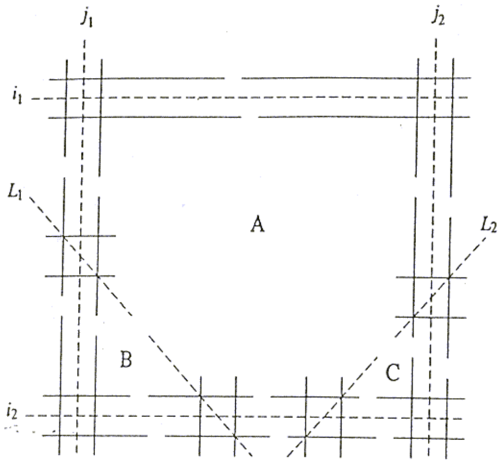

Fig. 7. Illustration of parallel shrinking process.

Theorem 3: The maximum number of steps necessary for completely shrinking a given component with an $m \times n$ bounding rectangle by using φ_t is $\lceil \frac{m}{2} \rceil + n-1$.

Proof: Suppose a component is bounded by an $m \times n$ rectangle as shown in Fig. 7. As we observed in the proof of Theorem 2, every application of φ_t moves both line L_1 and line L_2 up one pixel. Thus, hey are two pixels closer along the bottom side of the bounding rectangle. After at most $\lceil \frac{m}{2} \rceil - 1$ applications of φ_t, line L_1 and line L_2 will be one or two pixels apart on the bottom side of the rectangle. From now on, every application of φ_t will turn the black pixel(s) on he bottom to white and leave one or two black pixels on the above row. Thus, n more applications of φ_t will completely shrink the component. Therefore, the maximum of steps to completely shrink a component bounded by an $m \times n$ rectangle is $\lceil \frac{m}{2} \rceil + n-1$. ◇

When the parallel shrinking algorithm using the shrinking operator φ_t is adopted in an image component labeling algorithm using local operators, a 2×3 neighborhood should be used in the label-propagate operation. In addition, when a new label is given to a component, the bel-propagate operation should check if it is the case in which two horizontally connected pixels represent the component. If so, the two pixels should be assigned the same label.

It is easy to see that using the shrinking operator φ_l, which shrinks each component toward the top of its bounding rectangle, a parallel shrinking algorithm can completely shrink a component bounded by an $m \times n$ rectangle in $m + \lceil \frac{n}{2} \rceil - 1$ steps. Thus, we can make our parallel shrinking algorithm choose a shrinking operator depending on the height and the width of the source image. Given an $m \times n$ binary image, if $m > n$, choose the shrinking operator φ_l; otherwise, choose the shrinking operator φ_t. Therefore, the parallel shrinking algorithm es at most $\min(\lceil \frac{m}{2} \rceil + n-1,\ m + \lceil \frac{n}{2} \rceil - 1)$ steps to completely hrink an $m \times n$ binary image.

IV. Conclusion

For a component bounded by an $n \times n$ rectangle, it is not difficult to see that any shrinking algorithm that shrinks components one pixel one pixel preserving its 8-connectivity in the shrinking process es at least $\frac{n}{2}$ steps in the worst case since there cannot exist a xel in the rectangle such that the maximum 4-connected distance ween that pixel and any pixel on the component boundary is less than $\frac{n}{2}$. Thus, this kind of parallel shrinking algorithm shrinks an $n \times n$ binary image with complexity $\Omega(n)$.

Levialdi's parallel shrinking algorithm shrinks an $n \times n$ binary image within $O(n)$ steps by changing black pixels to white and changing some white pixels to black as necessary. Thus, his algorithm is optimal asymptotically but with a multiplicative constant of 2.

The new parallel shrinking algorithm we presented is also optimal asymptotically and shrinks an $n \times n$ binary image within $O(n)$ steps. However, we have a smaller mulitiplicative constant, namely, 1.5.

Acknowledgment

The authors wish to thank Dr. S. Lambert and Dr. P. Coffield of Wright Laboratories, Eglin AFB, for their continued support of this research. The authors also wish to acknowledge the helpful suggestions made by the reviewers of this manuscript.

References

[1] R. Cypher, J. L. C. Sanz, and L. Snyder, "Algorithms for image component labeling on SIMD mesh connected computers," *IEEE Trans. Comput.*, vol. 39, no. 2, pp. 276–281, 1990.

[2] H. M. Alnuweiri and V. K. Prasanna, "Fast image labeling using local operators on meshconnected computers," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 13, no. 2, pp. 202–207, 1991.

[3] ——, "Parallel architectures and algorithms for image component labeling," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 14, no. 10, pp. 1014–1034, 1992.

[4] H. Shi and G. X. Ritter, "$O(n)$-time and $O(\log n)$-space image component labeling with local operators on SIMD mesh connected computers," in *Proc. 1993 Int. Conf. Parallel Processing*, 1993.

[5] ——, "A fast algorithm for image component labeling with local operators on mesh connected computers," *J. Parallel Distributed Comput.*, vol 23, no. 3, pp. 455–461, 1994.

[6] S. Levialdi, "On shrinking binary picture patterns," *Communi. ACM*, vol. 15, no. 1, 1972.

Erratum and Addendum to "Iterative Image Reconstruction Algorithms Based on Cross-Entropy Minimization"

Charles L. Byrne

***Abstract*—In the above-named paper [1], there is a gap in the proof of Theorem 2 for case b) (the inconsistent case) with no regularization (α = 1). In addition, convergence can be established without using the "full rank" assumption for the matrix P. The aim here is to prove Theorem 2 with neither the gap nor the assumption.**

There is a gap in the proof of Theorem 2, case b) (the inconsistent case) with α=1. Equations (13) and (14) of [1] allow us to conclude only that $\mathrm{KL}(Px^*, y) \leq \mathrm{KL}(Px, y)$ for all $x \geq 0$ for which

Manuscript received June 21, 1993; revised December 1, 1993. The associate editor coordinating the review of this paper and approving it for publication was Prof. Yoram Bresler.

The author is with the Department of Mathematics, University of Massachusetts at Lowell, Lowell, MA 01854 USA.

IEEE Log Number 9407599.

$\mathrm{KL}(x, x^*)$ is finite. However, from lemmas 5 and 6, we have that

$$\begin{aligned} &\mathrm{KL}(x, x^m) + \mathrm{KL}(Px, y) - \mathrm{KL}(Px, Px^m) \\ &= \mathrm{KL}(x, x^{m+1}) + \mathrm{KL}(Px^{m+1}, y) \\ &\quad + \mathrm{KL}(x^{m+1}, x^m) - \mathrm{KL}(Px^{m+1}, Px^m). \end{aligned} \tag{1}$$

Let $x = x'$ be any nonnegative minimizer of $\mathrm{KL}(Px, y)$; then, Px' is unique, even if x' is not. From (1), it follows that $\mathrm{KL}(x', x^m) \geq \mathrm{KL}(x', x^{m+1})$, from which we conclude that $\mathrm{KL}(x', x^*)$ is finite for all x'. Consequently, x^* is a global minimizer of $\mathrm{KL}(Px, y)$. From (1), it follows that the difference

$$\begin{aligned} &\mathrm{KL}(x', x^m) - \mathrm{KL}(x', x^{m+1}) \\ &= \mathrm{KL}(Px^{m+1}, y) - \mathrm{KL}(Px', y) + \mathrm{KL}(Px', Px^m) \\ &\quad + \mathrm{KL}(x^{m+1}, x^m) - \mathrm{KL}(Px^{m+1}, Px^m) \end{aligned} \tag{2}$$

depends only on Px' and not directly on x'. Therefore, the difference $\mathrm{KL}(x', x^0) - \mathrm{KL}(x', x^*)$ is independent of x', for x' as above. It follows that the choice $x' = x^*$ is the unique nonnegative global minimizer of $\mathrm{KL}(Px, y)$ for which $\mathrm{KL}(x', x^0)$ is minimized. Therefore, the x^* is unique, and the sequence $\{x^m\}$ must converge to x^*. For the case α=1, we have the following result:

Theorem: The sequence $\{x^m\}$ converges to a limit x^∞ for all I and J, for all starting points $x^0 > 0$. The limit $x' = x^\infty$ is the unique nonnegative minimizer of $\mathrm{KL}(Px', y)$ for which $\mathrm{KL}(x', x^0)$ is minimized if there is no nonnegative solution of $y = Px$. If there are nonnegative solutions of $y = Px$, then the limit may depend on the starting value; we have $y = Px^\infty$, with $x = x^\infty$ the unique solution minimizing $\mathrm{KL}(x, x^0)$, and support(x) is contained within support(x^∞) for all solutions x.

References

[1] C. Byrne, "Iterative image reconstruction algorithms based on cross-entropy minimization," *IEEE Trans. Image Processing*, vol. 2, no. 1, pp. 96–103, Jan. 1993.