches then boils down to finding a ragged boundary in the overlapped portions Δ_{T_1} and Δ_{T_2} such that a minimal discontinuity in texture flow is caused across the boundary. This method is explained in Fig. 8.

Given T_1 (the first T_1 can be selected randomly from S_T), the first step is to select a patch, T_2 , of the same size from S_T . This is done by finding one or more patches from S_T having cumulative square difference between the overlapping regions, Δ_{T_1} and Δ_{T_2} , within a given tolerance. The patch T_2 is then selected randomly from such patches. The next step is to stitch the two patches seamlessly by founding the best boundary cut.

Let E be a matrix of size $l_T \times l_\Delta$ representing the square difference between Δ_{T_1} and Δ_{T_2} as follows:

$$E^{i,j} = (\Delta_{T_1}^{i,j} - \Delta_{T_2}^{i,j})^2 \quad (27)$$

where the superscripts i, j designate row and column of the matrix. Then the ragged boundary is found as the cut through E which minimizes the cumulative square difference along the boundary and is found through the following steps.

- Let E_c be a matrix of size $l_T \times l_\Delta$ denoting the cumulative square difference for a boundary cut.
- Initialize the first row of E_c to be equal to the first row of E i.e. $E_c^{i,\cdot} = E^{i,\cdot}$.

- Calculate the rest of the elements of E_c as follows:

$$E_c^{i,j} = E_c^{i,j} + \min\{E_c^{i-1,j}, E_c^{i-1,j-1}, E_c^{i,j-1}\} \quad (28)$$

- Find the minimum value in the last row of E_c . This is the last element of the cut with the minimum square distance. The best boundary can be found by tracing back vertically the cut from this element to the first row of E_c .

2) *Constrained Texture Synthesis to Fill Image Gaps*: The removal of wrinkled skin results in several gaps of irregular shapes as can be seen in Fig. 11(b). Filling such gaps requires modifications to the texture synthesis method presented in the last section which was originally used to synthesis rectangular texture samples. Fig. 7 presents our algorithm to fill irregular gaps using constrained texture synthesis. The algorithm performs two steps for every gap detected by the GMM-MRF algorithm. The first step consists of finding the bounding box for the current gap and fitting it with a rectangular grid of square patches. Then, each patch in the grid is visited to determine if it overlaps with any pixel(s) in the gap. Fig. 9 illustrates this step. The patches which do not overlap with the gap are marked as ‘X’ and are not considered in the second step. In the second step, the patches containing image gap pixels are replaced with the patches of the source skin texture. Each patch is stitched from two (top and left) or more sides depending on its location in the grid. Fig. 10 illustrates this step. Patches are visited in a raster scanning manner. Patch ‘A’ is stitched from top and left side with the rest of the skin image. Patch ‘B’ is stitched from all four sides because there is no patch to be visited on its right or bottom side. Patch ‘C’ is stitched from bottom side as there is no patch below it to be visited. Fig. 11 depicts an example of detected wrinkles (Fig. 11(b)) and a regular grid of patches to remove one of the detected wrinkles (Fig. 11(c)). Fig. 12 shows a further close up of the process of the removal of a wrinkle.

3) *Selection of Texture Source Template*: Usually a texture sample image, S_T , is provided to a texture synthesis algorithm. However, in our case, to minimize the user interaction, a skin texture source template has to be determined automatically. Since facial skin texture varies greatly, for every patch to be inpainted, we use the skin texture nearest to that patch as a source template. A texture source template is selected so that (a) it is of size $1.3 \times T$, where we choose 1.3 because it is small enough for faster computation time and big enough to provide a compatible texture source, (b) it does not overlap with any of the wrinkle gaps and (c) it is nearest to the current patch to be inpainted. Once an S_T is selected this way, a suitable patch, T , can then be found within this texture source template.

4) *Compensation for Skin Tone Variations*: This is a post-processing step and is applied specifically to the areas under eyes. This is due to the fact that the skin under eyes is not only wrinkled, but, frequently, has discolorations due to sagging, under-eye bags or dark circles as well. Although image quilting provides seamless stitching of two patches, its main focus is the overlapping areas δ_T of the two patches. In under-eye regions, the interior of such patches may still present a significant skin tone difference. Therefore, a simple stitching step cannot provide the needed adjustment to the overall tone of the inpainted patch. This is illustrated in Fig. 13 where

boundaries of several patches stitched together are obvious due to the skin tone variation. We use the Poisson image editing [18] to compensate for this tone variation. Bugeau et al. [10] used a similar approach as a post-processing step to compensate for the illumination discontinuities. In our constrained texture synthesis algorithm, once the patch has been stitched, in case of eyes, the Poisson image editing is used as a post-processing step to compensate for the tone variation. The difference in the inpainting results with and without Poisson image editing is shown in Fig. 13.



Fig. 13. Poisson compensation for color variation under eyes. (a) Inpainted skin without Poisson compensation. (b) Inpainted skin with Poisson compensation.



Fig. 14. Level of user interaction. Users have to select polygonal skin patches to remove wrinkles/skin imperfections from.

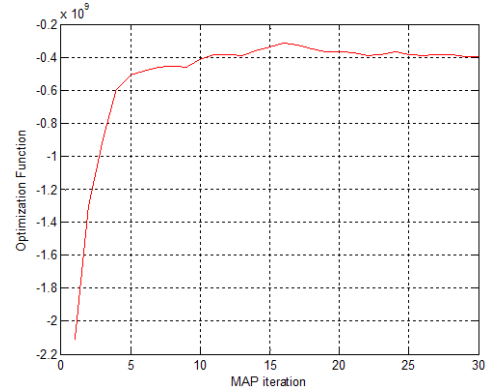


Fig. 15. A typical graph for the objective function vs. iterations of GMM-MRF function. There is little variation in the function for most images after 10 iterations.

IV. EXPERIMENTS

A. Experimental Setup

Experiments were conducted on two sets of images downloaded from the Internet. The first set consists of images of public figures, e.g. celebrities and politicians, and was used to remove facial wrinkles. The second set consists of portions of facial images of other people and was used to remove other skin imperfections. We had to crop portions of interest (e.g. cheeks, forehead) from these facial images to hide identities of

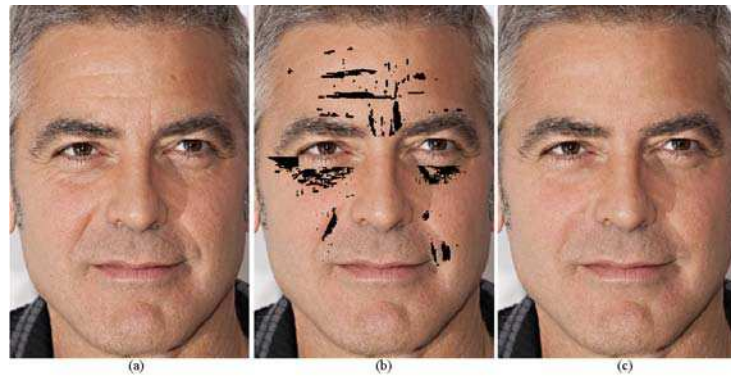


Fig. 16. Results of wrinkle detection and removal for a subject. (a) Original image. (b) Detected wrinkled areas. (c) Image after wrinkle removal.

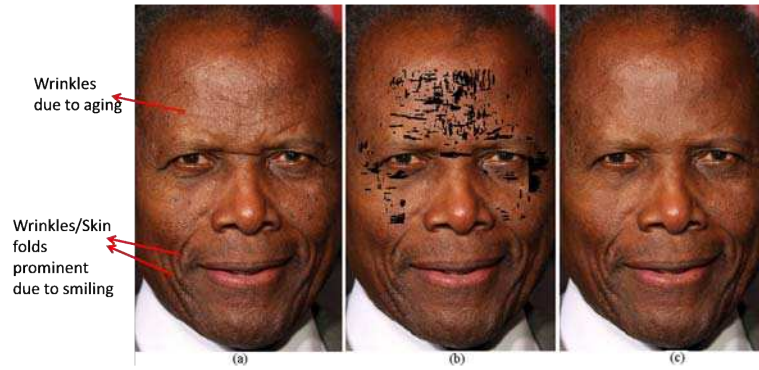


Fig. 17. Results of wrinkle and dark spot detection and removal for a subject with darker skin. (a) Original image. (b) Detected wrinkled areas and dark spots. (c) Image after inpainting, most of the dark spots are removed as well.

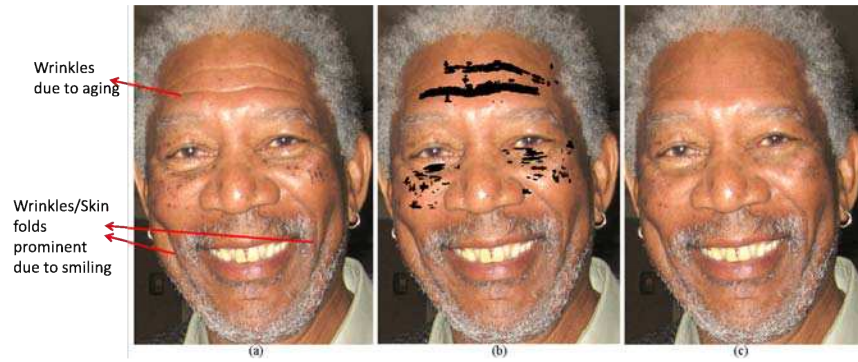


Fig. 18. Results of wrinkle and dark spot detection and removal for a subject with darker skin. (a) Original image. (b) Detected wrinkled areas and dark spots. (c) Image after inpainting, most of the dark spots are removed as well.

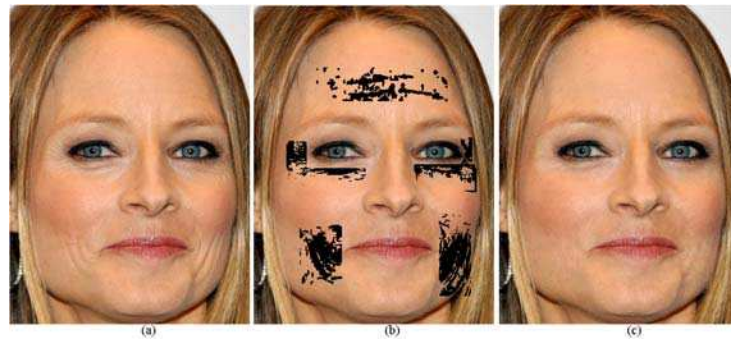


Fig. 19. Results of wrinkle detection and removal for a subject. (a) Original image. (b) Detected wrinkled areas. (c) Image after wrinkle removal.

subjects. All of the images were taken in uncontrolled settings and varied in identity and age of the subjects, illumination and expression. Facial images of public figures consisted of frontal pose whereas other images consisted of side poses as well. All of the images were of high resolution, larger than 1024 pixels x 768 pixels, showing detailed texture of skin. Here we make an interesting observation that facial wrinkles were more prominent in images of male celebrities than those of female celebrities. Hence, most of our experiments were conducted on images of male celebrities. Each image was cropped to have face only and then resized so that the larger dimension was equal to 1100 pixels.

One of the objectives of this work is to minimize user interaction. The only user interaction was to provide polygonal areas of skin to be investigated. Fig. 14 shows two polygonal areas provided by a user to be inpainted for a facial image. The only constraint for these patches was to contain skin and no other facial parts. Since GMM-MRF detection is based on Gabor features which depend on image gradients, the inclusion of facial features other than skin, e.g. hair, eyes, would result in erroneous detection results.

Regarding the parameters of the Gabor filter bank, the spatial frequency of the sinusoid, f , was set to be $\frac{1}{5}$ cycles per pixels. The variances of the Gaussian envelopes were chosen to be $\sigma_{x_2} = L\sigma_{x_1}$ where σ_{x_1} was set to 0.5 and L was set to 25. The value of K was set to be 180. In EM algorithm, the number of iterations were selected to be in the range 5-10 where a smaller number of iterations was sufficient for smaller cropped facial regions. Fig. 15 shows a typical plot between optimization function in EM algorithm and number of iterations where the function depicts smaller variations after 10 iterations. In patch based image quilting the margin width, l_Δ , was set to be equal to $\frac{1}{3}l_T$ whereas the patch size, l_T , was set to be 11 pixels.

B. Results and Discussion

1) *Removal of Wrinkles:* Figures 16, 17, 18 and 19 show the results of removal of wrinkles in facial images of public figures. In all the figures, the original images are shown in parts (a), the gaps resulted by GMM-MRF algorithm are shown in parts (b) and inpainted images are shown in parts (c). The user provided areas for these images contained wrinkles due to aging on forehead and around eyes and mouth. Here we would like to mention briefly the expression wrinkles. Facial expressions not only highlight aging wrinkles but also cause extra folds on facial skin. These folds of expressive wrinkles are usually wider than those of aging wrinkles. When inpainting images for the removal of wrinkles, we make a distinction between expressive and aging wrinkles. The removal of expressive wrinkles may result in unnatural looking facial shapes. Fig. 17 and Fig. 18 show two examples where wrinkles due to smiling were not removed.

Overall, our experiments demonstrate that most of the wrinkles and skin imperfections are detected and inpainted. However, a few with less contrast with the surrounding skin are not detected. For example, small parts of wrinkles on the upper ride sides of subjects' foreheads in Fig. 16 and Fig. 18 remain

undetected. Similarly, very few wrinkles around the corners of the eyes in Fig. 17 remain undetected as well. Regarding areas under eyes, the algorithm removes most of the wrinkles while maintaining the skin tone variation due to dark circles. This effect was desirable as the goal was to remove wrinkles without other beautification of the skin. The under eye dark circles remain unaltered due to the skin tone compensation step described in section III-B4.

2) *Removal of Moles/Dark Spots/Scars:* We also applied our algorithm on other types of skin imperfections, e.g. moles, dark/brown spots, acne, wound scars and freckles. These imperfections also appear as a disruption in the surrounding texture. Since these images did not contain full facial images to hide the identities of the subjects, these was resized to have the maximum dimension equal to 500 pixels instead of 1100 pixels in case of full facial images. Fig. 20 and Fig. 21 show results of removal of these imperfections for subjects of different skin color. Overall, the detection is better in images where skin imperfections have sharper color contrast with the surrounding skin. The imperfections are removed irrespective of the cause, color, size and shape. We observe that acne or wound scars, moles and darker brown spots are detected and removed. Lighter and smaller brown spots were the most difficult kind of imperfection to be detected. Fig. 17 and Fig. 18 show two subjects with darker skin and several moles/dark spots on cheeks which are removed along with wrinkles by the inpainting algorithm.

Fig. 22 shows two examples where some of the brown spots/freckles remained undetected. In part (a), the freckles are over segmented due to their less color contrast with the surrounding skin. In part (b), a large brown spot is present along with few smaller spots. A significant color variation is present within the larger spot which creates inhomogeneous orientation texture field within the spot and not equal to zero throughout the spot. Fig. 22 (b2) shows the inhomogeneous orientation field. Hence, the areas within the brown spot with nonzero, inhomogeneous orientation field are given lower probability of being an imperfection according to the factor $h(j, \theta_i)$ in equation 17 and are left out.

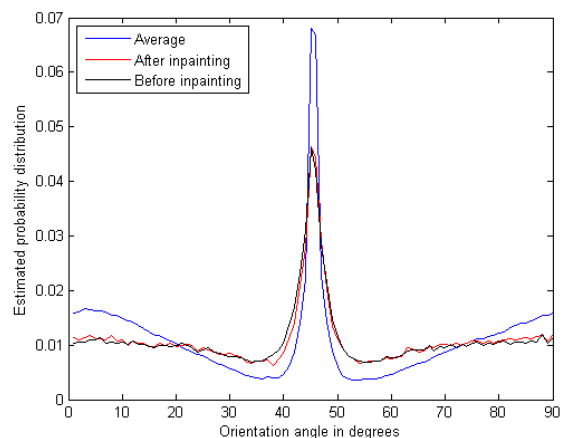


Fig. 24. Histograms of gradient orientations.

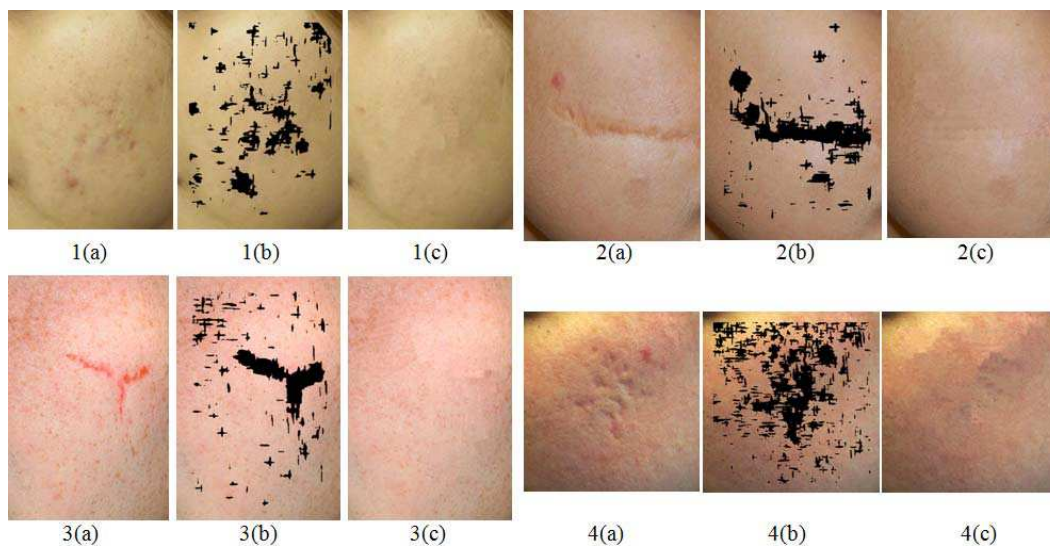


Fig. 20. Results of detection and removal of skin imperfections including wound scars, acne, brown spots and moles. (a) Original images. (b) Detected imperfections. (c) Images after inpainting.

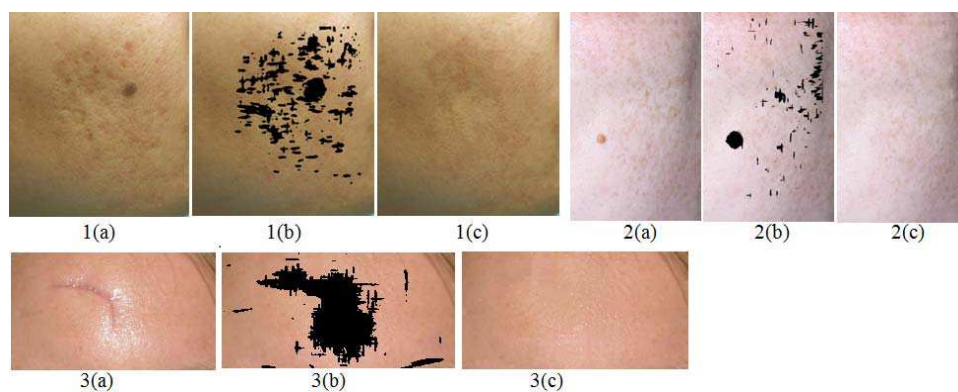


Fig. 21. Results of detection and removal of skin imperfections including wound scars, acne, brown spots and moles. (a) Original images. (b) Detected imperfections. (c) Images after inpainting.

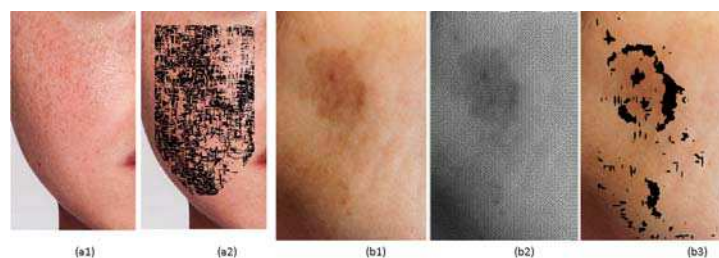


Fig. 22. Challenges in the detection of brown spots. (a) Brown spots are over segmented due to lower contrast with surrounding skin. (b) Brown spot is not detected due to the inhomogeneity of texture orientation field within the spot as shown in (b2).

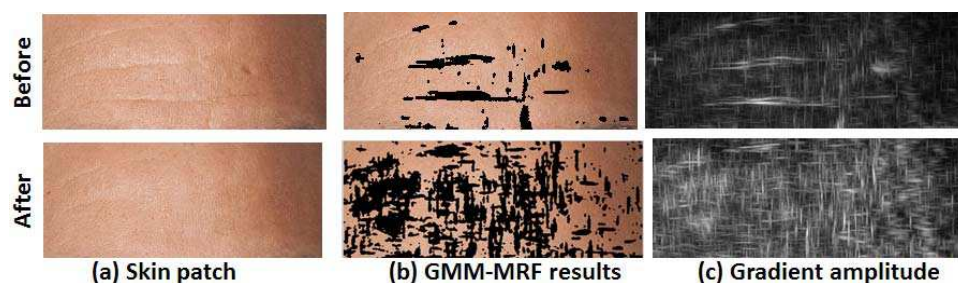


Fig. 23. Changes in skin texture and corresponding GMM-MRF detection results and Gabor amplitude response after inpainting.

C. Quantitative Evaluation of Inpainting Results

In this section, we present quantitative analysis of detection and inpainting results by evaluating smoothness of the inpainted skin texture. In texture synthesis community, visual evaluation of results is usually deemed sufficient. To the best of our knowledge two pieces of work, Lin et al. [28] and AlMeraj et al. [29], are available for the evaluation of texture synthesis for regular to near-regular textures. Since skin textures are natural, stochastic and irregular/inhomogeneous, the evaluation techniques developed for regular textures are not applicable. Here we present an alternate evaluation scheme based on the assumption that the normal skin has irregular, granular texture and imperfections appear as artifacts of some shape (curvilinear, round) in surrounding skin texture. Hence, the random intensity gradients of skin texture are changed by those caused by imperfections. For example, Fig. 23 depicts Gabor amplitude responses and GMM-MRF detection results before and after image inpainting. The skin patch with wrinkles results in distribution of high gradient sites localized in areas around wrinkles with dominant orientation angles. Whereas in inpainted skin patch the high amplitude of intensity gradients is more evenly distributed spatially. For this reason, we use the orientations of intensity gradients as image features to assess image inpainting.

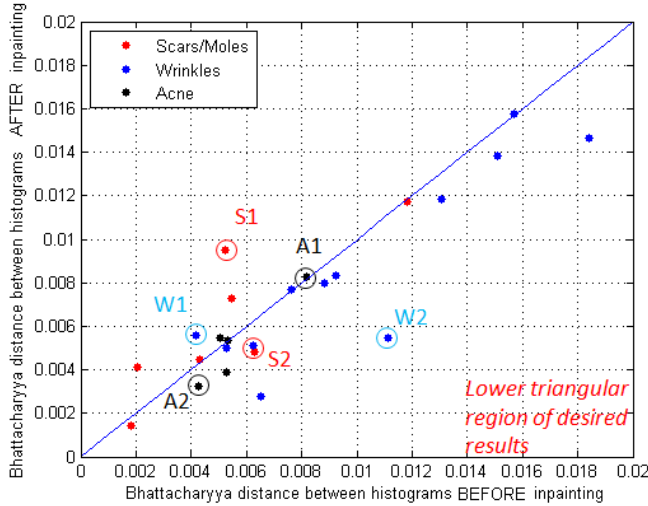


Fig. 25. Quantitative evaluation: Plot between $\text{dist}_B^{(S,W)}$ (x-axis) and $\text{dist}_B^{(S,P)}$ for wrinkles (12 images), scars (7 images) and acne (5 images).

1) *Assessment of Changes in Texture Orientation Field using Gradient Orientation Histograms:* Our quantitative evaluation is based on the calculation of gradient orientation histograms since the presence of skin imperfections causes dominant orientations in otherwise random skin texture orientation field. After inpainting these imperfections can include (a) any scars/wrinkles not detected by GMM-MRF algorithm and (b) any artifacts introduced by the inpainting algorithm. Hence, for quantitative evaluation, how far the gradient histograms of inpainted skin is removed from that of normal skin gives an indication of the remaining imperfections in the skin patch. To begin with, this requires the knowledge of the

Image	Original $\text{dist}_B^{(S,W)}$	Exampler-based $\text{dist}_B^{(S,P)}$	HSR $\text{dist}_B^{(S,P)}$	Ours $\text{dist}_B^{(S,P)}$
Top	0.0065	0.0041	0.0063	0.0028
Middle	0.0062	0.0043	0.0068	0.0080
Bottom	0.0053	0.0070	0.0071	0.0050

TABLE I
COMPARISON OF BHATTACHARYYA DISTANCES.

gradient orientation histogram of normal skin. We estimate the histogram of normal skin, denoted by \mathbf{H}_S , by manually cropping 32 skin patches, free of any imperfections, from different facial regions of 12 subjects. Then, the orientation field θ_I is calculated and divided into 90 bins for each patch. The histogram \mathbf{H}_S is assumed to be equal to the average of the 32 orientation histograms obtained from imperfection-free skin patches. Let the calculated gradient orientation histogram of a given skin patch before and after inpainting be denoted by \mathbf{H}_W and \mathbf{H}_P respectively. Fig. 24 represents three plots, \mathbf{H}_S , \mathbf{H}_W and \mathbf{H}_P where the estimated histogram of gradient orientations for normal skin is similar to the difference of Gaussians distribution. Based on our assumption, \mathbf{H}_P should be more similar to \mathbf{H}_S as compared to \mathbf{H}_W .

Assuming a histogram with large number of bins to be a discrete probability distribution, we calculate the Bhattacharyya distance between \mathbf{H}_S and \mathbf{H}_W denoted by $\text{dist}_B^{(S,W)}$ and between \mathbf{H}_S and \mathbf{H}_P denoted by $\text{dist}_B^{(S,P)}$. Ideally, the distance $\text{dist}_B^{(S,P)}$ should be minimized after inpainting, however, $\text{dist}_B^{(S,P)} < \text{dist}_B^{(S,W)}$ is an indicator of removal of most imperfections and inpainting without introducing artifacts. That translates to the lower triangular part in the plot between $\text{dist}_B^{(S,P)}$ and $\text{dist}_B^{(S,W)}$ given in Fig. 25. We plotted the two distances for 12 wrinkle images, 7 scar images and 5 acne images. It can be seen from the plot that inpainting results in reduction of dominant orientation caused by imperfections for most of the images. The original and inpainted images corresponding to the plot points ‘S1’, ‘W1’, ‘A1’, ‘S2’, ‘W2’ and ‘A2’ are shown in Fig. 26. The plot points ‘S1’, ‘W1’ and ‘A1’ are less than the desired results because inpainting caused artifacts for the image ‘S1’ and did not improve original skin texture significantly for the images ‘W1’ and ‘A1’ which had very light imperfections. For the points ‘S2’, ‘W2’ and ‘A2’ inpainting improved texture because the imperfections in these images were more prominent. Images corresponding to other plot points followed the same pattern.

Overall, we conclude that our algorithm performs much better when the wrinkles, scars and moles are prominent in the surrounding texture. Interestingly, these are the cases where the available retouching software do not perform well as will be described in the following section.

2) *Comparison with existing techniques:* Although our method performs both detection and inpainting, in this section, we present comparison of our inpainting results only with existing inpainting techniques [31], [7] and with two pieces of software, Adobe PhotoshopTM and Perfect365TM. The two inpainting techniques consist of exemplar-based texture filling [7] to remove objects and sparse dictionaries [31] to reconstruct images by removing scratches/small occlusions.

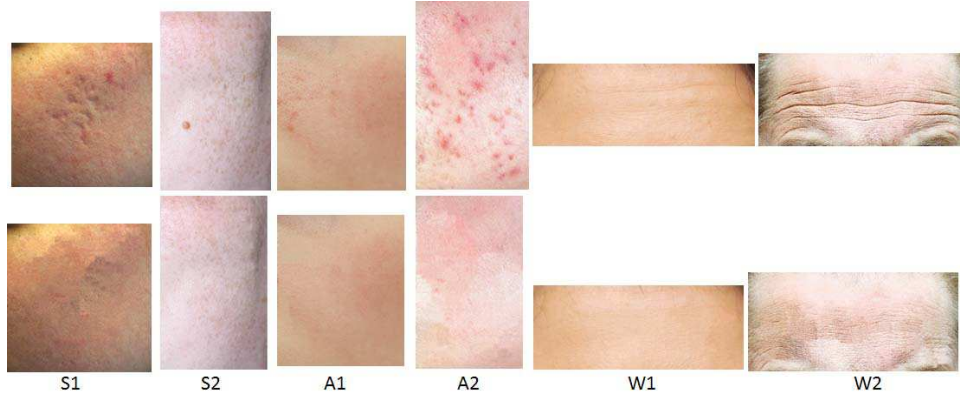


Fig. 26. Skin patches corresponding to plot points in Fig. 25

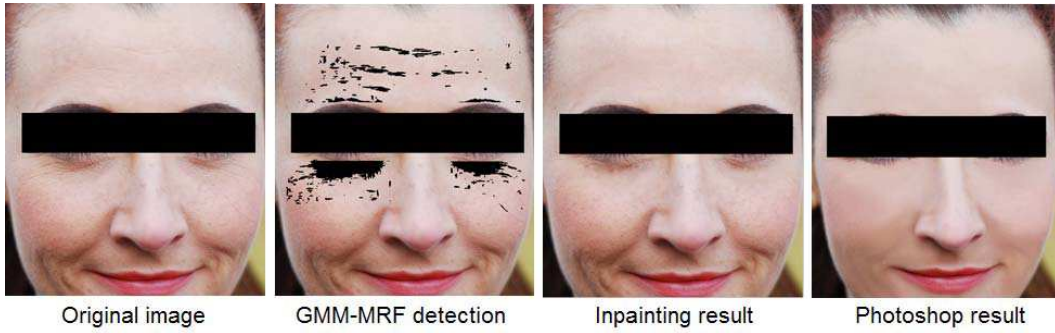


Fig. 27. Comparison of results of wrinkle removal with Adobe Photoshop™ [30].

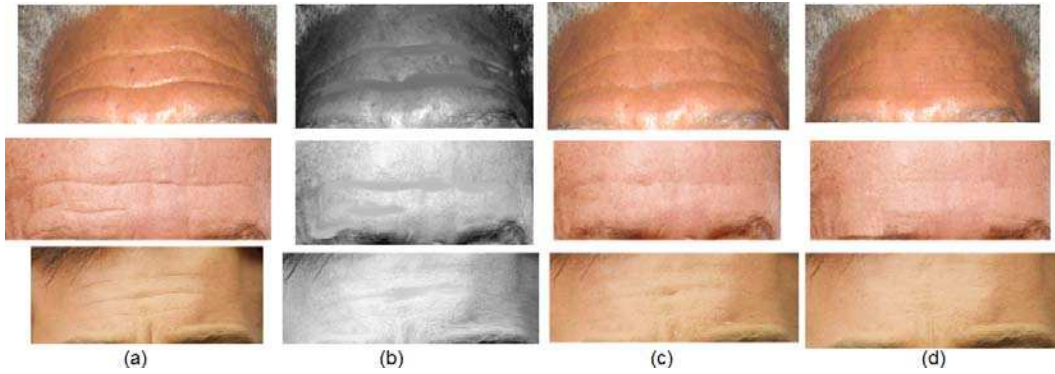


Fig. 28. Comparison of results of wrinkle removal with other inpainting techniques. (a) Original image (b) Sparse dictionaries based image recovery [31] (c) Exemplar-based texture filling [7] (d) Our method.

Three forehead images with two inpainting techniques along with our method are shown in Fig. 28. The inpainting results of exemplar-based method in column (c) are visually more similar to our method in column (d) but wrinkles are not removed completely. On the other hand, our method introduces some artifacts. The results of sparse dictionaries fill in the gaps with smooth regions because of the smoothness priors. Table I includes corresponding calculated histogram distances for three inpainting techniques. Our method has decreased $\text{dist}_B^{(S,P)}$ as compared to other methods for top and bottom images and has increased $\text{dist}_B^{(S,P)}$ for the middle image which can be attributed to the artifacts caused by texture synthesis.

Regarding user-friendly smart phone applications, we selected Perfect365™ because of its popularity and availability of a trial version for PC. However, Perfect365™ provided the options of blemish removal and beautification only and not that of wrinkle removal and required full facial images instead of skin patches. Hence, we used images having blemishes/acne marks for comparison with Perfect365™. Figure 29 shows experimental results on images of three subjects with mild to moderate acne where facial feature have been masked to protect the subjects' identities. It can be seen that Perfect365™ removes less number of blemishes especially on forehead and chin as compared to our algorithm.

Fig. 27 shows an example of the removal of wrinkles with Adobe Photoshop™ and with our algorithm. The original and

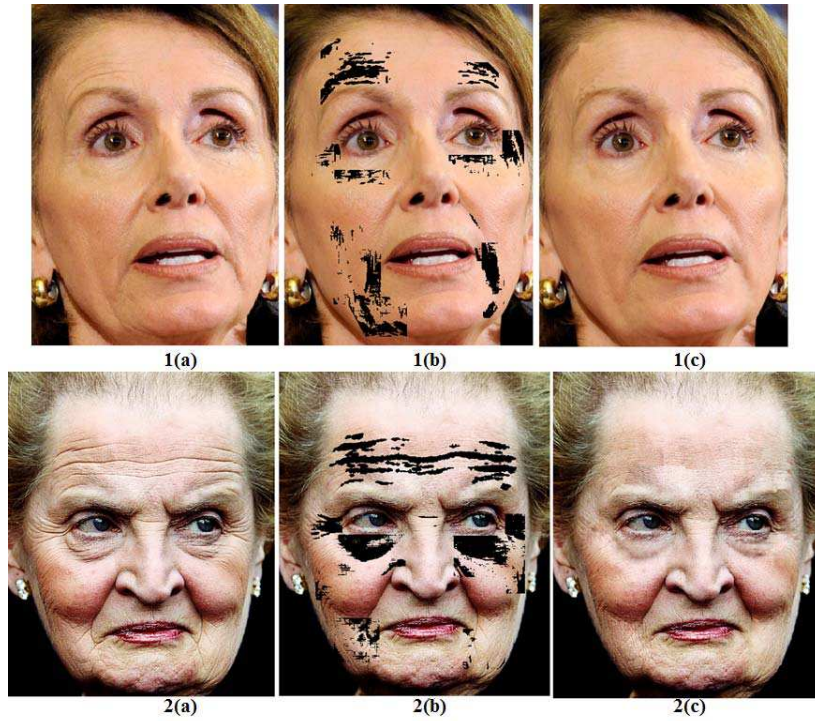


Fig. 30. Results of wrinkle detection and removal for a subject with sagging skin. (a) Original image. (b) Detected wrinkled areas. (c) Image after inpainting, most wrinkles are removed but skin sagging is visible.

the processed high-resolution images were downloaded from the Internet [30] (with author's permission). The differences in results are visible in terms of overall beautification of skin. Our algorithm removes most of the wrinkles (except expression wrinkles around mouth) without any beautification of skin whereas in PhotoshopTM results, facial skin has been changed drastically in addition to removal of wrinkles.

D. Challenges

1) *Effect of Age/Sagging Skin*: Advanced aging skin shows signs of overall sagging of skin as well in addition to rough texture and wrinkles. Facial images where subjects had sagging of skin along with wrinkles posed specific challenges to inpainting. In advanced ages, wrinkles are present very closely, and the area of the non-wrinkled skin is limited. This results in the selection of the same patch as a source skin texture for several patches to be inpainted and results in detectable repetitive patterns in the inpainted skin. Then, the overall sagging of skin results in changes of facial muscles and shape which are visible to the eye but not to the GMM-MRF algorithm and hence cannot be detected and removed. Fig. 30 shows images of two subjects where most of the wrinkles have been removed but the signs of sagging of skin around mouth, jawline and under eyes remain.

2) *Effect of Illumination*: The images used in our experiments had a variety of illumination settings. Overall, small illumination variations did not effect the detection results. However, in cases of significant illumination variations e.g. due to pose or bright spots on skin, the intensity changes due to wrinkles were masked by those due to illumination. Two such examples are presented in Fig. 31. In Fig. 31(1a,1b),

wrinkles are masked by the dark areas due to low illumination and in Fig. 31(2a,2b), wrinkles are masked by the bright spots. However, in both figures, the wrinkles not masked by illumination were correctly detected by the GMM-MRF algorithm.

3) *Artifacts due to the Repetition of Patches*: In few cases, the filling of gaps by the seamless stitching of patches resulted in the artifacts which could be detected by a user. For example in Fig. 18(c) and Fig. 30(2c), some of the inpainted areas can be detected due to the slight change in the textures of the inpainted and the surrounding skin. This happens when the same non-wrinkled, texture source image is the nearest to more than two patches to be inpainted. The algorithm then uses the same source skin texture to paint more than one patch. This results in the absence of randomness found naturally in the slowly varying skin texture. The absence of the randomness and the consistency of the painted texture looks like an artifact to the eye. It was observed that a smaller patch size l_{Δ} provided better, natural looking inpainted skin texture as it is easier to find smaller non-wrinkled, source skin texture patches to inpaint from.

V. CONCLUSION

In this paper, we addressed some of the limitations of current facial retouching applications. We presented an algorithm incorporating Gabor features and texture orientation field of facial skin in the framework of GMM and MRF representations to detect wrinkles and other skin imperfections. Then, we presented an algorithm based on exemplar-based texture synthesis to automatically inpaint the irregular gaps left by the removal of skin wrinkles/imperfections. Experiments on images

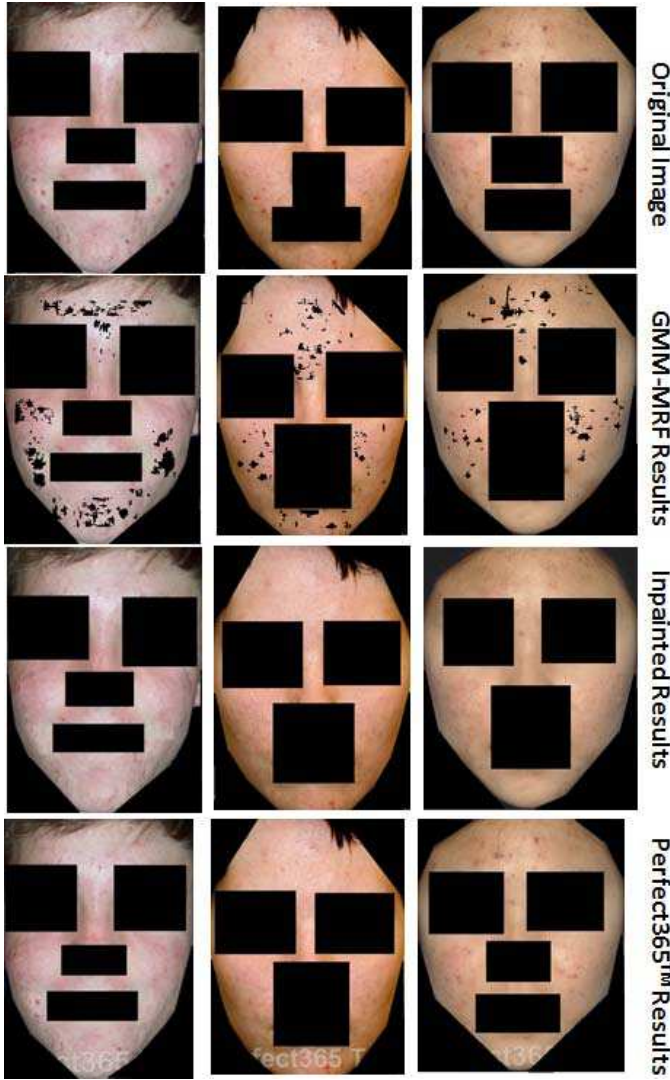


Fig. 29. Comparison of results of acne removal with the user-friendly application Perfect365™.

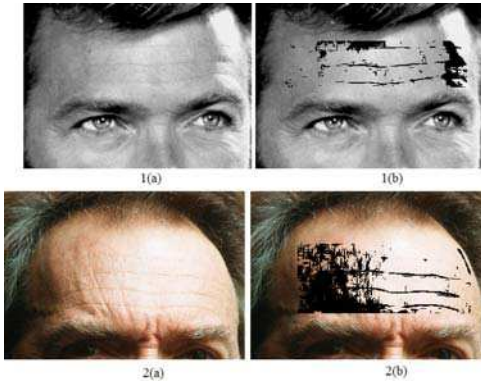


Fig. 31. Effect of Illumination on Detection of Wrinkles. (1a,1b) Wrinkles are masked by bright spots. (2a,2b) Wrinkles are masked by shadows.

downloaded from the Internet show the effectiveness of our algorithms. With minimum user interaction, the algorithms are able to detect and remove most of the wrinkles/imperfections. We discussed some of the challenges in detection and in-

painting. Overall, our algorithm presents significant improved inpainting in cases where skin imperfections are more visible in the surrounding skin. This work can be extended to address the sagging of skin in more advanced stages of aging as well as to improve inpainting to address the artifacts caused by repetition.

APPENDIX A

Let the Lagrange multipliers be denoted by $\{\eta_{x_1, x_2}; x_1 = 1, \dots, N; x_2 = 1, \dots, N\}$. Then the Lagrangian function is given by:

$$\Lambda(x_1, x_2, \eta_{x_1, x_2}) = \mathbf{Q}_{MAP} - \sum_{x_1=1}^{N_1} \sum_{x_2=1}^{N_2} \eta_{x_1, x_2} \left(\sum_{j=1}^J \pi_{x_1, x_2}^{j, (t)} - 1 \right) \quad (29)$$

The values of the Lagrange multipliers at a stationary point of the function $\Lambda(x_1, x_2, \eta_{x_1, x_2})$ can be determined by solving the following set of equations:

$$\frac{\partial \Lambda(x_1, x_2, \eta_{x_1, x_2})}{\partial \pi_{x_1, x_2}^{j, (t)}} = 0 \quad (30)$$

By solving partial derivatives, we obtain:

$$\frac{z_{x_1, x_2}^{j, (t-1)}}{\pi_{x_1, x_2}^{j, (t)}} + \frac{G_{x_1, x_2}^{j, (t-1)}}{\pi_{x_1, x_2}^{j, (t)}} + \eta_{x_1, x_2} = 0 \quad (31)$$

By taking sum of the above equation over J classes and by using the constraint $\sum_{j=1}^J \pi_{x_1, x_2}^{j, (t)} = 1$, values of Lagrange multiplies are determined as follows:

$$\sum_{j=1}^J \left(\frac{z_{x_1, x_2}^{j, (t-1)}}{\pi_{x_1, x_2}^{j, (t)}} + \frac{G_{x_1, x_2}^{j, (t-1)}}{\pi_{x_1, x_2}^{j, (t)}} + \eta_{x_1, x_2} \right) = 0 \quad (32)$$

$$\sum_{j=1}^J \left(\frac{z_{x_1, x_2}^{j, (t-1)}}{\pi_{x_1, x_2}^{j, (t)}} + \frac{G_{x_1, x_2}^{j, (t-1)}}{\pi_{x_1, x_2}^{j, (t)}} \right) = \sum_{j=1}^J \pi_{x_1, x_2}^{j, (t)} \quad (33)$$

$$\eta_{x_1, x_2} = 1 + \sum_{j=1}^J G_{x_1, x_2}^{j, (t-1)} \quad (34)$$

By using the value of η_{x_1, x_2} in (31), the optimal value for $\pi_{x_1, x_2}^{j, (t)}$ can be obtained as:

$$\pi_{x_1, x_2}^{j, (t)} = \frac{(z_{x_1, x_2}^{j, (t-1)} + G_{x_1, x_2}^{j, (t-1)})}{\sum_{j=1}^J (z_{x_1, x_2}^{j, (t-1)} + G_{x_1, x_2}^{j, (t-1)})} \quad (35)$$

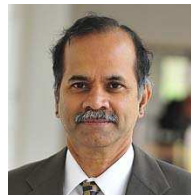
REFERENCES

- [1] M. Elad, J.-L. Starck, P. Querre, and D. Donoho, "Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA)," *Applied and Computational Harmonic Analysis*, vol. 19, no. 3, pp. 340 – 358, 2005.
- [2] G. S. M. Bertalmio, L. Vesa and S. Osher, "Simultaneous structure and texture image inpainting," *IEEE Trans. Image Process, TIP*, vol. 12, no. 8, p. 882889, 2003.
- [3] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '00. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2000, pp. 417–424.
- [4] A. Wong and J. Orchard, "A nonlocal-means approach to exemplar-based inpainting," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, Oct 2008, pp. 2600–2603.

- [5] T. F. Chan, S. H. Kang, Kang, and J. Shen, "Euler's elastica and curvature based inpainting," *SIAM J. Appl. Math.*, vol. 63, pp. 564–592, 2002.
- [6] A. Criminisi, P. Perez, and K. Toyama, "Object removal by exemplar-based inpainting," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 2, June 2003, pp. II–721 – II–728 vol.2.
- [7] —, "Region filling and object removal by exemplar-based image inpainting," *Image Processing, IEEE Transactions on*, vol. 13, no. 9, pp. 1200–1212, Sept 2004.
- [8] H. Grossauer, "A combined PDE and texture synthesis approach to inpainting," in *Computer Vision - ECCV 2004*, ser. Lecture Notes in Computer Science, T. Pajdla and J. Matas, Eds. Springer Berlin Heidelberg, 2004, vol. 3022, pp. 214–224.
- [9] A. Z. A. Levin and Y. Weiss, "Learning how to inpaint from global image statistics," vol. 1, pp. 305–312.
- [10] V. C. A. Bugeau, M. Bertalmio and G. Sapiro, "A comprehensive framework for image inpainting," *IEEE Trans. Image Process, TIP*, vol. 19, no. 10, pp. 2634–2645, 2010.
- [11] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, "Filling-in by joint interpolation of vector fields and gray levels," *Image Processing, IEEE Transactions on*, vol. 10, no. 8, pp. 1200 –1211, aug 2001.
- [12] Z.-N. L. Zinovi Tauber and M. S. Drew, "Review and preview: Disocclusion by inpainting for image-based rendering," *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, vol. 37, no. 4, pp. 527–540, 2007.
- [13] T. K. Shih and R.-C. Chang, "Digital inpainting - survey and multilayer image inpainting algorithms," *Third International Conference on Information Technology and Applications, ICITA*, vol. 1, pp. 15–24, 2005.
- [14] V. LLC, "Visage lab face retouch." [Online]. Available: <http://app.lk/visage-lab>
- [15] U. Video, "Beautify." [Online]. Available: <https://itunes.apple.com/us/app/beautify/id434842935?mt=8>
- [16] ArcSoft, "Perfect365." [Online]. Available: <http://perfect365.arcsoft.com>
- [17] T. Georgiev, "Image reconstruction invariant to relighting," *EUROGRAPHICS*, 2005.
- [18] A. B. P. Prez, M. Gangnet, "Poisson image editing," *Proceedings of ACM SIGGRAPH*, pp. 313–318.
- [19] L.-Y. Wei and M. Levoy, "Fast texture synthesis using tree-structured vector quantization," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '00, 2000, pp. 479–488.
- [20] A. Efros and T. Leung, "Texture synthesis by non-parametric sampling," in *8th International Conference on Computer Vision*, 1999, pp. 1033–1038.
- [21] N. Batool and R. Chellappa, "Modeling and detection of wrinkles in aging human faces using marked point processes," in *ECCV Workshops (2)*, 2012, pp. 178–188.
- [22] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '01. New York, NY, USA: ACM, 2001, pp. 341–346.
- [23] A. Rao and R. Jain, "Computerized flow field analysis: oriented texture fields," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 14, no. 7, pp. 693 –709, jul 1992.
- [24] F. J. Ayres and R. M. Rangayyan, "Design and performance analysis of oriented feature detectors," *Journal of Electronic Imaging*, vol. 16, no. 2, pp. 023 007–023 007–12, 2007.
- [25] P. Orbanz and J. Buhmann, "SAR images as mixtures of Gaussian mixtures," in *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, vol. 2, 2005, pp. II–209–12.
- [26] T. Nguyen and Q. Wu, "Fast and robust spatially constrained Gaussian mixture model for image segmentation," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 23, no. 4, pp. 621–635, 2013.
- [27] S. Sanjay-Gopal and T. Hebert, "Bayesian pixel classification using spatially variant finite mixtures and the generalized EM algorithm," *Image Processing, IEEE Transactions on*, vol. 7, no. 7, pp. 1014–1028, 1998.
- [28] W.-C. Lin, J. Hays, C. Wu, Y. Liu, and V. Kwatra, "Quantitative evaluation of near regular texture synthesis algorithms," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1, June 2006, pp. 427–434.
- [29] Z. AlMeraj, C. S. Kaplan, and P. Asente, "Towards effective evaluation of geometric texture synthesis algorithms," in *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*, ser. NPAR '13. New York, NY, USA: ACM, 2013, pp. 5–14.
- [30] C. Summers, "Not dressed as lamb," 2011–14. [Online]. Available: <http://www.notdressedaslamb.com/2012/11/musings-monday-my-40-year-old-face-uber.html#.U37LOvldWSr>
- [31] X. Li, "Image recovery via hybrid sparse representations: A deterministic annealing approach," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 5, pp. 953–962, Sept 2011.



Nazre Batool Nazre Batool received her B.Sc. (Hons.) degree in Electrical Engineering from the University of Engineering and Technology, Lahore, Pakistan in 2004, M.Sc. (Hons.) degree in Electrical and Electronics Engineering from Universiti Teknologi PETRONAS, Tronoh, Malaysia in 2008 and Ph.D. degree in Electrical Engineering from the University of Maryland, College Park, USA in 2014. Ms. Batool was awarded the university merit scholarship for her Bachelor's degree and graduate fellowship for her Master's degree. She is also a recipient of the 5 year Fulbright PhD scholarship from Pakistan. She joined AYIN team at INRIA Sophia-Antipolis as a post-doc researcher in February, 2014. Her research interests include computer vision, pattern recognition, stochastic image modeling, skin imaging and biometrics.



Rama Chellappa Rama Chellappa received the B.E. (Hons.) degree from University of Madras, India, in 1975 and the M.E. (Distinction) degree from Indian Institute of Science, Bangalore, in 1977. He received M.S.E.E. and Ph.D. Degrees in Electrical Engineering from Purdue University, West Lafayette, IN, in 1978 and 1981 respectively. Since 1991, he has been a Professor of Electrical Engineering and an affiliate Professor of Computer Science at University of Maryland, College Park. He is also affiliated with the Center For Automation Research (Director) and the Institute for Advanced Computer Studies (Permanent Member). In 2005, he was named a Minta Martin Professor of Engineering. Prior to joining the University of Maryland, he was an Assistant (1981-1986) and Associate Professor (1986- 1991) and Director of the Signal and Image Processing Institute (1988-1990) at University of Southern California, Los Angeles. Over the last 29 years, he has published numerous book chapters, peer-reviewed journal and conference papers. He has co-authored and edited books on MRFs, face and gait recognition and collected works on image processing and analysis. His current research interests are face and gait analysis, markerless motion capture, 3D modeling from video, image and video-based recognition and exploitation, compressive sensing, and hyper spectral processing. Prof. Chellappa served as the associate editor of four IEEE Transactions, as a Co-Editor-in-Chief of Graphical Models and Image Processing and as the Editor-in-Chief of IEEE Transactions on Pattern Analysis and Machine Intelligence. He served as a member of the IEEE Signal Processing Society Board of Governors and as its Vice President of Awards and Membership. He is serving a two-year term as the President of IEEE Biometrics Council. He has received several awards, including an NSF Presidential Young Investigator Award, four IBM Faculty Development Awards, an Excellence in Teaching Award from the School of Engineering at USC, two paper awards from the International Association of Pattern Recognition. He received the Society, Technical Achievement and Meritorious Service Awards from the IEEE Signal Processing Society. He also received the Technical Achievement and Meritorious Service Awards from the IEEE Computer Society. At University of Maryland, he was elected as a Distinguished Faculty Research Fellow, as a Distinguished Scholar- Teacher, received the Outstanding Faculty Research Award from the College of Engineering, an Outstanding Innovator Award from the Office of Technology Commercialization and an Outstanding GEMSTONE Mentor Award. He is a Fellow of the IEEE, the International Association for Pattern Recognition and Optical Society of America. He has served as a General Technical Program Chair for several IEEE international and national conferences and workshops. He is a Golden Core Member of the IEEE Computer Society and served a two-year term as a Distinguished Lecturer of the IEEE Signal Processing Society.