DESIGN OF MULTIPLE GABOR FILTERS FOR TEXTURE SEGMENTATION

Thomas P. Weldon[1] *William E. Higgins*[2]

[1]University of North Carolina, Department of Electrical Engineering, Charlotte, NC
[2]Penn State University, Department of Electrical Engineering, University Park, PA

ABSTRACT

This paper presents a method for the design of multiple Gabor filters for segmenting multi-textured images. Although design methods for a single Gabor filter have been presented recently, the development of general multi-filter multi-texture design methods largely remains an open problem. Previous multi-filter design approaches required one filter per texture or were constrained to pairs of textures. Other approaches employed *ad hoc* banks of Gabor filters for texture segmentation, where the parameters of the constituent filters were restricted to fixed values and were not necessarily tuned for a specific texture-segmentation problem. The proposed method removes these restrictions on the number of filters and the number of textures. This offers the potential to improve the segmentation performance or to reduce the number of filters. Further, the development of the design method and mathematical models provide new insight into the design of multiple Gabor filters for texture segmentation. Results are presented that confirm the efficacy of our filter-design method and support underlying mathematical models.

1. INTRODUCTION

[1] The segmentation of textured images remains a difficult problem in image processing. One successful approach to texture segmentation employs Gabor filters [1–3]. The motivation for using these filters in texture segmentation is discussed in the references [4, 5].

A central issue in applying Gabor filters to texture segmentation is the determination of the filter parameters. Past efforts have employed (1) a large bank of *ad hoc* selected Gabor filters with predetermined parameters or (2) a small set of Gabor filters with filter parameters designed to solve a particular problem [6–10].

In approach (1), some segmentation tasks may not require a bank of filters or may not tolerate the computational burden imposed by a large filter bank. In approach (2), an optimal method for designing multiple (≥ 2) Gabor filters has not been reported for the general $\mathcal{N}$-texture ($\mathcal{N} \geq 2$) segmentation problem. In addition, previous approaches have had limitations on the number of textures or on the number of filters per texture [9, 10].

This paper addresses the design of multiple Gabor filters for segmenting multi-textured (≥ 2) images. The proposed design procedure is a supervised method; i.e, we assume that representative texture samples are given, as is often the case in real segmentation problems [7, 9, 10]. We propose a near-optimal method for the design of multiple Gabor filters to segment multiple textures. The scheme is based on the development of an underlying mathematical model that relates the texture power spectra, filter parameters, and segmentation error [11]. The remainder of this paper outlines the scheme and provides some representative results.

2. MULTI-FILTER SCHEME

The multichannnel scheme is shown in Fig. 1. We will refer to a single cascade of a Gabor prefilter $h_j(x,y)$, magnitude operator, and Gaussian postfilter $g_{p_j}(x,y)$ as a *filter channel*. The multichannel architecture is essentially a replication of the single-filter architecture over the k channels. However, the values of the filter parameters $(u_j, v_j, \sigma_{g_j}, \sigma_{p_j})$ in each channel j are free to vary from channel to channel.

The input image $i(x,y)$ is assumed to be composed of disjoint regions of $\mathcal{N}$ textures $t_1, t_2, \ldots, t_{\mathcal{N}}$ with $\mathcal{N} \geq 2$. First, the input image $i(x,y)$ is filtered using a bandpass *Gabor prefilter* with impulse response $h_j(x,y)$, where the subscript j denotes the particular channel in Fig. 1 and $1 \leq j \leq k$:

$$h_j(x,y) \quad = \quad \frac{1}{2\pi\sigma_{g_j}^2}\, e^{-\frac{(x^2+y^2)}{2\sigma_{g_j}^2}} \quad e^{-j2\pi(u_j x + v_j y)} \,. \qquad (1)$$

The *Gabor function* $h_j(x,y)$ is a complex sinusoid centered at frequency (u_j, v_j) and modulated by a Gaussian envelope [10]. The parameter σ_{g_j} determines the scale of the envelope of $h_j(x,y)$. Thus, parameters (u_j, v_j, σ_{g_j}) completely determine the Gabor prefilter. For simplicity, we also assume that the Gaussian envelope of $h_j(x,y)$ is a symmetric function. The output of the prefilter stage $i_{h_j}(x,y)$ is the convolution of the input image with the filter response

$$i_{h_j}(x,y) = h_j(x,y) ** i(x,y) \qquad (2)$$

where $**$ denotes convolution in two dimensions. In the notation for $i_{h_j}(x,y)$, the subscript h_j indicates the output of the Gabor prefilter $h_j(x,y)$ in the j th filter channel. The magnitude of the output of the Gabor prefilter is computed in the following stage as

$$m_j(x,y) = |i_{h_j}(x,y)| = |\, h_j(x,y) ** i(x,y)\,|, \qquad (3)$$

where $m_j(x,y)$ has been shown to have approximately Rician statistics for filtered textures [6, 7]. A low-pass Gaussian postfilter $g_{p_j}(x,y)$ is applied to prefilter output $m_j(x,y)$ yielding the postfiltered image

$$m_{p_j}(x,y) \quad = \quad m_j(x,y) ** g_{p_j}(x,y) \qquad (4)$$

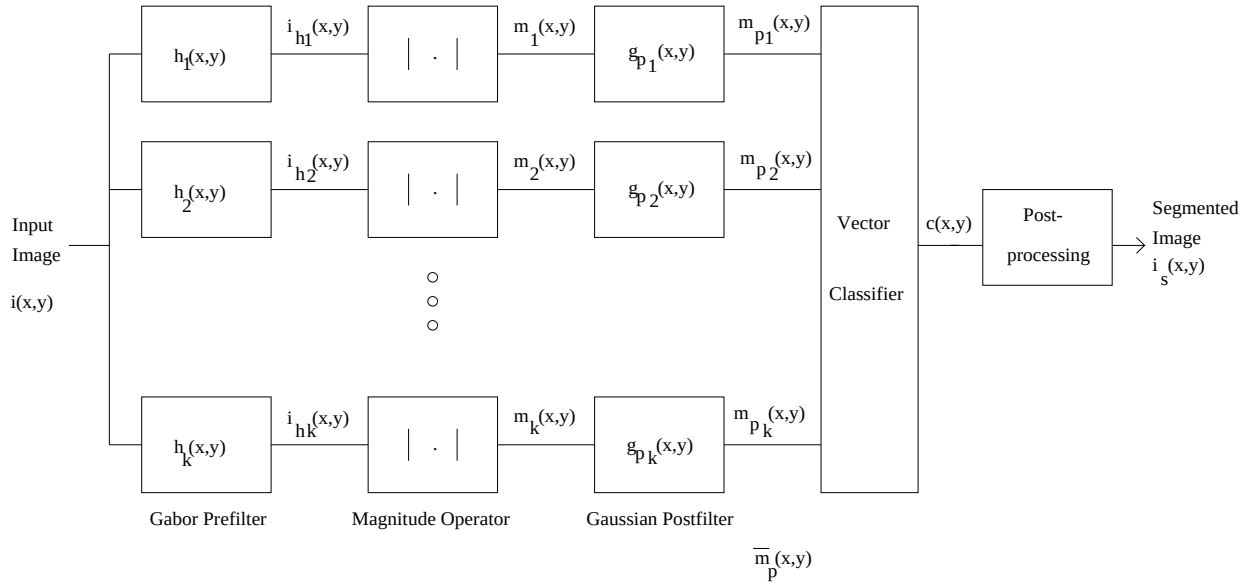

Figure 1. Multichannel scheme for segmenting a textured image.

with

$$g_{p_j}(x,y) \quad = \quad \frac{1}{2\pi\sigma_{p_j}^2}\, e^{-\frac{(x^2+y^2)}{2\sigma_{p_j}^2}} \tag{5}$$

and where the postfilter parameter σ_{p_j} determines the Gaussian postfilter in the j th channel. Generally, we will refer to $i_{h_j}(x,y)$ as the *prefiltered image*, $m_j(x,y)$ as the *prefilter output*, and $m_{p_j}(x,y)$ as the *postfilter output*.

A vector classifier generates the classified image $c(x,y)$ based on the vector output of the k postfilters. A Bayesian classifier based on predicted multivariate output statistics is used. Finally, provision is made for postclassification processing to address transient effects at boundaries between different textures. The result of the additional postprocessing of the classified image $c(x,y)$ gives the final segmented image $i_s(x,y)$.

3. DESIGN METHOD

Below, we describe portions of the underlying statistical models. Next, we describe an error measure that is used as the basis for the multi-filter design. The filter-design method is then outlined. Finally, the postprocessing is described.

3.1. Statistical Model

In previous research on the design of single filters, we presented a Gaussian statistical model for the postfiltered output $m_{p_j}(x,y)$ [7]. The success of this statistical model in predicting single-filter texture segmentation error and in developing single-filter algorithms leads us to consider a multivariate Gaussian model for the vector output statistics of a set of k filter channels.

Following the single-filter development in [7], the envelope $m_j(x,y)$ of the prefiltered image has a Rician distribution. The postfiltering operation in each of the k filter channels performs a spatial average of the prefilter output $m_j(x,y)$, leading to Gaussian distributions for the postfilter outputs $m_{p_j}(x,y)$. The multivariate Gaussian pdf of the filter-channel outputs for input texture t_i is then

$$p_i(\boldsymbol{m_p}) = \frac{1}{(2\pi)^{k/2}|\boldsymbol{C_i}\ |^{1/2}} \quad e^{-(\frac{(\boldsymbol{m_p}-\boldsymbol{\mu_i})^T \boldsymbol{C_i}^{-1}(\boldsymbol{m_p}-\boldsymbol{\mu_i})}{2})} \tag{6}$$

where $\boldsymbol{m_p}$ is a vector sample of the k-dimensional postfilter-output vector, $\boldsymbol{\mu_i}$ is the mean postfilter-output vector, $\boldsymbol{C_i}$ is the covariance matrix of the postfilter outputs. The parameters of each filter channel j are $\theta_j = (u_j, v_j, \sigma_{g_j}, \sigma_{p_j})$ corresponding to prefilter $h_j(x,y)$ and postfilter $g_{p_j}(x,y)$ in Fig. 1.

The components μ_{ij} of mean vector $\boldsymbol{\mu_i}$ can be determined for each texture t_i and each filter channel j using the single-filter methods of [7]. The covariance matrix $\boldsymbol{C_i}$ presents greater difficulty, since it implies a need for samples of the postfilter output for all candidate filters. However, the single-filter methods of [7] do yield the diagonal elements of the covariance matrix and these correspond to the variances $s_{p_{ij}}^2$ of individual channels in [7]. Thus, we propose using the values of $s_{p_{ij}}^2$ along the diagonal of $\boldsymbol{C_i}$ with all off-diagonal elements equal to zero. To reduce the likelihood of having strongly correlated features, the candidate filters are restricted such that the spatial-frequency responses of any of the constituent filters do not overlap significantly.

3.2. Segmentation Error Measure

The previous section established a multivariate *Gaussian* statistical model for the vector output of k filter channels. This suggests that the Bhattacharyya distance is an appropriate measure of feature performance [12]. Consider two textures $t_\alpha(x,y)$ and $t_\beta(x,y)$. The Bhattacharyya distance $B(t_\alpha, t_\beta)$, or *B-distance*, between the two textures is

$$B(t_\alpha, t_\beta) \quad = \quad \frac{1}{8}(\boldsymbol{\mu_\alpha} - \boldsymbol{\mu_\beta})^T \left[\frac{\boldsymbol{C_\alpha} + \boldsymbol{C_\beta}}{2}\right]^{-1} (\boldsymbol{\mu_\alpha} - \boldsymbol{\mu_\beta})$$

$$+ \quad \frac{1}{2} ln \left(\frac{\left| \frac{1}{2} (\boldsymbol{C_\alpha} + \boldsymbol{C_\beta}) \right|}{|\boldsymbol{C_\alpha}|^{1/2} |\boldsymbol{C_\beta}|^{1/2}} \right) \tag{7}$$

where $\boldsymbol{\mu_\alpha}$ and $\boldsymbol{\mu_\beta}$ are the mean vectors, and $\boldsymbol{C_\alpha}$ and $\boldsymbol{C_\beta}$ are the covariance matrices associated with the two textures. The B-distance provides an upper bound for the classification error $\mathcal{E}_c$ of the two textures. A similar upper error bound for $\mathcal{N}$ multivariate Gaussian classes is [13, 14]

$$\mathcal{E}_c < \sum_{\alpha=1}^{\mathcal{N}-1} \sum_{\beta=\alpha+1}^{\mathcal{N}} (\mathcal{P}_\alpha \mathcal{P}_\beta)^{1/2} \rho_{\alpha\beta} \text{ with } \rho_{\alpha\beta} = e^{-B(t_\alpha, t_\beta)} \tag{8}$$

where $\rho_{\alpha\beta}$ are the two-class Bhattacharyya coefficients. Equations (7) and (8) provide the relationship between the segmentation error and the multivariate Gaussian statistics of the vector output of the k filter channels. As in the single filter case, the *a priori* probabilities $\mathcal{P}_\alpha$ are taken to be equal.

In practice, the error measure in (8) is effective for multichannel filter design when the number of textures is small (< 4). However, the performance of the designed filters deteriorates rapidly as the number of textures increases. This deterioration is apparently caused by an overstatement of error in (8) as a larger number of textures crowd the feature space. To mitigate this problem, equation (8) is modified:

$$\mathcal{E}_c \quad < \quad \frac{1}{\mathcal{N}-1} \sum_{\alpha=1}^{\mathcal{N}-1} \sum_{\beta=\alpha+1}^{\mathcal{N}} (\mathcal{P}_\alpha \mathcal{P}_\beta)^{1/2} \rho_{\alpha\beta} \ \leq \frac{1}{2} \tag{9}$$

where the worst case upper bound on $\mathcal{E}_c$ becomes $\frac{1}{2}$.

The Bhattacharyya error $\mathcal{E}_c$ provides a measure for the accurate classification of texture within regions but does not directly address the problem of accurate localization of boundaries between regions. Since the texture-segmentation task involves the separation of an image into regions of differing texture, inaccuracies in boundary locations will necessarily contribute to the overall segmentation error. The critical function of the localization-error measure is to generate an error term that favors smaller spatial filter responses. To develop a simple and effective error measure, the problem is broken into two parts. First, a simple localization-error measure is proposed based on error in corners of rectangular regions. Then, the localization error at other types of region boundaries, or edges, is reduced by modifying the Bayesian decision surfaces. Alternatively, we have also been investigating methods such as relaxation labeling to reduce localization error near boundaries [11].

The total error measure $\mathcal{E}_t$ for filter selection is then the sum of the classification error $\mathcal{E}_c$ and the localization error $\mathcal{E}_l$:

$$\begin{aligned} \mathcal{E}_t \quad &= \quad \mathcal{E}_c + \mathcal{E}_l \\ &= \quad \frac{1}{\mathcal{N}-1} \sum_{\alpha=1}^{\mathcal{N}-1} \sum_{\beta=\alpha+1}^{\mathcal{N}} (\mathcal{P}_\alpha \mathcal{P}_\beta)^{1/2} \ e^{-B(t_\alpha, t_\beta)} \\ &+ \quad \frac{1}{k} \sum_{\gamma=1}^{k} \frac{2(\mathcal{N})(\sigma_{g\gamma}^2 + \sigma_{p\gamma}^2)}{N^2} \end{aligned} \tag{10}$$

where the image dimensions are $N \times N$ and the term $(\sigma_{g\gamma}^2 + \sigma_{p\gamma}^2)$ approximates the combined localization effects of the Gabor prefilter and Gaussian postfilter. The total error $\mathcal{E}_t$ is then used as the basis for designing the Gabor filters in the multichannel design.

3.3. Multichannel Design Algorithm

Drawing upon the single-filter design ideas [7], the multichannel design algoithm is described below. Refer to [11] for further detail.

1. Construct a large set of single candidate Gabor prefilters. A typical range of prefilter parameters would be $\sigma_g \in \{2, 4, 8, 16\}$, with candidate filter center frequencies spaced in proportion to filter bandwidths. This provides coverage of the frequency plane at each value of σ_g. Typical postfilter parameters would be $\sigma_p = 2\sigma_g$.
2. Efficiently compute the Gaussian statistics associated with each postfiltered output texture for each candidate filter using single-filter methods [7].
3. Design the first filter by selecting the single filter with the smallest error $\mathcal{E}_t$ from the candidate filter set. Select subsequent filters using a forward sequential filter selection and the vector error measure $\mathcal{E}_t$ [14].

3.4. Postprocessing

In practice, we have seen that a Bayesian vector classifier performs well within regions but poorly at boundaries [11]. We also have observed that localization error at these boundaries between regions is reduced by modifying the Bayesian classifier using a mixture density. This mixture density approach has the advantage that it is readily applied to the case of multivariate-Gaussian classes. The procedure is to first select the largest variances $s_{p_{ij}}^2$ along each feature axis, or filter output. Then, form a mixture covariance matrix $\boldsymbol{C_{max}}$, whose diagonal elements are the maximum variances for each feature axis. The multivariate form of the mixture distribution $p_{i\,mix}$ is then

$$p_{i\,mix}(\boldsymbol{m_p}) = \frac{1}{2}(p_i(\boldsymbol{m_p}, \boldsymbol{C_{max}}) + p_i(\boldsymbol{m_p}, \boldsymbol{C})) \ , \tag{11}$$

where $p_i(\boldsymbol{m_p}, \boldsymbol{C})$ is the original multivariate Gaussian pdf and $p_i(\boldsymbol{m_p}, \boldsymbol{C_{max}})$ is the multivariate Gaussian formed by taking the maximum diagonal elements of all the covariance matrices for the textures.

Another problem observed in practice is the appearance of narrow regions at the boundary between two textures that are misclassified as a third texture. These narrow misclassified regions appear to be caused by the trajectory of the feature vector as it makes the transition through feature space at the boundary. A two-step operation is used to remove these narrow misclassified regions. First, pixels whose neighborhood consists entirely of one texture class are left unchanged; otherwise, the pixel value is set to zero to indicate it is no longer assigned to any class. Then, the classified regions are propagated back into the unassigned regions based on the most common class within 8-neighborhoods.

Finally, alternative postprocessing approaches such as relaxation methods are the topic of ongoing research [11]. However, the present postprocessing methods serve to illustrate the effectiveness of the designed filters in a complete system.

4. RESULTS

Sample results using the new multichannel filter design algorithm, along with the proposed postprocessing, are shown in Figs. 2 and 3. The "Nat-5" image shown in Fig. 2 is composed of samples from the Brodatz texture album and has been used by previous investigators to test texture segmentation methods [1, 2, 15]. Fig. 3 shows the final segmented image using *only four* Gabor filters designed with the proposed method and the foregoing postprocessing. Jain and

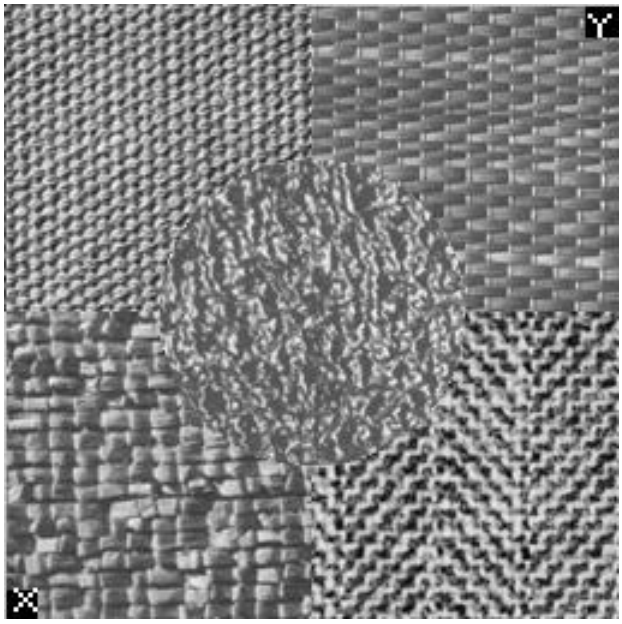

Figure 2. Input composite image "Nat-5", clockwise from top left: d77 "cotton canvas," d55 "straw matting," d17 "herringbone weave," d84 "raffia," d24 "pressed calf leather" in center.

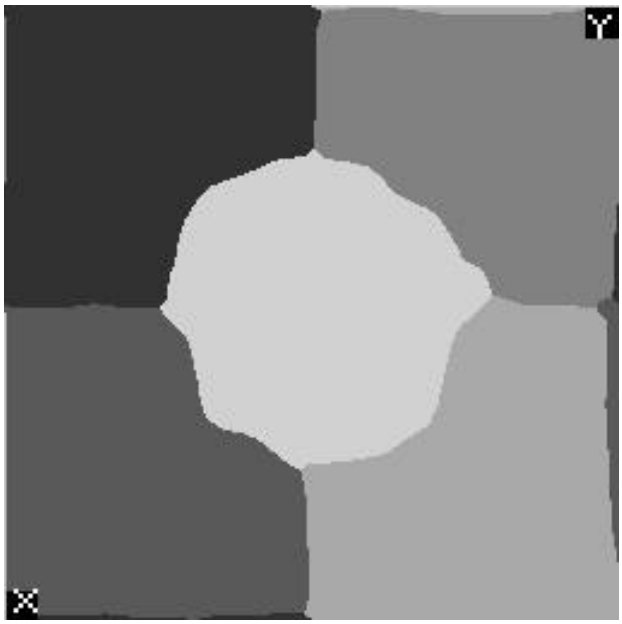

Figure 3. Segmented image using four filter channels, error=0.05.

Farrokhnia obtained similar segmentation results using 13 filters selected from a predetermined filter bank of 20 filters [1]. Randen and Husøy also achieved similar results using a subband-based approach using 13 to 40 filters [2]. Further results are presented in [11] that show similarly effective segmentations for combinations of three to eight textures. These results also show a tendency for increased segmentation error as the number of textures increases, and a tendency for decreased segmentation error as the number of filter channels increases.

REFERENCES

[1] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recognition*, vol. 23, no. 12, pp. 1167–1186, Dec. 1991.

[2] T. Randen and J. H. Husøy, "Multichannel filtering for image texture segmentation," *Optical Eng.*, vol. 33, no. 8, pp. 2617–2625, Aug. 1994.

[3] T. Chang and C. C. J. Kuo, "Texture analysis and classification with tree-structured wavelet transform," *IEEE Trans. Image Proc.*, vol. 2, no. 4, pp. 429–441, Oct. 1993.

[4] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *J. Opt. Soc. Amer. A*, vol. 2, no. 7, pp. 1160–1169, July 1985.

[5] D. Dunn, W. Higgins, and J. Wakeley, "Texture segmentation using 2-D Gabor elementary functions," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, no. 2, pp. 130–149, Feb. 1994.

[6] T. P. Weldon, W. E. Higgins, and D. F. Dunn, "Efficient Gabor filter design for texture segmentation," *Pattern Recognition*, vol. 29, no. 12, pp. 2005–2015, Dec. 1996.

[7] T. P. Weldon and W. E. Higgins, "Multiscale Rician approach to Gabor filter design for texture segmentation," in *IEEE Int. Conf. on Image Processing*, vol. II, (Austin, TX), pp. 620–624, 13-16 Nov. 1994.

[8] T. P. Weldon, W. E. Higgins, and D. F. Dunn, "Gabor filter design for multiple texture segmentation," *Optical Eng.*, vol. 35, no. 10, pp. 2852–2863, Oct. 1996.

[9] A. C. Bovik, "Analysis of multichannel narrow-band filters for image texture segmentation," *IEEE Trans. Signal Processing*, vol. 39, no. 9, pp. 2025–2043, Sept. 1991.

[10] D. F. Dunn and W. E. Higgins, "Optimal Gabor filters for texture segmentation," *IEEE Trans. Image Proc.*, vol. 4, no. 7, pp. 947–964, July 1995.

[11] T. P. Weldon, *Multiresolution Design of Multiple Gabor Filters for Texture Segmentation*. PhD thesis, The Pennsylvania State University, 1995.

[12] W. K. Pratt, *Digital Image Processing*. John Wiley and Sons, second ed., 1991.

[13] D. Lainiotis, "A class of upper bounds on probability of error for multihypotheses pattern recognition," *IEEE Trans. Inform. Theory*, vol. 15, pp. 730–731, Nov. 1969.

[14] L. Kanal, "Patterns in pattern recognition: 1968-1974," *IEEE Trans. Inform. Theory*, vol. 20, no. 6, pp. 697–722, Nov. 1974.

[15] P. Brodatz, *Textures: A Photographic Album for Artists and Designers*. New York, NY: Dover, 1966.