Available online at www.sciencedirect.com
SCIENCE DIRECT®
Real-Time Imaging 10 (2004) 365–370
www.elsevier.com/locate/rti

Robust minimization of lighting variation for real-time defect detection

Edmund Y. Lam*

Department of Electrical and Electronic Engineering, University of Hong Kong, Pokfulam Road, Hong Kong

Available online 14 October 2004

Abstract

In machine vision applications that involve comparing two images, it is necessary to match the capture conditions, which can affect their graylevels. Illumination and exposure are two important causes for lighting variation that we should compensate for in the resulting images. A standard technique for this purpose is to map one of the images to achieve the smallest mean square error (MSE) between the two. However, applications in defect detection for manufacturing processes are more challenging, because the existence of defects would affect the mapping significantly. In this paper, we present a robust method that is more tolerant to defects, and discuss its formulation as a linear programming to achieve fast implementations. This algorithm is also flexible and capable of incorporating further constraints, such as ensuring non-negativity of the pixel values.

1. Introduction

When an image is captured by a camera, its quality depends on a number of factors, such as illumination condition, exposure time of the film or sensor, optical aberration, and sensor noise [1]. Ordinarily, for consumer photography, the aberration and noise can be kept minimal. Therefore, two pictures taken under similar illumination and exposure would appear fairly identical. Yet for machine vision, when such images need to be further processed or combined together, the conditions are often more rigid and we need to pay attention to the difference in image capture conditions.

One particularly challenging application that further complicates the issue is in industrial inspection. Our interest here is in semiconductor manufacturing, but the technique we introduce can potentially be used in other industries as well [2]. Semiconductor manufacturing is a complicated task with very stringent requirements [3,4]. Scrupulous inspection is needed to avoid malfunctioning of the integrated circuits (IC) due to misplacement, contamination, or missing components, a process generally known as defect detection. This is usually performed after every major step in the manufacturing process with automated visual inspection (AVI) techniques by machine vision. While such techniques constantly need to be refined and improved to meet with the ever-shrinking feature size and ever-increasing complexity of the IC, a basic setup can be described as follows. A reference IC is first carefully examined to ensure that it is free from defect. A test IC is then compared against the reference, and we observe if they have any visual difference. This difference indicates the potential areas of defect [5]. Usually, some post-processing steps are required to analyze and classify the difference image, which would eventually result in a report detailing the existence and attributes of the defects. Clearly, the success of this defect detection step depends on an accurate comparison between the test and the reference. This comparison needs to be robust against variations in the image capture conditions. As usually a large volume of ICs has to be inspected, the vision algorithm also needs to be real-time.

Illumination and exposure differences are the two main variations in capture conditions that need to be accounted for [6]. The intensity of an image depends on the nature of illumination, the reflectance of the subject,

*Tel.: +852 2241 5942; fax: +852 2559 8738.
E-mail address: elam@eee.hku.hk (E.Y. Lam).

1077-2014/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.rti.2004.08.001

and the camera setting. We explore the details of the lighting variation and its model in Section 2. This brings us to the need of an illumination normalization scheme [7]. We explore two alternatives, first with a least-squares method in Section 2.1, and then with a linear method in Section 2.2. The latter is seen to be more robust against the existence of defects, and is capable of high speed, due to the efficiency of linear programming. Simulations are provided in Section 3 followed by some concluding remarks.

2. Lighting variation minimization

Consider that we have taken two images: the first one is with the reference die which we know to be defect-free, while the second one is with the test die that we are going to inspect. Let λ denote the wavelength of electromagnetic radiation. The first die has reflectance $R_r(\lambda; x, y)$ and is subject to illumination $I_r(\lambda; x, y)$ at location (x,y), while the second die has reflectance $R_t(\lambda; \hat{x}, \hat{y})$ and is subject to illumination $I_t(\lambda; \hat{x}, \hat{y})$ at location $(\hat{x}, \hat{y})$. The subscripts r and t refer to reference and test, respectively. Note that they use different coordinate systems because the two dies may not have been loaded at identical positions. It is therefore necessary to align them algorithmically. A correlation-based subregion alignment method is a possible and efficient technique for such a purpose, because this is not very sensitive to illumination variation. This can be achieved as follows: The reference and test images are divided into blocks of size 64×64 or 128×128. Correlations are computed on the corresponding blocks, with the test block shifted within ± 1 to ± 2 pixels in both the horizontal and vertical directions, the latter often preferred to allow for greater robustness against die positions. The shifts are at multiples of 0.25 pixel, and employ a simple bilinear interpolation when necessary. The corresponding blocks are aligned at the shift amount that maximizes the correlation value. In what follows, we use the same coordinate system assuming that the two images have been properly aligned spatially.

Let $f_r(x, y)$ denote the reference sub-image, and $f_t(x, y)$ denote the test sub-image, each of size $N \times N$. We know

$$f_r(x, y) = \int_{-\infty}^{\infty} c_r(\lambda; x, y) I_r(\lambda; x, y) R_r(\lambda; x, y)\, \mathrm{d}\lambda, \tag{1}$$

$$f_t(x, y) = \int_{-\infty}^{\infty} c_t(\lambda; x, y) I_t(\lambda; x, y) R_t(\lambda; x, y)\, \mathrm{d}\lambda, \tag{2}$$

where $c_r(\lambda)$ and $c_t(\lambda)$ are the camera sensor responses [8]. The reference and the test dies should have roughly the same spectral reflectance, i.e., $R_r(\lambda; x, y) \approx R_t(\lambda; x, y)$, if both are free from defect. If they were subject to illuminations with identical spectral power distribution and the same camera setting, we have $I_r(\lambda) \approx I_t(\lambda)$ and $c_r(\lambda) \approx c_t(\lambda)$. By Eqs. (1) and (2), they would imply $f_r(x, y) = f_t(x, y)$ except in areas where defects exist, which alter the spectral reflectance of the test die. Therefore,

$$\Delta f(x, y) = |f_r(x, y) - f_t(x, y)| \tag{3}$$

is a map for the defect locations.

In a more realistic situation, however, the two illuminations $I_r(\lambda)$ and $I_t(\lambda)$ can be quite distinct, and different settings of the exposure would imply $c_r(\lambda) \neq c_t(\lambda)$. Mathematically, it is not possible to relate $f_t(x, y)$ and $f_r(x, y)$ simply except in degenerate cases, such as when $c_r(\lambda) I_r(\lambda) = \alpha c_t(\lambda) I_t(\lambda)$ at all locations for a constant α. The existence of noise in sensors would render this impossible. However, it has been found experimentally that under most lighting variations in such a controlled environment as semiconductor inspection, we can approximate the relationships of $f_t(x, y)$ and $f_r(x, y)$, when there is no defect, with a transformation $\mathscr{T}$ such that [9]

$$f_r(x, y) \approx \mathscr{T}\{f_t(x, y)\} \tag{4}$$

$$= \sum_{i=1}^{M} a_i f_{t,i}(x, y), \tag{5}$$

where $f_{t,i}(x, y)$'s are images that depend on $f_t(x, y)$. They can incorporate various condition changes between taking the reference and the test images, such as linear scaling, x- and y-directional intensity changes, to name a few. Therefore, some possible terms for $f_{t,i}(x, y)$ include [9]

- $f_t(x, y)$,
- $x f_t(x, y)$,
- $y f_t(x, y)$,
- $f_t(x, y) - f_t(x - 1, y)$,
- $|f_t(x, y) - f_t(x - 1, y)|$,
- $\sqrt{x^2 + y^2} f_t(x, y)$,
- $1(x, y)$, which means that the entire image is a constant.

The variable a_i's are the design parameters. For simple illumination changes that may occur in the semiconductor manufacturing process, the approximation in Eq. (5) is fairly good [9]. Assuming that we are able to obtain accurate values of a_i's, we can modify the defect detection equation (3) to

$$\Delta f(x, y) = \left| f_r(x, y) - \sum_{i=1}^{M} a_i f_{t,i}(x, y) \right|, \tag{6}$$

which again provides a map for the locations of possible defects.

As such, an effective algorithm to determine the values of a_i's is essential. In what follows, we first

explain the tradition least-squares approach that is often used to solve such a mapping problem, and discuss the drawbacks that we would face. We then describe our approach based on linear programming, which is seen to provide us with a robust and fast solution.

2.1. A least squares approach

We seek to find the optimal a_i's, or $\mathbf{a}$ using a vector notation. A common method that is often employed to deal with lighting variation presented in Eq. (5) is by means of least square. Let $\mathbf{f_{t,i}}$ and $\mathbf{f_r}$ be the raster scan of $f_{t,i}(x,y)$ and $f_r(x,y)$, respectively, so each is a vector of length N^2. We also use $\mathbf{f}_\tau$ to denote the matrix $[\mathbf{f_{t,1}} \ldots \mathbf{f_{t,M}}]$. The optimal value $\mathbf{a}^*$ that gives the smallest mean square error (MSE) in the approximation in Eq. (5) can be found by

$$\mathbf{a}^* = \arg\min_{\mathbf{a}} \left\| [\mathbf{f_{t,1}} \ldots \mathbf{f_{t,M}}] \begin{bmatrix} a_1 \\ \vdots \\ a_M \end{bmatrix} - \mathbf{f_r} \right\|_2 \tag{7}$$

$$= \arg\min_{\mathbf{a}} ||\mathbf{f}_\tau \mathbf{a} - \mathbf{f_r}||_2, \tag{8}$$

where $||\cdot||_2$ denotes the L_2 norm. Analytical solution exists, which is given by

$$\mathbf{a}^* = (\mathbf{f}_\tau^{\mathrm{T}} \mathbf{f}_\tau)^{-1} \mathbf{f}_\tau^{\mathrm{T}} \mathbf{f_r}, \tag{9}$$

where T denotes the transpose of a matrix.

Mathematically, a least-squares solution is valid provided that the matrix $\mathbf{f}_\tau$ is well conditioned [10]. Yet, our application in defect detection has an extra complication that the direct least-squares solution cannot address easily. The test image may contain defects in addition to illumination variations. If we just compute Eq. (8), the value of $\mathbf{a}^*$ will be affected by the defects. To cope with this problem, we can use least squares iteratively with potential defect areas removed from each successive calculations [9]. However, this will slow down the inspection process. We therefore face a tradeoff between speed and accuracy when using the least-squares approach. Furthermore, a direct least-squares approach makes no guarantee that the resulting pixels in $\sum a_i f_{t,i}(x,y)$ will create pixel values that are within the allowable range, between zero and maximum intensity. If we add these as constraints, we no longer have an analytical solution like Eq. (9). An ad hoc solution would be to clip the resulting image in $\sum a_i f_{t,i}(x,y)$ to have intensity values within the allowable range, but then we would have no guarantee of optimality with respect to least squares error.

These factors bring us to a constrained minimization using L_1 norm.

2.2. A linear programming approach

We reformulate our optimization criterion as

$$\mathbf{a}^* = \arg\min_{\mathbf{a}} ||\mathbf{f}_\tau \mathbf{a} - \mathbf{f_r}||_1. \tag{10}$$

Comparing Eq. (10) with Eq. (7) above, we see that this is simply replacing L_2 norm with L_1 norm. Although the change appears to be small, it is instrumental to ensure that our solution is more robust against existence of defects. This is because defects constitute the areas where the mapping cannot be done. If we use the L_2 norm, we penalize with the square of the errors in the mapping; as such, the algorithm would tend to give less correct parameters that attempt to map the defects as much as possible and sacrifice the accuracy in the other regions where defects do not exist. In other words, $\mathbf{a}^*$ will be significantly affected by the defects. With L_1 norm, however, the error contribution is only linear, and the algorithm is likely to favor greater accuracy in the defect-free regions more. This change therefore produces better lighting variation minimization when the size of the defect cannot be neglected. In addition, we also like to have the constraint on the maximum and minimum intensity values, so

$$\mathbf{i}_{\min} \preceq \mathbf{f}_\tau \mathbf{a} \preceq \mathbf{i}_{\max}, \tag{11}$$

where $\preceq$ is an element-by-element inequality sign.

There is, however, no analytic solution to the constrained minimization problem above. We now show that it is possible to formulate it as a linear programming (LP) problem. This can be solved very efficiently using a variety of techniques, such as a primal-dual simplex method [11].

We first transform Eq. (10) to

$$\begin{aligned} &\text{minimize} && \phi(g_1) + \phi(g_2) + \cdots + \phi(g_{N^2}) \\ &\text{subject to} && \mathbf{g} = \mathbf{f}_\tau \mathbf{a} - \mathbf{f_r}, \end{aligned} \tag{12}$$

where g_i denotes the ith component of the vector $\mathbf{g}$, with a total of N^2 elements, and $\phi(g_i) = |g_i|$. ϕ is called the penalty function [11]. This formulation permits the use of other forms of the penalty function, such as the Huber function, which is a parabola in the vicinity of zero and increases linearly after a given level [12].

We further transform the optimization problem by introducing a new variable $\mathbf{t}$. Let

$$\mathbf{b} = \begin{bmatrix} \mathbf{a} \\ \mathbf{t} \end{bmatrix} \tag{13}$$

be our parameters, and we solve the following constrained optimization problem:

$$\begin{aligned} &\text{minimize} && [\mathbf{0}\ \mathbf{1}]\mathbf{b} \\ &\text{subject to} && [\mathbf{f}_\tau\ \mathbf{0}]\mathbf{b} - \mathbf{f_r} \preceq \mathbf{t}, \\ & && [\mathbf{f}_\tau\ \mathbf{0}]\mathbf{b} - \mathbf{f_r} \succeq -\mathbf{t}. \end{aligned} \tag{14}$$

Note that for each N^2 element, one of the inequalities above must be strict in order to minimize the objective function. Hence, we see that with $\mathbf{t} = |\mathbf{g}|$ the above equation is an equivalent formulation to Eq. (12).

We can further rearrange the terms to be

$$\begin{aligned} &\text{minimize} && [\mathbf{0}\ \mathbf{1}]\mathbf{b} \\ &\text{subject to} && [\mathbf{f}_\tau\ -I]\mathbf{b} \preceq \mathbf{f_r}, \\ & && [-\mathbf{f}_\tau\ -I]\mathbf{b} \preceq -\mathbf{f_r}, \end{aligned} \tag{15}$$

where I is the identity matrix. Moreover, we can add the constraints

$$\begin{aligned} &[\mathbf{f}_\tau\ \mathbf{0}]\mathbf{b} \preceq \mathbf{i}_{\max}, \\ &[-\mathbf{f}_\tau\ \mathbf{0}]\mathbf{b} \preceq -\mathbf{i}_{\min} \end{aligned} \tag{16}$$

to ensure that the adjusted pixel values fall between $[\mathbf{i}_{\min}, \mathbf{i}_{\max}]$.

It is important to realize that for the equations in (15) and (16), they are all linear in $\mathbf{b}$. In fact, we have already presented them in a standard form of an LP problem. This lends itself to direct usage of standard packages that can solve such problems efficiently. For details, the reader is referred to books such as Ref. [13].

3. Simulations

We apply the method described above on bump inspection, which is a critical process in die bonding in semiconductor assembly [3]. Bumps are the electrical and mechanical connection between the die and the substrate, and are formed from processes such as paste-deposition and electroplating. As such, the shape of the bumps may vary, and may have a few potential defects such as missing bumps, bridged bumps, contaminants on or between bumps, and incorrect bump volumes or heights [14].

We test in particular two common types of defects: missing bumps and bridged bumps. An image of the reference IC is shown in Fig. 1(a). This reference is defect-free. A test IC that missed some of the bumps is shown in Fig. 1(b). In addition, there are some lighting variations between the two images. If we just compute their differences, the resulting image is shown in Fig. 1(c). The intensity in this image has been magnified for us to observe the differences. Evidently, areas not due to the defect can still have significant difference between the reference and test images, which should be attributed to the lighting variations. Figs. 1(d) and (e) are the difference images after we performed mapping to the test images. In both cases, the differences due to lighting have been suppressed, with our linear programming approach (Fig. 1(e)) slightly better than the least-squares approach (Fig. 1(d)). It is implied that this defect does not affect the mapping too much.

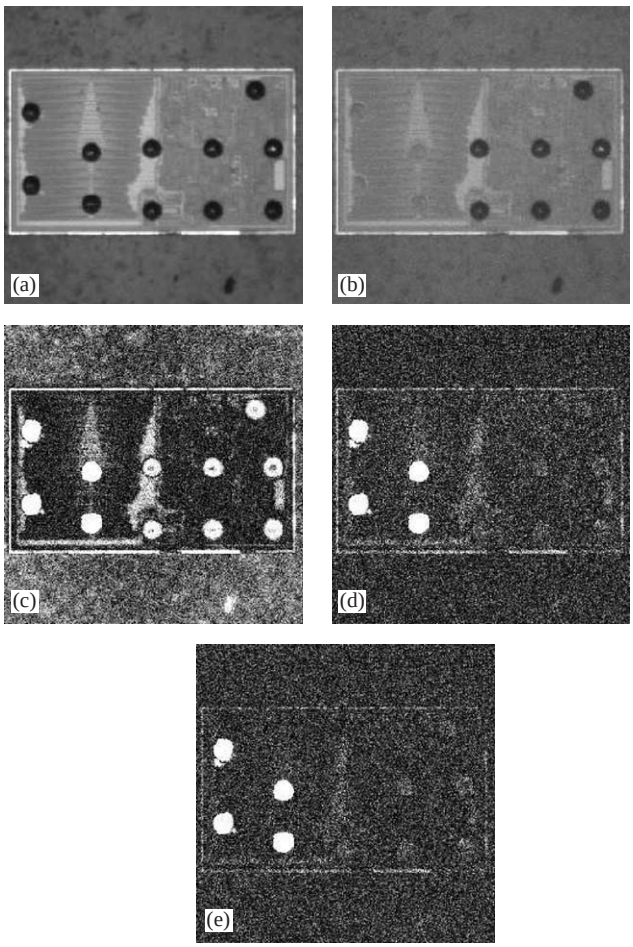

Fig. 1. An example of bump inspection with missing bump. (a) Reference image, (b) test image with missing bump, (c) $\Delta f(x, y)$ without mapping, (d) $\Delta f(x, y)$ with mapping using L_2 and (e) $\Delta f(x, y)$ with mapping using L_1.

Table 1
Quality improvement in bump inspection simulation

	Signal-to-noise ratio (SNR) gain	
	Least square (L_2) (dB)	Linear programming (L_1) (dB)
Missing bump	2.31	2.38
Bridged bump	1.49	3.43

A numerical comparison of the signal-to-noise ratio (SNR) gain is shown in Table 1, as a quantitative measure of our L_1 norm minimization approach. The SNR between an image f and a reference image f_r is computed as

$$\mathrm{SNR}(f, f_r) = 10 \log_{10} \left[\left(\frac{\sum_x \sum_y (f_r(x,y))^2}{\sum_x \sum_y (f(x,y) - f_r(x,y))^2} \right) \right]_{95\%}, \tag{17}$$

where the notation $[\cdot]_{95\%}$ means that we discard 5% of the pixels with the greatest errors after the grayscale adjustment, because those pixels are mostly likely defects. The percentage can change with a priori knowledge about the proportion of defects in the image. The SNR gain in Table 1 is computed with

$$\text{SNR gain} = \text{SNR}\left(\sum_{i=1}^{M} a_i f_{t,i}(x,y), f_r(x,y)\right) - \text{SNR}(f_t(x,y), f_r(x,y)), \quad (18)$$

where a larger value indicates more correction of the lighting variation. For the case of missing bump, the L_1 minimization method is slightly better than the one using L_2 norm.

We also consider the case for bridged bumps in Fig. 2. Again, the direct difference shows a lot of errors (Fig. 2(b)). When we compare the differences after the mapping, our approach (Fig. 2(d)) is significantly better than the one with least-square (Fig. 2(c)). For our method, most of the non-defect differences have been suppressed. This is because the defect imposes significant effects on the least-squares mapping to render it ineffective, while our method is more robust against it. Again, the numerical comparison of the SNR is shown in Table 1. The gain is more substantial for bridged bumps, in agreement with our visual comparisons.

We also compare the two methods in terms of speed. A QR factorization method is employed for the least-squares method, while an interior point algorithm is used for the linear programming [11,15]. The speed of execution for the entire illumination correction procedure is shown in Table 2. We can observe that in both cases, the linear programming method is slower by nearly a factor of 2. However, we must bear in mind that the least-squares method often has to be run multiple times to minimize the effect of the defects, as we discussed in Section 2.1. As such, it may not hold any advantage in terms of speed as compared to the linear programming method that we propose here.

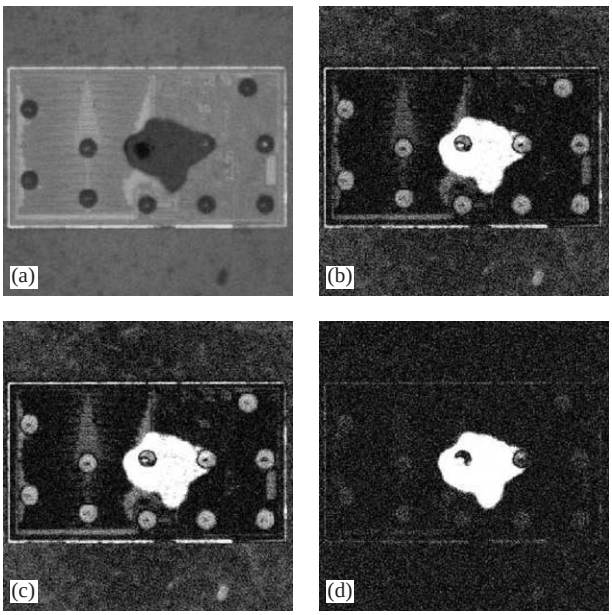

Fig. 2. An example of bump inspection with bridged bump. (a) Test image with bridged bump, (b) $\Delta f(x,y)$ without mapping, (c) $\Delta f(x,y)$ with mapping using L_2 and (d) $\Delta f(x,y)$ with mapping using L_1.

Table 2
Speed comparison in bump inspection simulation

	Speed of execution	
	Least square (L_2) (s)	Linear programming (L_1) (s)
Missing bump	0.6717	1.2529
Bridged bump	0.6479	1.2496

4. Conclusions

In this paper, we have described a linear programming formulation to minimize the lighting variation between two images. This method is particularly useful in inspection for manufacturing processes, where defects could impose significant effects on the mapping of intensity values. Simulation results on bump images in semiconductor assembly have shown support for our approach.

Acknowledgements

The financial support by ASM Assembly Automation Ltd and the Innovation and Technology Fund of the Hong Kong Special Administrative Region Government for this work is gratefully acknowledged.

References

[1] Lam EY. Image restoration in digital photography. IEEE Transactions on Consumer Electronics 2003;49(2):269–74.
[2] Lai S-H, Fang M. A novel illumination compensation algorithm for industrial inspection. In: Machine vision applications in industrial inspection. Proceedings of the SPIE, vol. 3652, 1999. p. 50–8.
[3] Van Zant P. Microchip fabrication: a practical guide to semiconductor processing, 4th ed. New York: McGraw-Hill; 2000.
[4] Quirk M, Serda J. Semiconductor manufacturing technology. Englewood Cliffs, NJ: Prentice-Hall; 2001.
[5] Barth M, Hirayama D, Beni G, Hackwood S. A color vision inspection system for integrated circuit manufacturing. IEEE Transactions on Semiconductor Manufacturing 1992;5(4): 290–301.

[6] Candocia FM. Simultaneous homographic and comparametric alignment of multiple exposure-adjusted pictures of the same scene. IEEE Transactions on Image Processing 2003;12(12):1485–94.
[7] Matsushita Y, Nishino K, Ikeuchi K, Sakauchi M. Illumination normalization with time-dependent intrinsic images for video surveillance. In: IEEE international conference on computer vision and pattern recognition, vol. 1, 2003. p. 3–10.
[8] Giorgianni EJ, Madden TE. Digital color management: encoding solutions. Reading, MA: Addison-Wesley; 1998.
[9] Lam EY. Graylevel alignment between two images using linear programming. In: 2003 International conference on image processing, vol. 2, 2003. p. 327–30.
[10] Kincaid D, Cheney W. Numerical analysis: mathematics of scientific computing, 3rd ed. Pacific Grove, CA: Brooks/Cole; 2002.
[11] Boyd S, Vandenberghe L. Convex optimization. Cambridge: Cambridge University Press; 2004.
[12] Huber P. Robust statistics. New York: Wiley; 1981.
[13] Bertsimas D, Tsitsiklis JN. Introduction to linear optimization. MA: Athena Scientific; 1997.
[14] Hibert S. Micro inspection for wafer bumping. Advanced Packaging 2002;11(2):19–22.
[15] Golub GH, Van Loan CF. Matrix computations, 3rd ed. Baltimore, MD: Johns Hopkins University Press; 1996.