SYNTHESIS OF IRIS IMAGES USING MARKOV RANDOM FIELDS

Sarvesh Makthal and Arun Ross

Lane Department of Computer Science and Electrical Engineering
West Virginia University, Morgantown, WV, 26506 USA
{makthal, ross}@csee.wvu.edu

ABSTRACT

Of all the physiological traits of the human body that help in personal identification, the iris is probably the most robust and accurate. A number of iris recognition algorithms have been proposed in the literature over the past few years; however, not all of them have been tested on large databases. The largest known iris database has about 350,000 images in it but is proprietary. In this paper, a synthetic iris generation method based on Markov Random Field (MRF) modeling is proposed. The synthesis procedure is deterministic and avoids the sampling of a probability distribution and is, therefore, computationally simple. Furthermore, it is shown that iris textures in general are significantly different from other non-stochastic textural patterns. Clustering experiments indicate that the synthetic irises generated using the proposed technique are similar in content to real iris images.

1. INTRODUCTION

The iris is probably the only internal organ of the body that is readily visible from the outside. Its purpose is to control the amount of light that enters the eye through the pupil by using the dilator and sphincter muscles that govern the pupil size. The elastic fibrous tissue gives the iris a very complicated texture pattern. There is little evidence that the structure of the iris changes over a person's life. Daugman [1] has shown that an iris pattern has about 250 degrees of freedom, i.e., the probability of two eyes having the same iris texture is about 1 in 7 billion. Even the 2 irises of an individual are different thereby suggesting that iris textures are independent of the genetic constitution of an individual. These observations have made iris a very popular biometric.

Numerous algorithms have been proposed for using iris as a biometric. Daugman [1] describes a system that uses Gabor transforms to extract the textural content of an iris image. Wildes [2] employs the Laplacian of a Gaussian (LOG) filter to extract features from the iris image. Noh et al. [3] make use of multiresolution Independent Component Analysis (ICA) to generate discriminating features. Though most iris recognition algorithms claim a very low false accept rate (FAR), only Daugman's algorithm has been tested on a large database ($\sim$ 350,000 iris images). The Chinese Academy of Sciences Institute of Automation (CASIA) has made available a public database which is modest in size comprising of 756 images pertaining to 108 users [4]. With the introduction of new iris recognition algorithms (e.g.,[2], [5], [6], [7]), there is a pronounced need for a large public database to test and compare multiple techniques. One way to address this need is to devise synthetic iris generation schemes to construct large databases which can then be used by algorithm developers to test their techniques on. The use of synthetic databases in biometrics is not new and has been previously studied in the context of fingerprints and face biometrics. Orlans et al. [8] have outlined the advantages of studying synthetic biometric capabilities which include (i) the development of parametric models that permit the *testing* of a biometric system under various conditions; (ii) the mitigation of *privacy* concerns typically associated with real-world data; (iii) the *efficiency* (with respect to cost and time) in assembling a large database representing a variety of intra-class variations; and (iv) the design of statistical procedures to *predict* performance in large-scale systems. Also, an analysis-by-synthesis approach helps researchers gain insight into the individuality of biometric patterns.

In the fingerprint literature, a technique for generating synthetic fingerprints was proposed by Cappelli et al [9]. The resulting software known as SFINGE has been used to generate artificial databases that have been incorporated in multiple editions of the Fingerprint Verification Contest (FVC2000, FVC2002, FVC2004 [10]). The authors observe that the performance of competing algorithms on these synthetic databases, is comparable with their performance on real datasets, thereby suggesting the ability of SFINGE to realistically model intra-class and inter-class dynamics. Similarly, the FaceGen software [11] along with 3D Studio Max and Viisage FaceTools has been used by Orlans et al. [12] to devise test protocols for face recognition.

There are, however, very few literature precedents discussing synthetic generation of iris images. Wang et al. [13] recently used the Principal Component Analysis(PCA) method to generate synthetic irises. The eigen-coefficients obtained from a training set of real iris images are modified in a controlled manner; the eigenbasis is then used to generate synthetic images using these new coefficients. Proper modification of these coefficients results in new classes of iris texture. Lefohn et al. [14] create renditions of the human iris by stacking several transparent layers with each layer describing a certain iris component (such as stroma, collarette, sphincter, etc.).

In this paper we propose a technique for generating synthetic iris images using Markov Random Field (MRF) models. The synthesis procedure uses single or multiple *primitives* to generate iris-like patterns. The validity of the generated images are confirmed via two experiments: (a) by a clustering process to distinguish iris images (real and synthetic) from non-iris textural patterns, and (b) by examining the genuine and impostor match score distributions of the synthetic images.

2. SYNTHESIS USING MARKOV RANDOM FIELDS

The iris exhibits a very rich texture consisting of "pectinate ligaments adhering into a tangled mesh revealing striations, ciliary processes, crypts, rings, furrows, a corona, sometimes freckles, vasculature, and other features" [15]. It is the randomness (and apparent stability) of the inherent textural structure that renders iris as a useful biometric (Figure 1).

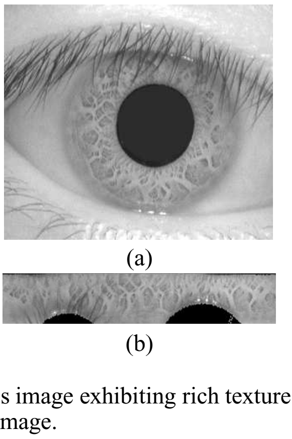


Figure 1: (a) An iris image exhibiting rich texture, and (b) the associated unwrapped image.

The task of texture synthesis, in general, can be divided into 2 major processes: (a) texture modeling, that characterizes the stochastic and structural properties of a sample, and (b) model sampling, to generate novel texture patterns described by the model. The model is usually a probability distribution that defines the spatial relationship between groups of pixels. Model sampling, in such a case, entails determining the pixel values for the new texture based on this probability distribution. In this work, the textural intricacy of the iris is captured via a Markov Random Field (MRF) model. MRFs have been used to successfully model a wide variety of textures [16, 17]. MRF has also been used in biometrics for synthesizing face-like images [18]. MRF models are used to describe the probability distribution governing the intensity values of pixels in a specific neighborhood also known as a *clique*.

The texture synthesis procedure described in this paper has been inspired by the work of Wei et al [19]. The technique uses texture *primitives* to guide the synthesis process. The pixels of a randomly initialized image are iteratively updated until an iris-like structure emerges. The proposed technique is deterministic in that it does not sample a probability distribution but rather observes neighborhood values in the primitive image to determine the value of the pixels in the synthetic image. The algorithm therefore synthesizes a texture image, that is *locally* similar to the primitive image but *globally* exhibits a different structure. While a single primitive is sufficient for the generating algorithm, two or more primitives may also be used. The use of multiple primitives may be necessary because (i) a typical iris exhibits rich texture in the immediate vicinity of the pupil which tapers away in intensity as one moves away from the pupil; and (ii) the blending of multiple primitives increases the randomness of the synthesized image. In the following subsections we describe synthesis based on single as well as multiple primitives.

2.1 Synthesis Using a Single Primitive

Let the primitive sample be denoted as I_p and let I_s represent the initial random noise image. For a pixel r in I_s, let $N_s(r)$ represent its neighborhood as shown in Figure 2. Similarly, let $N_p = \{N_p(r)\}$ be the set of *all* such neighborhoods in I_p. In our formulation, $N_s(r)$ is compared with individual elements of N_p using the Euclidean distance metric. The purpose of this exercise is to determine that neighborhood in I_p that closely resembles the current neighborhood of r in I_s. The pixel value $I_s(r)$ is then updated based on this comparison as $I_s(r) = I_p(t^*)$ where $t^* = \arg\min_{t \in I_p} |N_s(r) - N_p(t)|$ and $|.|$ is the L_2 norm. This process is repeated iteratively in a raster scan fashion for all pixels in I_s until the desired texture is generated. Note that more sophisticated comparison techniques other than the L_2 norm may be used to compare the neighborhood intensities of the two images; however, experiments suggest that the L_2 norm suffices for our purpose.

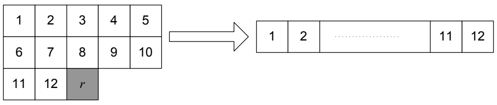
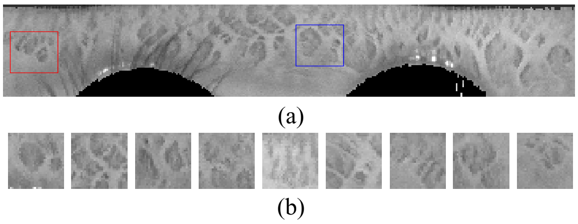


Figure 2: The neighborhood structure, $N_s(r)$, of pixel r.

2.1.1 *Choice of Neighborhood*

Since the neighborhood effectively decides the quality of the synthesized texture, its size and shape play an important role. For any regular repetitive texture pattern, the size should be at least equal to the largest regular structure present in the primitive pattern. But for the generation of iris images, which are non-homogenous, this constraint can be relaxed. One has to ensure, however, that the input texture block displays a good amount of 'random texture' elements. To facilitate this, images from the CASIA database [4] have been used to obtain primitive patterns in our experiments. Figure 3(a) shows two 30×30 blocks on an arbitrarily chosen iris image; Figure 3(b) shows some of the sample textures that have been selected from different images for the synthesis process. The neighborhoods were chosen based on the visual appearance of the iris texture, and their size varied anywhere between 9×9 pixels to 23×23 pixels in our experiments. Note that any other primitive, demonstrating iris-like properties, can be used.


(a)

(b)

Figure 3: (a) Selection of primitives from an actual iris image. The iris structure has been unwrapped and presented as a rectangular entity rather than a circular one. (b) Examples of sample primitives used in our experiments.

The shape of the neighborhood is made 'causal', i.e. the neighborhood will only include pixels that have already been generated during the raster scan. This ensures that every output pixel is updated based on previously generated pixels and not from random noise. This causal property of the neighborhood accelerates the convergence speed of the algorithm. Only the first few pixels of the output image use white noise as their neighborhoods in the first iteration, but subsequently, all pixels will use neighborhoods that have already been visited in an earlier pass.

The boundary pixels will not contain the full specified neighborhood. To circumvent this problem we assume the image to be toroidal in nature. Hence the pixel (x,y) can be mapped to $(x \bmod R, y \bmod C)$, where R and C are the rows and columns of the image, respectively. If the neighborhoods of boundary pixels are not handled properly, the output image will appear tiled and have a blocky appearance. The image synthesis algorithm is summarized in Figure 4.

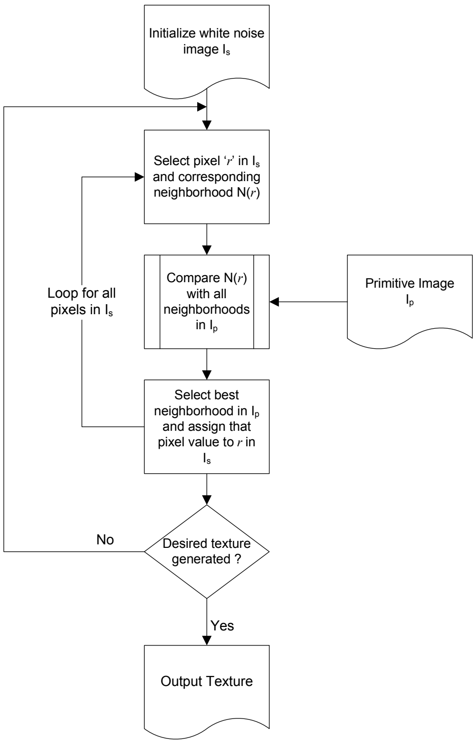

Initialize white noise image I_s
Select pixel 'r' in I_s and corresponding neighborhood N(r)
Compare N(r) with all neighborhoods in I_p
Primitive Image I_p
Loop for all pixels in I_s
Select best neighborhood in I_p and assign that pixel value to r in I_s
No
Desired texture generated ?
Yes
Output Texture

Figure 4: Flowchart summarizing the texture synthesis procedure.

2.2 Synthesis Using Multiple Primitives

In section 2.1 we described the generation of a texture based on a single primitive. This technique is useful for generating textures with repetitive patterns (i.e., primitives). Iris texture is, however, not repetitive in nature and exhibits significant randomness. To incorporate this randomness, we propose the use of more than one type of primitive for the synthesis process. Blending multiple primitives will result in a very unique and random structure.

The algorithm takes a set of source iris textures $\{I_{p_1}, I_{p_2}, \ldots, I_{p_k}\}$ as primitives with weight image W_j associated with primitive I_{p_j}. The weight image (whose size is the same as the desired synthetic image) specifies the importance of a primitive when determining the value of a pixel in the synthetic texture I_s. A real iris does not contain the same textured pattern all over the image. Instead, it exhibits significant variations across the image. For example, the region of the iris closer to the pupil, has very closely packed radial structures (*radial furrows*), spreading out from the edge of the pupil. Moving outward, the radial patterns begin to merge with large blob like structures, referred to as *crypts*. The outer region of the image (closer to the sclera) has limited texture information and is known as the *limbus*. The primitives are selected based on these observations of a true iris image; hence, three primitives are used, rather than just one, to incorporate the true structure of an iris into the synthetic image.

Figure 5(a) shows 3 primitives - I_{p_1}, I_{p_2}, I_{p_3} - representing different regions of the iris. The weighting scheme (W_1, W_2, W_3) for each of these primitives is shown in figure 5(b). Since the *crypts* appear only intermittently in the iris image, we randomly select locations in I_s to incorporate those primitives and accordingly the weight W_2 is generated. To incorporate randomness in the synthesis procedure, we treat the weight values as probability measures when updating pixel values in I_s. Thus, to update pixel r in I_s we first compare $N_s(r)$ with all neighborhoods in the 3 primitives using the L_2 norm. For each primitive, I_{p_j}, the best neighborhood, $N_{p_j}(t_j)$ is reported. Then $I_s(r)$ is assigned the value $I_{p_j}(t_j)$ with a probability $W_j(t_j)$. This ensures that sufficient randomness is introduced in the synthetic image.

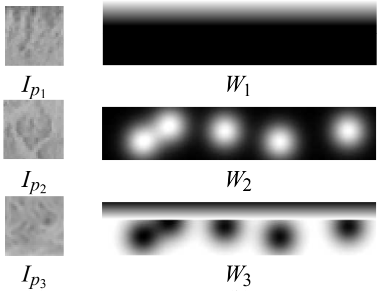


Figure 5: Three primitives representing the *radial furrows*, *crypts* and *limbus*. (a) Primitives, (b) Corresponding weights.

3. RESULTS OF SYNTHESIS

To demonstrate the efficacy of the proposed algorithm we manually extract 3 primitives from each of the 108 users in the CASIA database. A total of 324 separate blocks are thus extracted and, hence, the maximum number of classes (i.e., users) that can be generated using this technique would be 108^3. By combining more than three texture primitives or by altering the weight images, this number can be exponentially increased. The synthetic patterns are generated as rectangular entities and are subsequently mapped onto a circular structure to create iris-like images. Also, tree-structured vector quantization (TSVQ) is used to accelerate the comparison of a pixel neighborhood in the synthetic image with all neighborhoods in the primitive patterns. Figures 6 and 7 show a few results of the synthesis process.

3.1 Generating Multiple Samples Per User Class

In real world scenarios, multiple acquisitions of the same iris do not result in the same image. Variations are caused due to rotation (cyclovergence or tilting of the head), elastic deformation (due to pupil dilation) and changes in camera position, focus or zoom factor. We incorporate these factors in the synthesis process to generate multiple images of the same class.

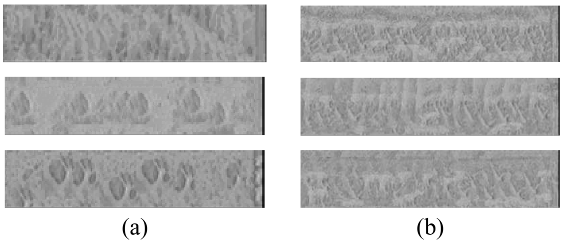


Figure 6: Synthesis using (a) a single primitive and (b) multiple primitives.

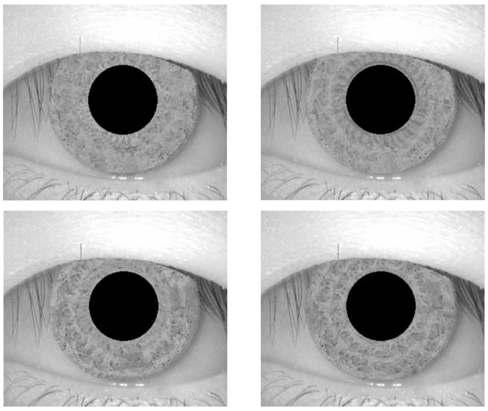


Figure 7: Synthesis results using 3 primitives.

The synthetic image (rectangular) is cyclically rotated in order to incorporate the rotation parameter. Translation is introduced by shifting the circular iris along the x- and y-axis by a few pixels. To include distortion due to the camera motion, perspective transformations are used [20]. If x_u and y_u are the undistorted co-ordinates and x_d and y_d are the co-ordinates after distortion, then the lens distortion model can be written as $x_u = x_d(1+kr_d^2)$ and $y_u = y_d(1+kr_d^2)$, where $r_d = \sqrt{x_d^2+y_d^2}$ is the distorted radius. The inverse distortion model is then given as $r_u = r_d(1+kr_d^2)$ where $r_u = \sqrt{x_u^2+y_u^2}$ is the undistorted radius. Solving for x_d and y_d we get $x_d = x_u\frac{r_d}{r_u}$ and $y_d = y_u\frac{r_d}{r_u}$. This model represents a simple distortion of the lens. Multiple k values (0.02, 0.03, 0.04, 0.05) were used in our experiments.

4. VALIDATING SYNTHETIC IRISES

Iris images have distinct texture properties when compared against other natural textures. A simple form of clustering using the k-means clustering method is sufficient to show that iris images are significantly distinct from other irregular texture images. We studied the clustering property of real and synthetic iris images relative to other textural patterns present in the Brodatz library. Since iris images are non-homogeneous and non-repetitive in nature, we choose only the non-stochastic textures present in the Brodatz library.

Some of the well known features for texture classification and analysis are the ones derived from the co-occurrence matrix of the pattern [16]. The spatial gray level co-occurrence matrix, C_d, of an image tries to capture the second order statistics of the image. The entry (i,j) of C_d denotes the number of pixel pairs separated by a distance $\mathbf{d}$=(d_x,d_y) and having gray levels i and j. We compute various properties of the covariance matrix in order to understand the texture structure of the underlying image: (a) Energy = $\sum_i \sum_j C_d^2(i,j)$; (b) Entropy = $-\sum_i \sum_j C_d(i,j)log(C_d(i,j))$; (c) Contrast = $_i \;\; _j (i-j)^2 C_d^2(i,j)$; and (d) Homogeneity = $_i \;\; _j \frac{C_d(i,j)}{1+|i-j|}$.

There are no well defined methods for selecting the displacement vector **d**. We consider multiple displacements by moving in steps of 5 and 1 along the x- and y-axis, respectively. For a (20×360) image we get 111 co-occurrence matrices. Thus a feature vector of length $111 \times 4 = 444$ is derived for each image.

These features were computed for 100 images in the CASIA dataset, 100 images in the synthesized dataset and 20 non-stochastic images in the Brodatz library. The *k-means* clustering technique was used to organize these images into 2 categories. In our clustering experiment, 97% of the real iris images and 100% of the synthetic iris images were classified into one group, and 75% of the Brodatz textures and 3% of the real iris images were classified into a separate group. This suggests that (a) the iris texture is distinct from other non-stochastic textural patterns, and (b) the synthetic iris images have similar textural content as real iris images.

We also generated the genuine and impostor match score distributions of the synthetic iris images (100 classes, 4 samples per class) by extracting features using a wavelet approach and comparing the features using the Manhattan distance (using Daugman's algorithm [1] with the Hamming distance will result in a different set of distributions). Figure 8 compares these distributions with those associated with real iris images (100 classes, 4 samples per user). The impostor distribution of the synthetic data is observed to be multimodal. This is not desirable and we are currently looking at ways to address this problem.

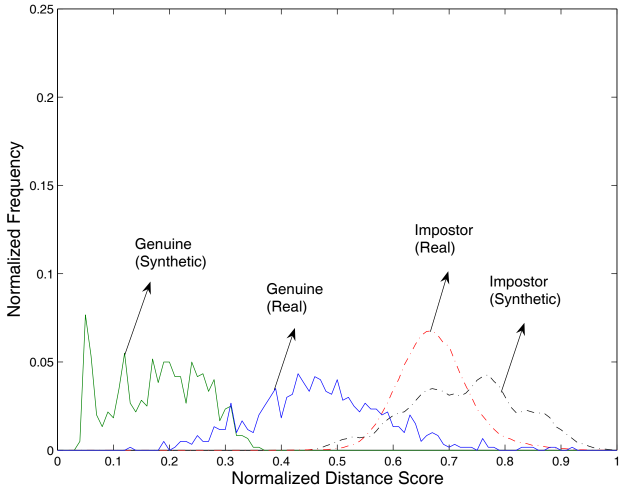


Figure 8: The genuine and impostor match score distributions for real and synthetic iris images. A simple wavelet-based algorithm was used to generate features. The features were compared using the Manhattan distance.

5. SUMMARY AND FUTURE WORK

We have presented a novel technique for synthesizing iris images by characterizing the texture using a Markov Random Field. The proposed technique uses primitive textural patterns to guide the synthesis of iris images from a random noise image. Tree structured vector quantization is used to accelerate the synthesis procedure. Clustering experiments using feature vectors extracted from co-occurrence matrices show that the textural content of both synthetic and real iris textures are very similar. Presently, the synthesis algorithm terminates after 10 iterations. We are looking at ways to dynamically determine the number of iterations that is required. We are also examining the possibility of developing individuality models for iris patterns based on the technique presented in this paper.

REFERENCES

[1] John G. Daugman, "High confidence visual recognition of persons by a test of statistical independence," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1148–1160, Nov 1993.

[2] Richard P. Wildes, "Iris recognition: An emerging biometric technology," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1348–1363, Sept 1997.

[3] Kwanghyuk Bae, Seungin Noh, and Jaihei Kim, "Iris feature extraction using independent component analysis," *Proc. 4^{th} Intl. Conf. on Audio and Video Based Biometric Person Authentication (AVBPA 2003)*, pp. 838–844, 2003.

[4] "The CASIA iris image database," Available at http://www.sinobiometrics.com.

[5] Li Ma, Yunhong Wang, and Tieniu Tan, "Personal identification based on iris texture analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1519–1533, Dec 2003.

[6] W.W. Boles and B. Boashash, "A human identification technique using images of the iris and wavelet transform," *IEEE Trans. on Signal Processing*, vol. 46, no. 4, pp. 1185–1188, Apr 1998.

[7] Li Ma, Tieniu Tan, and Yunhong Wang, "Efficient iris recognition by characterizing key local variations," *IEEE Trans. on Image Processing*, vol. 13, no. 6, pp. 739–750, June 2004.

[8] Nick M. Orlans, D.J. Buettner, and J. Marques, "A survey of synthetic biometrics: Capabilities and benefits," *Intl. Conf. on Artificial Intelligence*, 2004 (To appear).

[9] R. Cappelli, D. Maio, D. Maltoni, and A. Erol, "Synthetic fingerprint-image generation," *Proc. of the 15^{th} Intl. Conf. on Pattern Recognition*, vol. 3, pp. 471–474, 2000.

[10] "University of Bologna: Biometric systems laboratory," Available at: http://bias.csr.unibo.it/research/biolab/sfinge.html.

[11] "The FaceGen software," Available at: http://www.facegen.com/.

[12] Nick M. Orlans, Alan T. Piszcz, and Richard J. Chavez, "Parametrically controlled synthetic imagery experiment for face recognition testing," in *Proc. of the 2003 ACM SIGMM workshop on Biometrics methods and applications*, Berkley, CA, 2003, pp. 58–64.

[13] Jiali Cui, Yunhong Wang, JunZhou Huang, Tieniu Tan, and Zhenan Sun, "An iris image synthesis method based on PCA and super-resolution," *Proceedings of the 17^{th} International Conf. on Pattern Recognition*, vol. 4, pp. 471–474, Aug 2004.

[14] Aaron Lefohn, Brian Budge, Peter Shirley, Richard Caruso, and Erik Reinhard, "An ocularist's approach to human iris synthesis," *IEEE Computer Graphics and Applications*, vol. 23, no. 6, pp. 70–75, Nov/Dec 2003.

[15] John Daugman, "Demodulation by complex-valued wavelets for stochastic pattern recognition," *International Journal of Wavelets, Multi-resolution and Information Processing*, vol. 1, no. 1, pp. 1–17, 2003.

[16] L.F. Pau and P.S.P. Wang, *The Handbook of Pattern Recognition and Computer Vision*, chapter 2.1, pp. 207–248, World Scientific Publishing Co., 2^{nd} edition, 1998.

[17] Stan Z. Li, *Markov Random Field Modeling in Image Analysis*, Springer-Verlag, 2001.

[18] Anil K. Jain, Sarat C. Dass, and Xiaoguang Lu, "Face detection and synthesis using markov random field models," *Proc. International Conference on Pattern Recognition*, Aug 2002.

[19] Li-Yi Wei and Mark Levoy, "Fast texture synthesis using tree-structured vector quantization," *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 479–488, 2000.

[20] Frederic Devernay and Olivier Faugeras, "Straight lines have to be straight," *Machine Vision and Applications*, vol. 13, no. 1, pp. 14–24, 2001.