November 1988

A Contour Base Recovery of Image Flow: Iterative Method

Jian Wu
Harvard University

K. Wohn
University of Pennsylvania

Roger Brockett
Harvard University

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-88-91.

A Contour Base Recovery of Image Flow: Iterative Method

Abstract

We present an iterative algorithm for the recovery of 2-D motion, i.e., an algorithm for the determination of a transformation which maps one image onto another. The local ambiguity in measuring the motion of contour segments (called the "aperture problem") forces us to rely on measurements along the normal direction. Since the *measured* "normal flow" itself does not agree with the *actual* normal flow, the "full flow" recovered from this erroneous normal flow possesses substantial error too, and any attempt to recover the 3-D motion from such full flow is doomed to failure.

Our method is based on the observation that a polynomial approximation of image flow provides sufficient information for 3-D motion computation. The use of an explicit flow model enables us to improve normal flow estimates through an iterative process. We discuss the adequacy and the convergence of the proposed algorithm. The algorithm has been tested on synthetic and some of simple natural time-varying images. The image flow recovered from this scheme was sufficiently accurate so as to be useful in 3-D structure and motion computation.

Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-88-91.

A CONTOUR-BASED RECOVERY OF IMAGE FLOW: ITERATIVE METHOD

Jian Wu, K. Wohn
and Roger Brockett

MS-CIS-88-91
GRASP LAB 164

Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA 19104

November 1988

Acknowledgements: This research was supported in part by DARPA grants NOOO14-85-K-0018, N00014-88-K-0632, U.S Postal Service contract 104230-87-H0001/M-0195, NSF grants MCS-8219196-CER, IRI84-10413-AO2 and U.S. Army grants DAA29-84-K-0061, DAA29-84-9-0027.

A Contour-based Recovery of Image Flow: Iterative Method

Jian Wu[1]
Division of Applied Sciences
Harvard University

K. Wohn
Department of Computer and Information Science
University of Pennsylvania

Roger Brockett
Division of Applied Sciences
Harvard University

November 10, 1988

[1]current address: Laboratory for Sensory Robotics, Department of Electrical, Computer and System Engineering, Boston University

A Contour-based Recovery of Image Flow: Iterative Method

Abstract

We present an iterative algorithm for the recovery of 2-D motion, i.e., an algorithm for the determination of a transformation which maps one image onto another. The local ambiguity in measuring the motion of contour segments (called the "aperture problem") forces us to rely on measurements along the normal direction. Since the *measured* "normal flow" itself does not agree with the *actual* normal flow, the "full flow" recovered from this erroneous normal flow possesses substantial error too, and any attempt to recover the 3-D motion from such full flow is doomed to failure.

Our method is based on the observation that a polynomial approximation of image flow provides sufficient information for 3-D motion computation. The use of an explicit flow model enables us to improve normal flow estimates through an iterative process. We discuss the adequacy and the convergence of the proposed algorithm. The algorithm has been tested on synthetic and some of simple natural time-varying images. The image flow recovered from this scheme was sufficiently accurate so as to be useful in 3-D structure and motion computation.

1 Introduction

One issue in analyzing time-varying images is that of obtaining the motion of image from consecutive image frames. The way in which the image changes in time can be conceptualized as being the result of a motion generated by a 2-D vector field defined in the image coordinates. Such a vector field is the perspective projection of the three dimensional velocity vector at a point on the surface of the object. Our approach is based on a representation of the image flow as viewed in the Eulerian description (as opposed to the Lagrangian description). This was pioneered by Koenderink [5]. It relies on the continuity properties of the flow field within a small neighborhood. Waxman and Ullman [9] have shown that the 3-D motion and object structure (in terms of surface metric and curvature tensors) can be recovered from the image flow field and its first and second derivatives with respect to space. However their approach involves solving a set of non-linear equations whose behavior is largely determined by the second-order derivatives of flow field. Wohn and Wu [11] introduced an algorithm which is based on the flow field and its first derivatives over three consecutive frames, avoiding the use of second order spatial derivatives.

The Eulerian approach assumes that flow field and its partial derivatives are already available in a small region. Such a rich description of image motion is not likely to be obtained with sufficient accuracy from feature points unless there are many points which can be localized with extreme accuracy. Intensity itself is another possible candidate from which image motion might be recovered. Under the assumption of convected invariance of the image intensity, image motion can be measured by three different mechanisms: intensity correlation, differentiation in spatio-temporal space and filtering in frequency domain. Recently, analysis in the frequency domain method seems to regain favor (e.g., [2]).

The texture boundaries and physical surface discontinuities are another source of information about the flow field. By observing the contours associated with these features, one can compute a flow. The local ambiguity in determining the flow vector along contours (called the "aperture problem") can be overcome by introducing additional constraints obtained from physical aspect of 3-D motion or from a priori knowledge about the scene ([3, 4]). Waxman and Wohn proposed a second-order polynomial approximation for the flow, and developed an algorithm (called the velocity functional method) which determines the twelve coefficients of the polynomial approximation by solving a linear system [10]. Their model predicts the actual motion for planar surfaces under

general 3-D motion, given the exact normal components of flow along contours. However, in real situations, the normal flow can be measured only to within certain accuracy and the error in the normal flow propagates to the consecutive stages of the motion computation. Often the resulting flow cannot be recovered with sufficient accuracy so as to be useful for the 3-D motion recovery.

We shall perform an error analysis on the normal velocity measurement. Based on this analysis, we are led to propose an iterative method which utilizes a least-squares method as a basic module to predict the full flow, and re-estimates the normal flow based on the predicted value of the full flow. This process is applied iteratively until its further application results in no significant improvement. We choose a first order polynomial approximation as a flow model since, for our purpose, the flow up to the first order will suffice for the next stage of 3-D motion computation. Issues on the convergence of the method are discussed in Section 5.2.2. We study the performance of the proposed algorithms with experiments conducted on synthetic and real time-varying images.

The rest of the paper is organized as follow:

- *Solving the Aperture Problem* (Section 2): The aperture problem is overcome by imposing some constraints on the flow field. In our case temporal changes are modeled as first order polynomials in the image plane. Least-squares method is used to determine the coefficients of polynomials.

- *Estimating normal flows* (Section 3, Appendix A): We discuss how to measure the normal flow along contour segments. The formular for the error in the normal flow estimate is derived (Appendix A). The error in the normal flow is proportional to the size of motion.

- *Estimating full flows* (Section 4): When the least-squares method is used to solve for the full flow, the error in the normal flow is propagated to the full flow. The estimation error in the full flow is proportional to the error in the normal flow, and involves other parameters as well.

- *Improving normal/full flows* (Section 5.1, Appendix B): The full flow computed from the least-squares process is used to re-estimate the normal flow. Insofar as the normal flow optained in this way has less error than the previous estimate had, the least-squares process guarantees the better estimate for the full flow. Appendix B verifies this argument by showing how the method works on a simple case - a square undergoing the 2-D euclidean motion.

- *On the convergence of the method (Section 5.2, Appendix C):* The iterative method is sound – it stops when it produces the correct answer. We also disscuss the computational complexity of the algorithm. A proof on convergence is found in Appendix C.

2 Solving the Aperture Problem

Suppose that contour Γ at time t_0 evolves into contour Γ_1 at time $t_1 = t + \Delta t$ such that

1. There is a transformation T existing between the two contours, *i.e.*, $\Gamma_1 = T\Gamma$.

2. $T = T(t_0)$ is small relative to the size of the contour. The size of a contour Γ is define as

$$\gamma = \max_{\mathbf{x}\in\Gamma} |\mathbf{x} - \mathbf{x}_0|$$

where $\mathbf{x}_0 = \int_\Gamma \mathbf{x} ds / \int_\Gamma ds$ with $ds = |d\mathbf{x}|$ is the center of the contour Γ (treated as a point set). If we normalize γ to be 1, then we say T is small iff

$$|T\mathbf{x} - \mathbf{x}| \ll 1,$$

for all $\mathbf{x} \in \Gamma$.

Under this condition, the velocity vector at $t = t_0$ for every point $\mathbf{x} \in \Gamma$ can be estimated by the displacement of that point:

$$\begin{aligned} \mathbf{v}(\mathbf{x}) &= \mathbf{v}(\mathbf{x}, t_0) \\ &= \lim_{t \longrightarrow t_0} \frac{T(t)\mathbf{x} - \mathbf{x}}{t - t_0} \\ &\approx \frac{T(t_0)\mathbf{x} - \mathbf{x}}{\Delta t}. \end{aligned}$$

If we choose Δt as the unit of the time scale then

$$\mathbf{v}(\mathbf{x}) \approx T\mathbf{x} - \mathbf{x}. \tag{1}$$

The right side of the above equation is the displacement vector from $\mathbf{x}$ to $T\mathbf{x}$. $T\mathbf{x} - \mathbf{x}$ can be measured directly if we can match the point $\mathbf{x}$ of Γ to its corresponding point $T\mathbf{x}$ of $T\Gamma$. Such a point-to-point match is only possible at so-called "feature points", *i.e.*, those points with features preserved under the transformation. For example, most of corners and junctions are preserved

under T, so they can be used to establish the point-to-point match. However, most points on the contour are not "feature points". For those points, it is well known that both component of $T\mathbf{x}-\mathbf{x}$ cannot be measured directly, due to the "aperture problem". Marr and Ullman have argued that it is impossible to measure the tangential component of $T\mathbf{x}-\mathbf{x}$ on short segments of evolving contours [8].

For the above reason, the measurement from any contour-based method is not $T\mathbf{x}-\mathbf{x}$, but the component of $T\mathbf{x}-\mathbf{x}$ in the direction of the unit normal vector $\mathbf{n}(\mathbf{x})$ of Γ at $\mathbf{x}$. This direction is often called the "normal direction"[1]. This component is called "normal flow" by many researchers. However, we shall not use the term "flow" since it has different meaning in mathematics. We shall use *displacement* for $T\mathbf{x}-\mathbf{x}$ and *velocity* for $\mathbf{v}(\mathbf{x})$. Consequently, we use *normal displacement*, denoted as $d_n(\mathbf{x})$ and *normal velocity*, denoted as $v_n(\mathbf{x})$ for their components in the direction of the unit normal vector.

Taking the inner product of $\mathbf{n}(\mathbf{x})$ with both sides of Equation (1) we have

$$\begin{cases} v_n(\mathbf{x}) \approx d_n(\mathbf{x}) & (2.a) \\ v_n(\mathbf{x}) \equiv \mathbf{n}^T(\mathbf{x})\mathbf{v}(\mathbf{x}), \quad \mathbf{x}\in\Gamma - N & (2.b) \\ d_n(\mathbf{x}) \equiv \mathbf{n}^T(x)(T(\mathbf{x})-\mathbf{x}) & (2.c) \end{cases} \tag{2}$$

where $N=\{\mathbf{x}_k\}$ is a finite set of points at which $\mathbf{n}(\mathbf{x}_k)$ is not well defined.

The Equations $(2.a)$ indicates that the value of the normal velocity $v_n(\mathbf{x})$ may be estimated by the normal displacement $d_n(\mathbf{x})$. The procedure by which we measure $d_n(\mathbf{x})$ from two given contours is based on Equation $(2.c)$ and will be discussed in the next section.

Assuming $v_n(\mathbf{x})$ is known, Equation $(2.b)$ can be used to recover $\mathbf{v}(\mathbf{x})$. Let us rewrite $\mathbf{n}^T(\mathbf{x})\mathbf{v}(\mathbf{x})$ as $\mathbf{c}^T(\mathbf{x})\mathbf{p}$, where $\mathbf{p}$ is a vector consisting of transformation parameters and $\mathbf{c}(\mathbf{x})$ is a vector determined by $\mathbf{x}$ and $\mathbf{n}(\mathbf{x})$. The explicit forms of $\mathbf{c}^T(\mathbf{x})$ and $\mathbf{p}$ for each transformation model can be calculated as follows.

1. Euclidean Motion Model:

$$\mathbf{c}(\mathbf{x}) = \begin{bmatrix} n_x \\ n_y \\ xn_y - yn_x \end{bmatrix}; \qquad \mathbf{p} = \begin{bmatrix} V_x \\ V_y \\ \Omega \end{bmatrix}.$$

[1] In the intensity-based method, "normal direction" refers to the direction of the gradient of the intensities.

2. Affine Motion Model:

$$\mathbf{c}(\mathbf{x}) = \begin{bmatrix} n_x \\ n_y \\ xn_x \\ xn_y \\ yn_x \\ yn_y \end{bmatrix}; \qquad \mathbf{p} = \begin{bmatrix} V_x \\ V_y \\ a_{11} \\ a_{21} \\ a_{12} \\ a_{22} \end{bmatrix},$$

where $\mathbf{v}(\mathbf{x}) = A\mathbf{x} + \mathbf{V}$ and a_{ij} is the i-j'th element of 2x2 matrix A. This model is locally valid for planar surfaces under rigid-body motion [11].

3. Second-Order Motion Model:

$$\mathbf{c}(\mathbf{x}) = \begin{bmatrix} n_x \\ n_y \\ xn_x \\ xn_y \\ yn_x \\ yn_y \\ x^2n_x + xyn_y \\ xyn_x + y^2n_y \end{bmatrix}; \qquad \mathbf{p} = \begin{bmatrix} m_{13} \\ m_{23} \\ m'_{11} \\ m_{21} \\ m_{12} \\ m'_{22} \\ -m_{31} \\ -m_{32} \end{bmatrix}.$$

where

$$\mathbf{v}(\mathbf{x}) = b + A\mathbf{x} + C\mathbf{x}^{[2]},$$

$$A = \begin{bmatrix} m'_{11} & m_{12} \\ m_{21} & m'_{22} \end{bmatrix}; \qquad b = \begin{bmatrix} m_{13} \\ m_{23} \end{bmatrix};$$

$$C = \begin{bmatrix} -m_{31} & -m_{32} & 0 \\ 0 & -m_{31} & -m_{32} \end{bmatrix}; \qquad \mathbf{x}^{[2]} = \begin{bmatrix} x^2 \\ 2xy \\ y^2 \end{bmatrix},$$

$$m'_{11} = m_{11} - m_{33},$$

$$m'_{22} = m_{22} - m_{33}.$$

This model is globally valid for planar surfaces under rigid-body motion and is locally valid for curved surfaces too [10].

Replacing $\mathbf{n}^T(\mathbf{x})\mathbf{v}(\mathbf{x})$ by $\mathbf{c}^T(\mathbf{x})\mathbf{p}$, Equation (2.*b*) becomes

$$\mathbf{c}^T(\mathbf{x})\mathbf{p} = v_n(\mathbf{x}); \qquad \mathbf{x} \in \Gamma - N \tag{3}$$

It is linear in the unknown $\mathbf{p}$. Premultiplying both sides by $\mathbf{c}(\mathbf{x})$ and integrating along the contour Γ, we have

$$S\mathbf{p} = \int_\Gamma \mathbf{c}(\mathbf{x})v_n(\mathbf{x})ds \tag{4}$$

where

$$S = \int_\Gamma \mathbf{c}(\mathbf{x})\mathbf{c}^T(\mathbf{x})ds$$

is called the *gramian* by Brockett [1]. The uniqueness of $\mathbf{p}$ is determined by the rank of S: If $\det S \neq 0$, we get

$$\mathbf{p} = S^{-1}\int_\Gamma \mathbf{c}(\mathbf{x})v_n(\mathbf{x})ds \tag{5}$$

This is the *least-squares solution* for Equation (3). Notice that S is determined by the shape of the contour only, and is independent of the coordinate system chosen to compute it. Thus the singularity of the gramian S is intrinsic. Contours with singular S have poor observability for the motion, therefore we should not use them as the visual cue for recovering the motion. For this reason we will not discuss singular or near singular cases. We will assume that the condition number of S is reasonably far away from zero, in the sence that its smallest eigenvalue is $\gg 0$.

3 Estimation of Normal Velocity

Estimation of normal velocity is based on Equations (2.*a*) and (2.*c*). That is, we measure $d_n(\mathbf{x})$ from the two given contours and treat it as an estimate of $v_n(\mathbf{x})$. However $d_n(\mathbf{x})$ itself can not be measured exactly. Figure 1 shows the true $d_n(\mathbf{x})$ and two estimates $\bar{d}_n(\mathbf{x})$ and $\hat{d}_n(\mathbf{x})$ obtained by the two most commonly used normal velocity estimation procedures[2]. They are as follows.

[2]Since we estimate d_n for the purpose of estimating v_n, such procedures are traditionally called normal velocity (or flow) estimation procedures, but they in fact estimate the normal displacement.

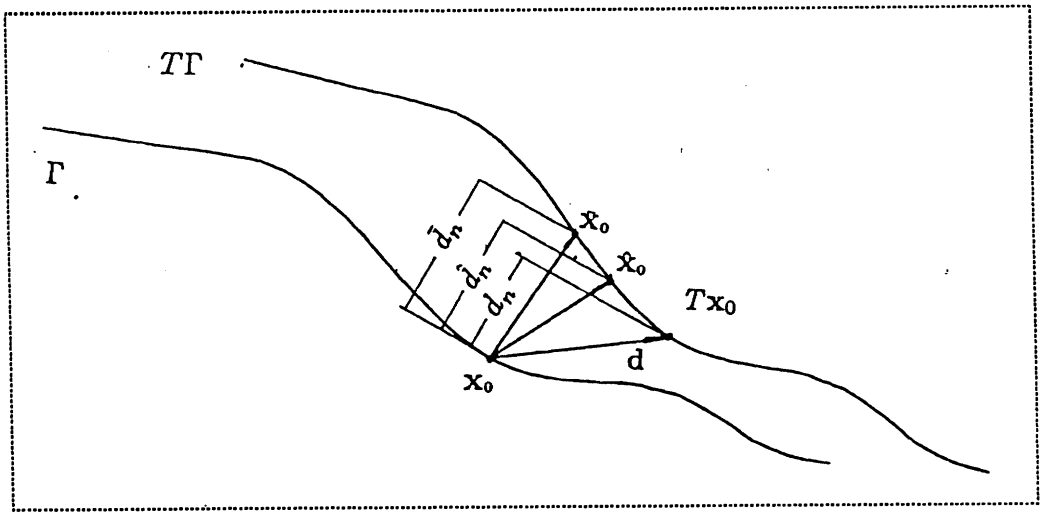


Figure 1: Estimate of v_n.

Definition 1 *Normal Distance Estimation*

$d_n(\mathbf{x})$ *is estimated by* $\bar{d}_n(\mathbf{x})$, *the distance from the point* $\mathbf{x} \in \Gamma$ *to the transformed contour* $T\Gamma$ *in the direction of the normal vector* $\mathbf{v}(\mathbf{x})$.

Definition 2 *Nearest Point Estimation*

$d_n(\mathbf{x})$ *is estimated by*

$$\begin{aligned} \hat{d}_n(\mathbf{x}) &= \mathbf{n}^T(\mathbf{x})(\hat{\mathbf{x}} - \mathbf{x}) \\ \hat{\mathbf{x}} &= \sup_{\hat{\mathbf{x}} \in \{\hat{\mathbf{x}}\}} |\mathbf{n}^T(\mathbf{x})\,(\hat{\mathbf{x}} - \mathbf{x})|. \end{aligned}$$

where $\{\hat{\mathbf{x}}\}$ *is the set of points of* $T\Gamma$ *nearest to* $\mathbf{x}$.

In most cases there is only one point $\hat{\mathbf{x}}$ of $T\Gamma$ nearest to $\mathbf{x}$. Intuitively $\bar{\mathbf{x}} - \mathbf{x}$ is parallel to the normal vetor $\mathbf{n}(\mathbf{x})$ of Γ, and $\hat{\mathbf{x}} - \mathbf{x}$ is parallel to the normal vector $\mathbf{n}(\hat{\mathbf{x}})$ of $T\Gamma$.

The normal distance estimate is frequently used in practice since it requires only a linear search for each point. But it has the disadvantage in that, on some part of the contour, $\bar{d}_n$ may not be well-defined or the estimation error may be large. An example is shown in Figure 2. The part of contour where $\bar{d}_n(\mathbf{x})$ is not well-defined is marked by the dashed line and the part of of the contour where $\bar{d}_n$ is a bad estimate of $d_n(\mathbf{x})$ is marked by the thick line.

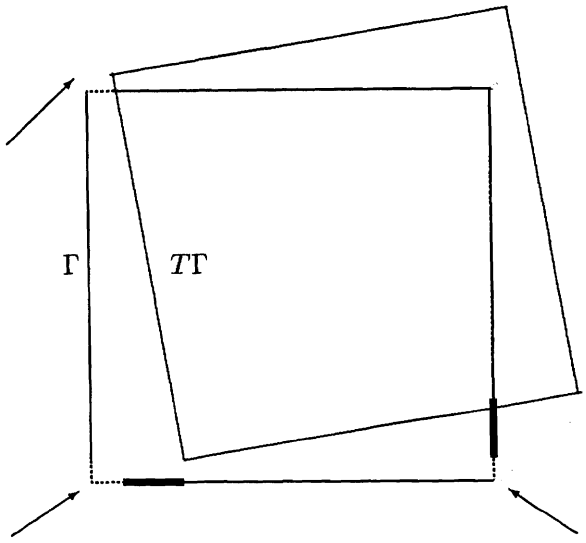


Figure 2: Places where $\bar{d}_n$ is undefined (dashed line) or is a bad estimate of d_n (thick line).

The nearest point estimate requires two-dimensional search, which makes this method unattractive. However, it has an advantage that it is always well-defined wherever the normal vector is well-defined. Another nice property of the nearest point method is that

$$|\hat{\mathbf{x}} - \mathbf{x}| \leq |T\mathbf{x} - \mathbf{x}| \equiv |\mathbf{d}(\mathbf{x})|. \tag{6}$$

So $\hat{d}_n(\mathbf{x})$ satisfies the same inequality as $d_n(\mathbf{x})$ does:

$$\begin{aligned} |\hat{d}_n(\mathbf{x})| &= |\mathbf{n}^T(\mathbf{x})\,(\hat{\mathbf{x}} - \mathbf{x})| \\ &\leq |\hat{\mathbf{x}} - \mathbf{x}| \\ &\leq |\mathbf{d}(\mathbf{x})|. \end{aligned}$$

Consequently, the error involved in the estimation

$$\begin{aligned} |\hat{d}_n(\mathbf{x}) - d_n(\mathbf{x})| &= |\mathbf{n}^T(\mathbf{x})\,(\hat{\mathbf{x}} - \mathbf{x} - T\mathbf{x} + \mathbf{x})| \\ &\leq |\mathbf{n}^T(\mathbf{x})\,(\hat{\mathbf{x}} - \mathbf{x})| + |\mathbf{n}^T(\mathbf{x})\,(T\mathbf{x} - \mathbf{x})| \\ &\leq 2|\mathbf{d}(\mathbf{x})| \end{aligned} \tag{7}$$

is bounded by the value of the displacement itself. For this reason we shall use the nearest point estimate.

Equation (7) shows the upperbound of measurement error at any point along Γ. In fact we can establish the tighter bound by imposing a moderate assumption on the contour geometry. We only

estimate the error for those points on a smooth piece of the contour such that in a neighborhood of the point, the piece of the contour can be estimated as a piece of a second order curve.

Suppose $T\Gamma$ is parameterized by arc length s:

$$T\Gamma: \qquad \begin{bmatrix} x \\ y \end{bmatrix} = F(s) \equiv \begin{bmatrix} \psi(s) \\ \phi(s) \end{bmatrix}.$$

Then we can parameterize Γ as

$$\Gamma: \qquad \begin{bmatrix} x \\ y \end{bmatrix} = T^{-1}F(s).$$

Note that Γ is no longer parameterized by its arc length.

Let $\mathbf{x}_0 = T^{-1}F(s_0)$ be any point on Γ at which Γ is smooth. Then $T\mathbf{x}_0 = F(s_0)$ is the (true) corresponding point on $T\Gamma$. Let $\hat{\mathbf{x}}_0 = F(\hat{s})$ be the estimated corresponding point obtained by the nearest point estimation procedure (See Figure 1.). $T\mathbf{x}_0 - \mathbf{x}_0$ is the true displacement vector at $\mathbf{x}_0$ and $\hat{\mathbf{x}}_0 - \mathbf{x}_0$ is the estimated displacement vector. We call the difference of the two vectors, $\hat{\mathbf{x}}_0 - T\mathbf{x}_0$, the error vector at $\mathbf{x}_0$. These three vectors projected on the unit normal $\mathbf{n}(\mathbf{x}_0)$ of Γ give us the true normal displacement $d_n(\mathbf{x}_0)$, the estimated normal displacement $\hat{d_n}(\mathbf{x}_0)$ and the error $e(\mathbf{x}_0) \equiv \hat{d_n}(\mathbf{x}_0) - d_n(\mathbf{x}_0)$, respectively.

By the definition of the nearest point estimate, $\hat{s}$ should satisfy the following minimum condition:

$$\hat{s} = \arg \min_{s \in T\Gamma} \left\{ [x_0 - \psi(s)]^2 + [y_0 - \phi(s)]^2 \right\} \tag{8}$$

Since $\hat{s}$ is on a smooth piece of contour, we can obtain the minimum by differentiation. Therefore $\hat{s}$ must be a root of the equation

$$(x_0 - \psi(s))\, \dot{\psi}(s) + (y_0 - \phi(s))\, \dot{\phi}(s) = 0 \tag{9}$$

where $\dot{\psi}(s)$ and $\dot{\phi}(s)$ are derivatives of ψ and ϕ with respect to s respectively.

In order to solve Equation (9) we need to know the functions $\psi(s)$ and $\phi(s)$, *i.e.* the shape of the contour in a neighborhood of $\hat{\mathbf{x}}_0$. We assume that $\hat{\mathbf{x}}_0$ is in a small neighborhood of $T\mathbf{x}_0$ and is connected to $T\mathbf{x}_0$ by a smooth piece of the contour. (In this situation, we say that the two points are *locally smoothly connected*). We then estimate this piece of the contour by a second order curve.

To make the computation simple, let us introduce a new coordinate system with the origin at $T\mathbf{x}_0$ and such that the x-axis coincides with the tangent vector $\tau(T\mathbf{x}_0)$ of $T\Gamma$. We re-parameterize $T\Gamma$ starting at the new origin. Then we have

$$T\mathbf{x}_0 = \begin{bmatrix} \psi(0) \\ \phi(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

and the first order derivative, *i.e.* the unit tangent vector, has the form

$$\tau(T\mathbf{x}_0) = \begin{bmatrix} \dot{\psi}(0) \\ \dot{\phi}(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \tag{10}$$

We also can calculate the second order derivatives

$$\ddot{\psi}\,(0) = \frac{d}{ds}\sqrt{1-\dot{\phi}^2(s)}\Big|_0 = (1-\dot{\phi}^2)^{-1/2}\dot{\phi}\,\ddot{\phi}\Big|_0 = 0.$$

$\ddot{\phi}\,(0)$ may not equal to 0. Since $\dot{\psi}(0) = 1$,

$$\ddot{\phi}\,(0) = \dot{\psi}(0)\,\ddot{\phi}\,(0) - \dot{\phi}(0)\,\ddot{\psi}\,(0) = \kappa$$

is the curvature of $T\Gamma$ at the point $T\mathbf{x}_0$. So the best-fitting second order curve at the point $T\mathbf{x}_0$ is:

$$\begin{bmatrix} \psi(s) \\ \phi(s) \end{bmatrix} = \begin{bmatrix} s \\ \frac{1}{2}\kappa s^2 \end{bmatrix}. \tag{11}$$

Substituting Equation (11) back into Equation (9) we have:

$$x_0 - (1-\kappa y_0)s - \frac{1}{2}\kappa^2 s^3 = 0 \tag{12}$$

Assuming κ is not too big, so that $|\kappa s| < 1$, then $|\kappa y_0| < \frac{1}{2}$. Since $x_0, y_0 \ll 1$, the equation can be solved by linear iteration; we get the root

$$\hat{s} = \frac{x_0}{1-\kappa y_0} + \frac{\frac{1}{2}\kappa^2 x_0^3}{(1-\kappa y_0)^4} + \cdots \tag{13}$$

Thus the error vector is,

$$\hat{\mathbf{x}}_0 - T\mathbf{x}_0 = \begin{bmatrix} \psi(\hat{s}) \\ \phi(\hat{s}) \end{bmatrix} = \begin{bmatrix} \hat{s} \\ \frac{1}{2}\kappa\hat{s}^2 \end{bmatrix} + O(\hat{s}^3). \tag{14}$$

$e(\mathbf{x}_0)$, the error in the normal velocity estimate is the inner product of the unit normal $\mathbf{n}(\mathbf{x}_0)$ of Γ and the above vector. In order to get the expression for $\mathbf{n}(\mathbf{x}_0)$, let us look at the tangent vector $\tau(\mathbf{x}_0)$ of Γ. The non-normalized tangent, by definition, is

$$\tau(\mathbf{x}_0) = \frac{d}{ds}(T^{-1}(F(s)))\Big|_{s=0}$$

Since $F(0) = 0$, only the linear part of T^{-1} contributes to $\tau(\mathbf{x}_0)$. Assume

$$T^{-1}\mathbf{x} = \mathbf{b}_0 + A_0\mathbf{x} + \cdots;$$

then we have the non-normalized tangent vector

$$\tau(\mathbf{x}_0) = A_0 \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

Therefore the non-normalized normal vector is:

$$\mathbf{n}'(\mathbf{x}_0) = A_0^{-T} \begin{bmatrix} 0 \\ -1 \end{bmatrix}.$$

Let $A_0 = I + A'$; with $A' = \{a_{ij}\}$, then

$$A_0^{-T} = \frac{1}{|A_0|} \begin{bmatrix} 1 + a_{22} & -a_{21} \\ -a_{12} & 1 + a_{11} \end{bmatrix}.$$

The normalized normal vector is

$$\mathbf{n}(\mathbf{x}_0) = \frac{1}{(1 + a_{11})^2 + a_{21}^2} \begin{bmatrix} a_{21} \\ -1 - a_{11} \end{bmatrix}.$$

The inner product of Equation (14) and $\mathbf{n}(\mathbf{x}_0)$ gives us the error.

$$\begin{aligned} e(\mathbf{x}_0) &= \left[\hat{s} \quad \frac{1}{2}\kappa\hat{s}^2\right] \mathbf{n}(\mathbf{x}_0) + O(\hat{s}^3) \\ &= \frac{\hat{s}a_{21} - \frac{1}{2}\kappa\hat{s}^2(1 + a_{11})}{(1 + a_{11})^2 + a_{21}^2} + O(\hat{s}^3) \end{aligned} \tag{15}$$

Parameters involved in equation (15) are a_{21}, a_{11}, κ and $\hat{s}(x_0, y_0)$ as in Equation (13). Notice $(x_0\ y_0) = T^{-1}F(0) = \mathbf{b}_0$; $e(\mathbf{x}_0)$ is determined by the parameters of the transformation T^{-1} and

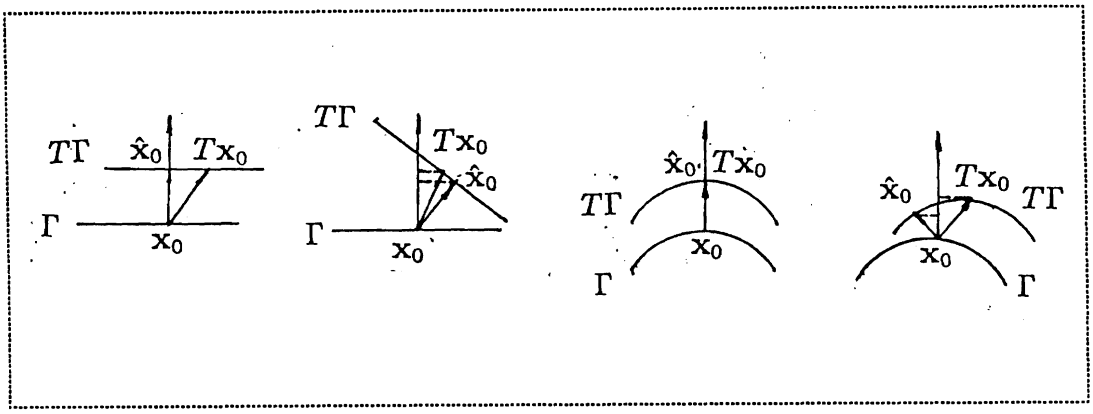


Figure 3: $e(\mathbf{x})$ caused by $d_\tau, a_{n\tau}$ and κ_0. (a) $e(\mathbf{x}) = 0$ since $d_\tau = \kappa_0 = 0$. (b) $e(\mathbf{x}) \neq 0$ due to d_τ and $a_{n\tau}$. (c) $e(\mathbf{x}) = 0$ since $d_\tau = a_{n\tau} = 0$. (d) $e(\mathbf{x}) \neq 0$ due to d_τ and κ_0.

the curvature of $T\Gamma$ at $T\mathbf{x}_0$. By the assumption that the transformation is small, one can observe $a_{21}, a_{11}, x_0, y_0 \ll 1$. So the error can be estimated as

$$e(\mathbf{x}_0) = \frac{a_{21}x_0}{1-\kappa y_0} - \frac{\frac{1}{2}\kappa x_0^2}{(1-\kappa y_0)^2} + O((a_{21}+a_{11})x_0^2 + a_{11}a_{22}x_0 + x_0^3). \tag{16}$$

Furthermore, by the assumption that the curvature κ is not too big (or for given κ that y_0 is small enough), so that $\kappa y_0 \ll 1$, we have

$$e(\mathbf{x}_0) = a_{12}x_0 - \frac{1}{2}\kappa x_0^2 + O((a_{21}+a_{11})x_0^2 + a_{11}a_{21}x_0 + a_{21}x_0y_0 + x_0^2y_0 + x_0^3). \tag{17}$$

The parameters x_0, a_{12} and κ are associated with the transformation T^{-1} and the point $T\mathbf{x}_0$ of $T\Gamma$. Physically $x_0 = [1\ 0](T^{-1}0 - 0)$ is the tangent component of the displacement, and $a_{21} = (\frac{\partial([0\ 1]T^{-1}\mathbf{x})}{\partial x})_{\mathbf{x}=0}$, acts like a local rotation. (See Figure 3). A set of parameters corresponding to $\{x_0, a_{21}, \kappa\}$ but associated with T and $\mathbf{x}_0$ is

$$\begin{cases} d_\tau &= -\tau^T(\mathbf{x}_0)\,(T\mathbf{x}_0 - \mathbf{x}_0) \\ a_{n\tau} &= -\frac{\partial(\mathbf{n}(\mathbf{x}_0)\,T\mathbf{x})}{\partial x}\Big|_{\mathbf{x}=\mathbf{x}_0} \\ \kappa_0 &= \text{curvature of } \Gamma \text{ at } \mathbf{x}_0. \end{cases} \tag{18}$$

We can check that the differences between the corresponding quantities are the third order terms of x_0, y_0 and a_{ij}. So we can replace $\{x_0, a_{21}, \kappa\}$ of Equation (17) with $\{d_\tau, a_{n\tau}, \kappa_0\}$.

We have seen that if the transformation is small in the neighborhood of $\mathbf{x}_0$, then the error in the normal displacement is of high order in the parameters of the transformation, provided that

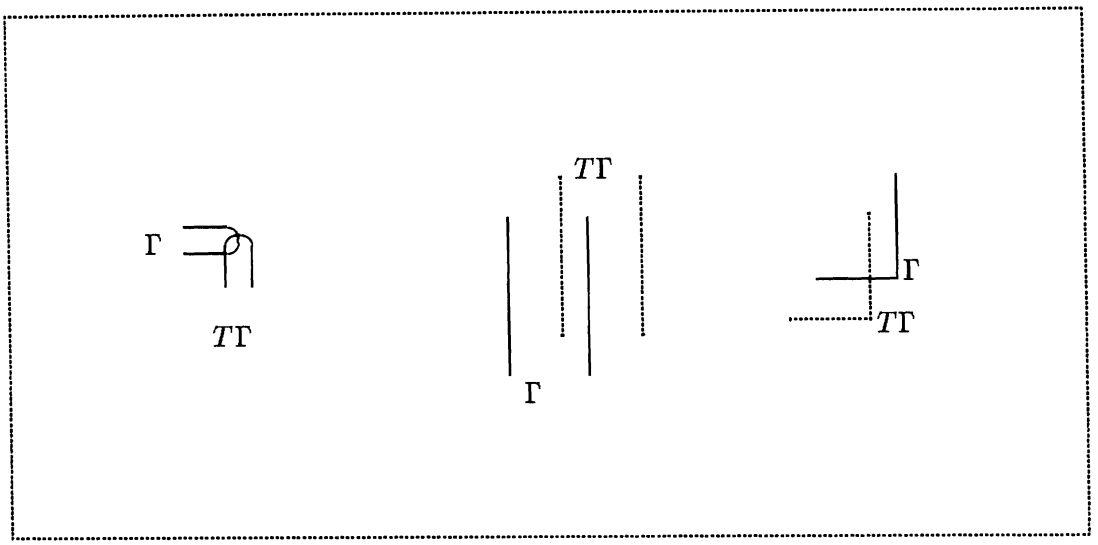


Figure 4: Points of unreliable normal displacement estimation.

1. the true corresponding point $T\mathbf{x}_0$ and approximated corresponding point $\hat{\mathbf{x}}$ are locally smoothly connected, and

2. the curvature of the contour around that point is not too large.

These two conditions typically do not hold everywhere on a contour. (See Figure 4). The places where the above two conditions are violated are

1. Where $T\mathbf{x}_0$ and $\hat{\mathbf{x}}_0$ are connected by a sharp turn such that $\kappa y_0 \approx 1$. Physically, this means that the curve turns around within a neighborhood about the size of the displacement between the two points.

2. Where the contour has two lines too close to each other, so that the displacement along the normal direction of the lines is bigger than the half of the interval between the lines[3].

3. Near a corner of the contour when $T\mathbf{x}_0$ and $\hat{\mathbf{x}}_0$ are separated by the corner.

Of these three categories the first two are motion related, that is, if the motion is small enough, the error estimation of Equation (17) will still hold. Only the third is intrinsic to the contour itself.

[3]This situation may reasonably be expected for the two edges of a bar; in this case the lines may be distinguished by checking direction of the intensity change across the lines.

In this case, Equation (17) simply does not hold at those points where the contour is not smooth. In general we have following theorem;

Theorem 1 *Let $T(t)$ be a transformation, let $d_\tau(t)$, $a_{n\tau}$ and κ_0 be as defined in Equation (18). For any point $\mathbf{x}_0 \in \Gamma$ where $\mathbf{n}(\mathbf{x}_0)$ is defined. there exists a constant time $t_0 > 0$, such that*

$$e(\mathbf{x}_0) = a_{n\tau}(t)d_\tau(t) - \frac{1}{2}\kappa_0 d_\tau^2(t) + O(p^3) \qquad t \in [0, t_0]. \tag{19}$$

where $O(p^3)$ means the third order terms in transformation parameters.

Proof of the above theorem can be found in Appendix A.

Later in Section 5, we will present the iterative algorithm which reduces the error as the iteration continues. For the purpose of discussing its convergence, the exact behavior of error as in Equation (19) will not be necessary; we have

Corollary 1 *Let $\|\mathbf{p}\|$ be the L_∞ norm of $\mathbf{p}$, where $\mathbf{p}(T)$ is the parameter vector as in Equation (5). For any point $\mathbf{x} \in \Gamma - N$, where N is the set contains all non-differentiable points, there exist $p = p(\mathbf{x}) > 0$ and $k(\mathbf{x}, p) > 0$ such that*

$$e(\mathbf{x}, T) \leq k(\mathbf{x}, p)\, \|\mathbf{p}\|^2, \qquad \forall T \in \{T \mid \|\mathbf{p}(T)\| \leq p\}. \tag{20}$$

where $p(\mathbf{x})$ is required to be small enough so that the following conditions are satisfied:

1. *The true correspondence $T\mathbf{x}$ and the estimate correspondence $\hat{\mathbf{x}}$ are locally smoothly connected; by this we mean that they are close enough so that Equation (14) holds, and*
2. *The parameters d_τ, $a_{n\tau}$ and κ_0 are small enough so that Equation (19) holds.*

Since each of $a_{n\tau}$ and d_τ has the order of $\|\mathbf{p}\|$, and κ_0 is a constant, one can easily obtain Equation (20) from Equation (19).

4 Error Formula for the Least-Squares Procedure

The presence of error in the normal velocity measurement will affect the accuracy of the recovered parameter vector $\mathbf{p}$. However, if the transformation is small enough compared to the size of the contour, we know by Theorem 1 that on most of the contour the errors are quite small. For those points where error cannot be estimated by Equation (19), we know from Equation (6) that the error is bounded by a linear combination of the parameters vector $\mathbf{p}$. Therefore after the least-squares procedure we can obtain a good estimate of the transformation parameters.

Suppose the estimated normal velocity at $\mathbf{x}$ is

$$\hat{v}_n(\mathbf{x}) = v_n(\mathbf{x}) + e(\mathbf{x}),$$

where $v_n(\mathbf{x})$ is the true value and $e(s)$ is the error. Replacing $v_n(\mathbf{x})$ in Equation (3) by $\hat{v}_n(\mathbf{x})$ we have

$$\mathbf{c}(\mathbf{x})\mathbf{p} = v_n(\mathbf{x}) + e(\mathbf{x}); \qquad \mathbf{x} \in \Gamma - N$$

In light of the error $e(\mathbf{x})$, the least-sqaures solution for the above equation should become:

$$\hat{\mathbf{p}} = \mathbf{p} + \mathbf{p}_e$$

where $\mathbf{p}$ is the true motion parameter vector and

$$\mathbf{p}_e = S^{-1} \int_\Gamma c(\mathbf{x})e(\mathbf{x})ds \tag{21}$$

is the vector of errors for each component of $\mathbf{p}$.

Before we go on to estimate the magnitude of the error, let us start from a simple case in which a contour undergoes the one-dimensional translation. The motion V is restricted to the x direction only; the true value of V is V_0 and the x component of the unit normal at $\mathbf{x}$ is denoted by $n_x(\mathbf{x})$. So Equation (4) becomes

$$n_x(\mathbf{x})V = n_x(\mathbf{x})V_0 + e(\mathbf{x}); \qquad \mathbf{x} \in \Gamma \tag{22}$$

and the least-squares solution is $\hat{V} = V_0 + V_e$, where V_e is the error term.

$$V_e = S^{-1} \int_\Gamma n_x e(s)ds \tag{23}$$

$$S = \int_\Gamma n_x^2 ds \tag{24}$$

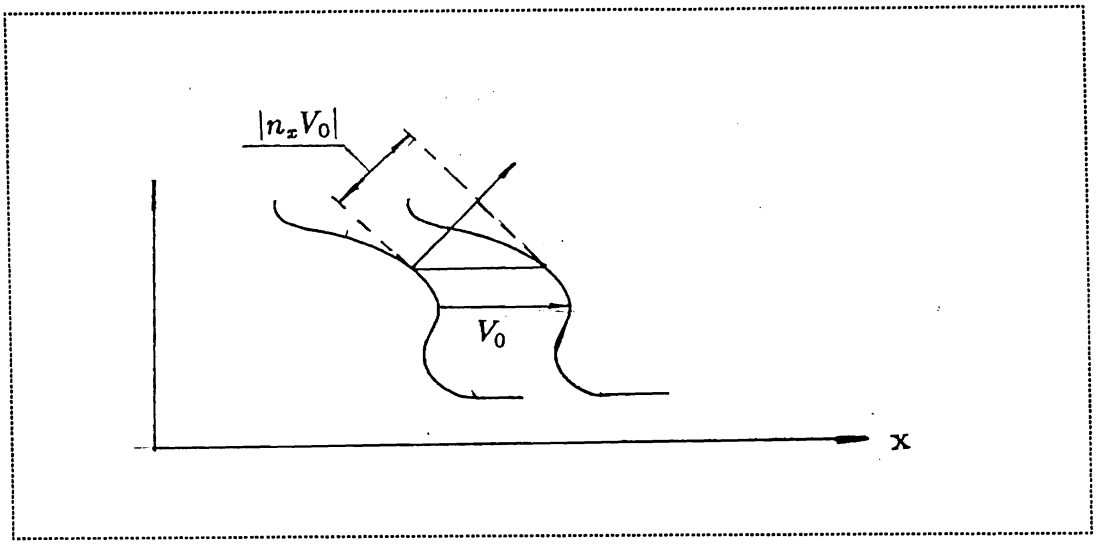


Figure 5: A simple motion - one dimensional translation.

We can see that the translation V_0 is what we want to detect, but we cannot observe it directly. What we can observe is $n_x(\mathbf{x})V_0$ on the right side of Equation (22) with the measurement error $e(\mathbf{x})$. Thus $n_x(\mathbf{x})$ determines the local observability of the translation V_0. For those points with $n_x(\mathbf{x})$ nearly parallel to the direction of translation, $|n_x V_0|/|V_0| \approx 1$, we have a reliable measurement of V_0. For those points with $n_x(\mathbf{x})$ nearly perpendicular to the direction of the translation, $|n_x V_0|/|V_0| \approx 0$, we have a poor observation of V_0. The relative error involved in the observation is $\frac{e(\mathbf{x})}{n_x(\mathbf{x})V_0}$; we can see that for fixed amount[4] of the absolute error $e(\mathbf{x})$, the relative error will be large where the observability is poor.

The gramian $S = \int n_x^2 ds$ describes the observability of the whole contour. If $S = 0$, then $n_x(\mathbf{x}) = 0$ for all $\mathbf{x} \in \Gamma$, and we have no way to detect the translation even if there is one. If $S \neq 0$ but is close to zero, we have a nearly singular situation in which many points on the contour have poor observability, so the relative errors are big. Consequently, the translation detected will also include large errors unless the absolute error $e(\mathbf{x})$ is extremely small. The S^{-1} in Equation (23) reflects this fact.

To estimate the error V_e, we have $|\int n_x e ds| \leq \int |e| ds = \lambda e_M$, where λ is the length of Γ and

[4] $e(\mathbf{x})$ may change while $n_x(\mathbf{x})$ is changing, but there is no indication that $e(\mathbf{x})$ will decrease as $n_x(\mathbf{x})$ decreases. On the contrary, Theorem 1 suggests that decreasing $n_x(\mathbf{x})$ may increase the absolute error $e(\mathbf{x})$ because of the increase in the tangent component of the translation.

$e_M = \sup_{\mathbf{x}\in\Gamma} |e(\mathbf{x})|$. But we anticipate that large errors may occur only on a small part of the contour. So if we can decompose Γ into two groups Γ_g (Subscript g denotes "good".) and Γ_b (Subscript b denotes "bad".) such that Γ_g contains smooth contour segments while Γ_b contains the relatively small portion of contour which is not smooth.

$$\begin{cases} \Gamma = \Gamma_g \cap \Gamma_b \\ \lambda_g = \int_{\Gamma_g} ds \\ e(\mathbf{x}) < e_g \qquad \mathbf{x} \in \Gamma_g \\ \lambda_b = \int_{\Gamma_b} ds \\ e(\mathbf{x}) < e_b \qquad \mathbf{x} \in \Gamma_b \end{cases} \tag{25}$$

for $e_g \ll e_b \leq e_M$ and $\lambda_b \ll \lambda_g$, then we can get a better estimate

$$\left| \int_\Gamma n_x(\mathbf{x}) e(\mathbf{x}) ds \right| \leq \lambda_g e_g + \lambda_b e_b$$

So

$$|V_e| \leq |S^{-1}|(\lambda_g e_g + \lambda_b e_b)$$

Now let us look at the general case of arbitrary image motion, in which the error formula (21) becomes

$$\mathbf{p}_e = \begin{bmatrix} p_{e_1} \\ \vdots \\ p_{e_m} \end{bmatrix} = \left\{ \int_\Gamma \begin{bmatrix} c_1^2(\mathbf{x}) & \cdots & c_1(\mathbf{x})c_m(\mathbf{x}) \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & c_m^2(\mathbf{x}) \end{bmatrix} ds \right\}^{-1} \int_\Gamma \begin{bmatrix} c_1(\mathbf{x}) \\ \vdots \\ c_m(\mathbf{x}) \end{bmatrix} e(\mathbf{x}) ds \tag{26}$$

Let $\hat{c}_i = \sup_{\mathbf{x}\in\Gamma} |c_i(\mathbf{x})|$ ($\hat{c}_i \leq 2$, since $|\mathbf{x}|, |\mathbf{n}| \leq 1$). If the gramian S is diagonal (In the Euclidean motion model, there is always a particular coordinate system so that the S computed in it is diagonal. See [1]), $\mathbf{p}_e$ may be thought as m one-dimensional cases stacked together, and we can estimate the errors as in the one-dimensional case:

$$|p_{e_j}| \leq \left(\int c_j^2(\mathbf{x}) \right)^{-1} \hat{c}_j(\lambda_g e_g + \lambda_b e_b); \qquad j = 1, \cdots, m.$$

For non-diagonal S we will have:

$$|p_{e_j}| \leq \left(\sum_{i=1}^{m} \tilde{s}_{ji} \hat{c}_i \right) (\lambda_g e_g + \lambda_b e_b); \qquad j = 1, \cdots, m. \tag{27}$$

where the $\tilde{s}_{ji}$ are elements of S^{-1}.

The extent to which S is diagonal is not particularly important in error analysis. (In fact, since S is a symmetric matrix, it is always diagonalizable in the space where the parameter vector $\mathbf{p}$ lives. Although by doing so, we may lose the physical meaning of $\mathbf{p}$.) So as long as the original problem is not a nearly singular one, the sum $\sum_{j=1}^{k} \tilde{s}_{ij}\hat{c}_j$ in Equation (27) will not be too large.

5 The Iterative Method for 2-D Motion Recovery

We have seen that in general the parameter vector $\mathbf{p}$ of the 2-D transformation cannot be recovered exactly because of the imperfect measurement on v_n. We have also shown that the error involved in the measurement is small when the two contours are close. In fact, any 2-D motion recovery method which uses a matching procedure has this property. Intuitively, matching is easier and more accurrate when two contours are closer than when they are far away.[5] An extreme case occurs when the two contours are identical. In this case the problem become trivial and we can recover the flow field exactly. Consideration of this case leads us to an iterative method.

5.1 Description of the Method

Suppose two contours Γ and Γ^0 are given. We want to determine the transformation T which will bring the contour Γ to the contour Γ^0, i.e., $\Gamma^0 = T\Gamma$. We begin by making an initial estimate T_0 of T. If $T_0\Gamma \neq \Gamma^0$, we need to make an extra transformation T' posterior to the initial transformation. I.e., $T = T_0T'$, where T' is such that $T'\Gamma = T_0^{-1}\Gamma^0$. The correction term needed is just the motion from Γ to $\Gamma^1(= T_0^{-1}\Gamma^0)$, so that the problem of finding the correction term is that of recovering the transformation from Γ to Γ^1. This is exactly the same problem as we faced in the initial step except that this time Γ^1 is closer to Γ than Γ^0 is, so that we expect to get a better estimate T_1 of T' at this step. This estimate will in turn, move Γ^1 to Γ^2. So, as the iteration progresses, we estimate a sequence of transformations T_i and a generate sequence of contours Γ^i such that $\Gamma^{i+1} = T_i^{-1}\Gamma^i$ and each Γ^i is supposedly closer to Γ than its previous one. Finally, if at a certain step n, Γ^n is identical to Γ (or if the difference between the Γ^n and the Γ is small enough), we stop and integrate all the transformations recovered at eash step together, then report it as the transformation which takes Γ to Γ^0.

[5] Apparantly, this argument does not hold when we are dealing with digitized images.

The whole procedure can be summarized as follows:

1. Initialize $i = 0$.
2. Calculate $\hat{v}_n^{(i)}(\mathbf{x})$, the estimate of normal velocities from Γ to Γ^i.
3. Calculate the transformation $\hat{T}_i$ from $\hat{v}_n^{(i)}(\mathbf{x})$.
4. Use $\hat{T}^i$ to generate $\Gamma^{i+1} = \hat{T}_i^{-1}\Gamma^{(i)}$.
5. Calculate $\hat{v}_n^{(i+1)}(\mathbf{x})$, the estimate of normal velocities from Γ to Γ^{i+1}.
6. If $\int_\Gamma |\hat{v}_n^{(i+1)}| ds$ is small enough, go to step 7. Otherwise increase i, then go to step 3.
7. Output the transformation obtained as $\tilde{T}_i = \hat{T}_0\hat{T}_1 \cdots \hat{T}_i$.

5.2 Some Properties of the algorithm

For any iterative method one must show that

1. the method converges,
2. it converges to the correct solution, and
3. it converges effectively.

In this section we address these issues.

5.2.1 Terminating condition

The iteration stops whenever the solution is found. This property is summarized as the following proposition.

Proposition 1 *If at step j of the iterative procedure, the estimate $\hat{T}_j$ is exact,* i.e. $\hat{T}_j\Gamma = \Gamma^j$, *then*

1. *Further iteration will not alter the result.* i.e. $\tilde{T}_i = \tilde{T}_j$ *for all* $i > j$.
2. *The final estimate $\tilde{T}_j = T$ — the true transformation which carries Γ to Γ_0.*

Proof:

1. $\Gamma^{(j+1)} = \hat{T}_j^{-1}\Gamma^j = \Gamma \implies v_n^{(j+1)}(\mathbf{x}) = 0 \implies \mathbf{p}^{(j+1)} = 0 \implies \hat{T}_{j+1} = T_I$. By deduction $\hat{T}_i = T_I$ for all $i > j$, therefore $\tilde{T}_j = \tilde{T}_i$.

2. $\Gamma = \hat{T}_j^{-1}\Gamma^j = \hat{T}_j^{-1}\hat{T}_{j-1}^{-1}\cdots\hat{T}_0^{-1}\Gamma^0 = \tilde{T}_j^{-1}\Gamma^0 \implies \tilde{T}_j\Gamma = \Gamma^0$. Since Γ and Γ^0 are non-degenerate, the transformation which carries Γ to Γ_0 is unique. Therfore $\tilde{T}_j = T$.

■

5.2.2 Convergence

In Appendix B we have calculated explicitly the iterative procedure for a square undergoing (1) pure translation and (2) pure rotation to demonstrate that the method converges to the correct solution.[6] We observe that the error $e(\mathbf{x})$ of $\hat{v}_n(\mathbf{x})$ satisfies the following:

1. Large errors occur only at the pieces of the contour near corners. The total length of these pieces is on the order of the tramsformation parameters themselves.

2. For the rest of the contour the error is bounded by terms which are second-order in the transformation parameters. This agrees with Equation (20).

Therefore by Equation (27) the error in the parameter vector after the least-squares estimation is also second-order in the transformation parameters. The errors both in the normal flow vector and in the parameter vector will decrease monotonically. At the same time the total length of the bad contour segments will decrease, thereby accelerating the convergence rate.

Of course, the method may diverge or converge to the wrong solution if the motion is considerably larger than the size of contour (or the size of neighborhood in which the least-squares procedure is applied). Unfortunately, there is no single formular to predict how large the radius of converegence is. This issues is discussed in detail in Appendix C, along with the proof of convergence for arbitrary motion and contour. It was shown that in most cases the method converges if the maximal motion is smaller than the one eigth of contour size. So the error eventually goes to zero as the iterative procedure continues and the iterative procedure converges to the correct solution.

5.2.3 Computational complexity

A single iteration consists of two steps; one to measure the normal flow, the other to solve the least-squares problem. Let n be the number of contour points inside the neighborhood where the

[6] You may want to read the appendix at this point.

full flow is modeled as polynomials, and n_p be the size of $\mathbf{p}$. For the affine model $n_p = 6$, and for the second-order model $n_p = 12$. For each point, finding the nearest point in the second image requires two dimensional search of $O(m^2)$, where m is the searching radius. Since there are n points, the time complexity of the first step is given as $O(nm^2)$.

For the second step, it takes $O(nn_p^2)$ to construct the pseudo-inverse matrix, followed by $O(n_p^3)$ to decompose it into triangular matrices using the Gaussian elimination. The backsubstitution of the inverse matrix needs $O(n_p^2)$. Hence, it takes $O(n_p^3)$ to solve the least-squares problem. However one can notice that, after the first iteration, the matrix inversion is not required since the system matrix which is determined by the contour geometry does not change, and a single operation of matrix multiplication ($O(n_p^2)$) and backsubstitution ($O(n_p^2)$) complete the least-squares procedure.

Overall, the first iteration is the most expensive because (i) m, the searching radius has to be large, and (ii) the matrix inversion must be performed. Each additional iteration takes $O(nm^2+n_p^2)$.

5.3 Experimental Results

The performance of the iterative transformation method was studied with an example shown in Figure 6. Figure 6.a and Figure 6.b show two synthetic images of contours generated by a non-trivial 3-D motion of a planar surface containing two ellipses. The ideal velocity field for this motion is calculated and shown in Figure 6.c. In Figure 6.d the normal velocity vectors $v_n^0(\mathbf{x})$ along the contour are measured from the input contours. From $v_n^0(\mathbf{x})$, the full velocity field $\mathbf{v}^0(\mathbf{x})$ is calculated by the velocity functional method, which is shown in Figure 6.e. Notice that this velocity field is quite different from the ideal velocity field in Figure 6.c. The velocity field obtained after three iterations is shown in Figure 6.f. Each flow vector was recovered within 5 percents of accuracy in the relative error.

In the next test performed at the Robotics Laboratory, Harvard University, the robot-held camera traveled with $\mathbf{V} = (0.6, -0.25, 0.4)$ *units/frame* and $\Omega = 0$ *degrees/frame* (Figure 7.a and Figure 7.b). The slope of table was measured approximately as $p = -12$ *degrees* and $q = 55$ *degrees* with respect to the camera coordinates. Zerocrossing operator was used to extract contours [7]. The estimated flow is shown in Figure 7.c. The 3-D motion parameters were recovered from the flow field using the algorithm developed by Wohn and Wu [11]. Their algorithm computes the 3-D parameters (three translations, three rotations and two slope parameters) from the first order

spatio-temporal derivative of flows. Assuming that the entire image consists of a single planar surface, 3-D parameters computed from this flow field are: $\mathbf{V} = (0.53, -0.18, 0.47)$ $units/frame$ and $\Omega = (0.021, -0.031, 0.021)$ $degrees/frame$, $p = -18.53$ $degrees$ and $q = 52.92$ $degrees$.

The third experiment was conducted at the GRASP Laboratory, University of Pennsylvania. Figures 8 show four consecutive frames of the campus scene. The camera "flies" over the 33'rd and Walnut Street with the translational velocity of $(42.43, 30.0, 42.43)$ $mm/frame$, while keeping the pitch angle constant ($= -45$ $degrees$) with respect to the ground. The distance from the camera to the ground was measured as 700 mm with respect to the first camera position (frame 1). Figure 9.a shows the contour image obtained from frame 1 (Figure 8.a). The normal flow vectors measured between frame 1 and frame 2 are shown in Figure 9.b. The full flows after four iterations are shown in Figure 9.c. They were interpolated over the entire image plane for display purpose only. Figure 9.d shows the full flows at frame 4.

The ideal flow near the fovea, according to the motion parameters used in the experiment, is given as follow;

- At frame 2

$$\begin{aligned} v_x &= -0.060614, & \tfrac{\delta v_x}{\delta t} &= -0.007348, \\ v_y &= -0.042857 - 0.042857x + 0.060614y, & \tfrac{\delta v_y}{\delta t} &= -0.005196 \end{aligned}$$

- At frame 3

$$\begin{aligned} v_x &= -0.068182, & \tfrac{\delta v_x}{\delta t} &= -0.009298, \\ v_y &= -0.048701 - 0.048701x + 0.068182y, & \tfrac{\delta v_y}{\delta t} &= -0.006641 \end{aligned}$$

The actual flow obtained from the images is given as

- At frame 2

$$\begin{aligned} v_x &= -0.059307 - 0.005889x - 0.006868y, & \tfrac{\delta v_x}{\delta t} &= -0.007556, \\ v_y &= -0.044259 - 0.052978x + 0.057710y, & \tfrac{\delta v_y}{\delta t} &= -0.005373 \end{aligned}$$

- At frame 3

$$\begin{aligned} v_x &= -0.069855 - 0.002285x - 0.001433y, & \tfrac{\delta v_x}{\delta t} &= -0.010894 \\ v_y &= -0.052227 - 0.074892x + 0.054162y, & \tfrac{\delta v_y}{\delta t} &= -0.007866 \end{aligned}$$

The flows were evaluated with respect to the normalized image plane (focal length = 1). The 3-D motion parameters obtained from the above measurements are

- At frame 2

$$\begin{aligned} V &= (31.91, 22.69, 35.23)\ mm/frame \\ \Omega &= (-0.6789, 0.7865, 0.1628)\ degree/frame \\ p &= -57.11\ degrees, \quad q = -12.02\ degrees \end{aligned}$$

- At frame 3

$$\begin{aligned} V &= (39.62, 28.60, 39.41)\ mm/frame \\ \Omega &= (-0.3312, 0.3172, 0.6979)\ degree/frame \\ p &= -53.48\ degrees, \quad q = -11.95\ degrees \end{aligned}$$

The translational velocity reported above is the rescaled quantity with respect to the known distance. We suspect that the large portion of the error in the 3-D parameter estimates is due to the poor camera calibration. Notice that the solutions at frame 2 and at frame 3 are somewhat consistent, although they disagree with the input parameters.

6 Concluding Remark

We proposed an iterative algorithm for recovering a 2-D motion which would account for the transformation of one contour into another. The nearest point method was used for the normal flow estimation because it is provably accurate for small displacements and it leads to a bounded estimation error in any case. The least-squares procedure serves as a basic module and is iteratively applied to the normal flow estimates. The estimated value of the full flow is adjusted under the assumption of *convected invariance* of contour, and the estimated value of the full flow is projected onto the normal direction so as to estimate the normal flow more accurately. The entire process is repeated until the least-squares error no longer falls off significantly. We showed that the method converges to the solution under reasonably weak conditions: local affinity of motion and smoothness of the contour.

The entire process may be viewed as a cooperative process in which inhibitory and excitatory mechanisms co-exist. Each flow vector is described locally by flow vectors in the form of polynomials, while being constrained to satisfy exact matching between contours. The inhibition and excitatory processes do not agree initially, but as the iteration progresses, they reach a point at which the disagreement is insignificant. The error was shown to vanish under suitable assumptions described in the paper.

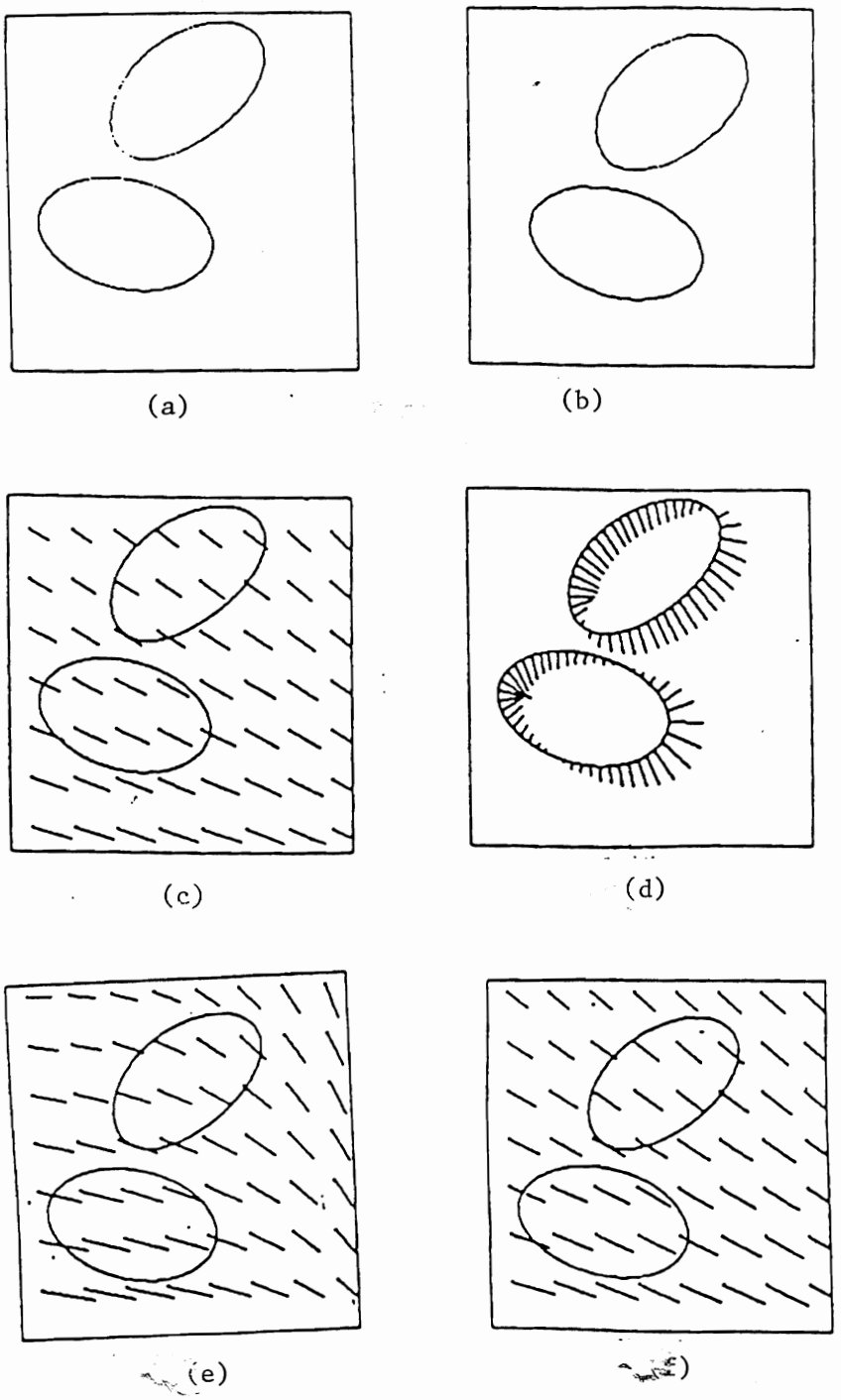
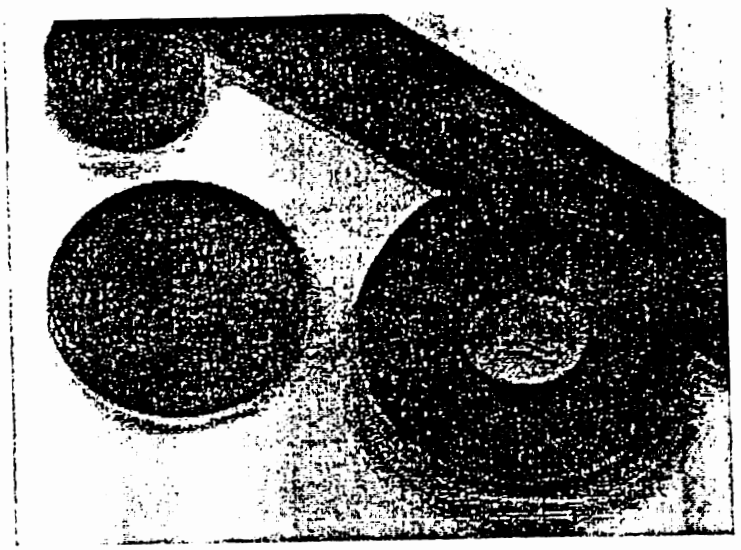
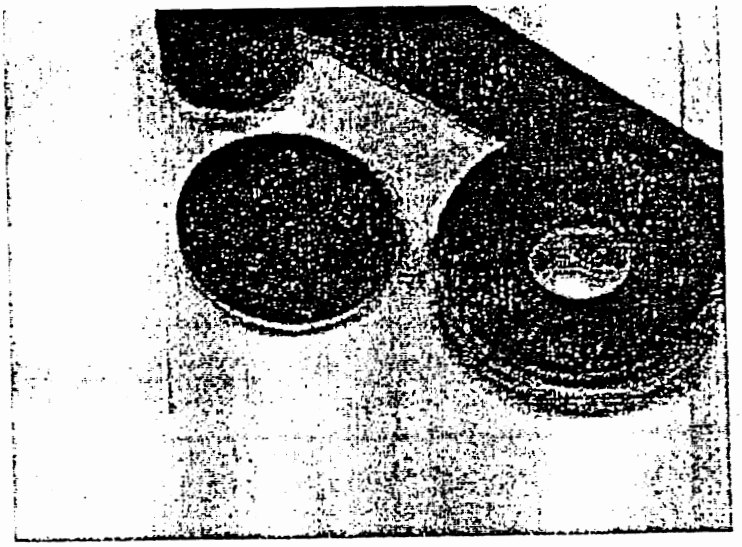


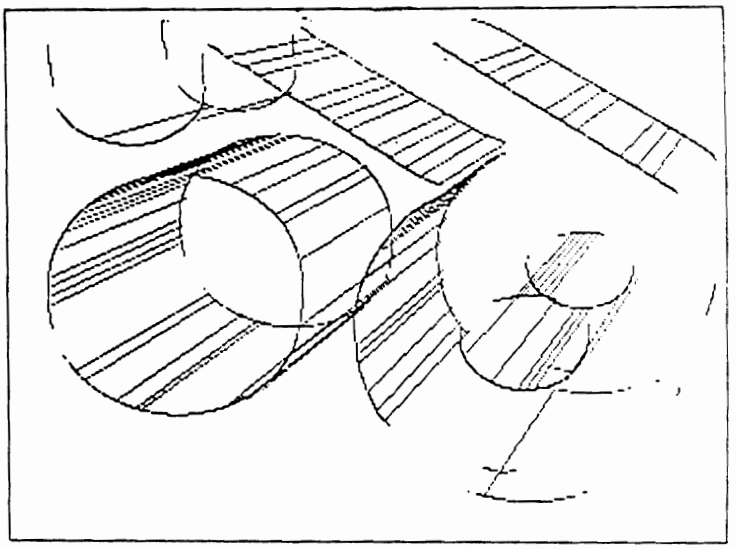
Figure 6: Recovering 2-D motion (synthetic images). (a) frame 1. (b) frame 2. (c) the ideal flow. (d) the normal flow measured. (e) the full flow recovered (0 iteration). (f) the full flow recovered (after three iterations).



(a)



(b)



(c)

Figure 7: Recovering 2-D motion (real images). (a) frame 1. (b) frame 2. (c) full flow recovered (after three iterations).

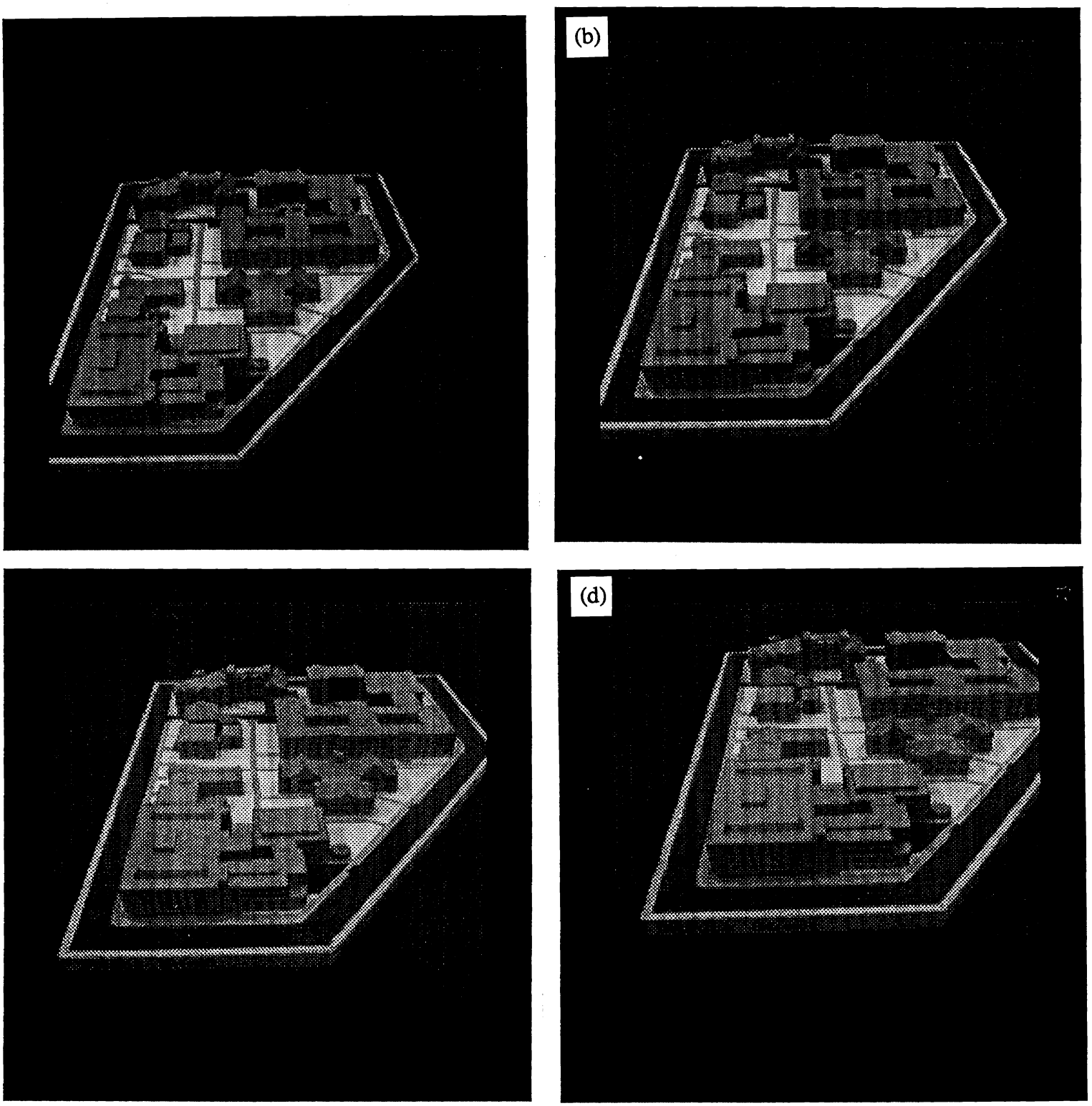


Figure 8: Image sequence used in the experiment (four frames are shown). (a) frame 1. (b) frame 2. (c) frame 3. (d) frame 4.

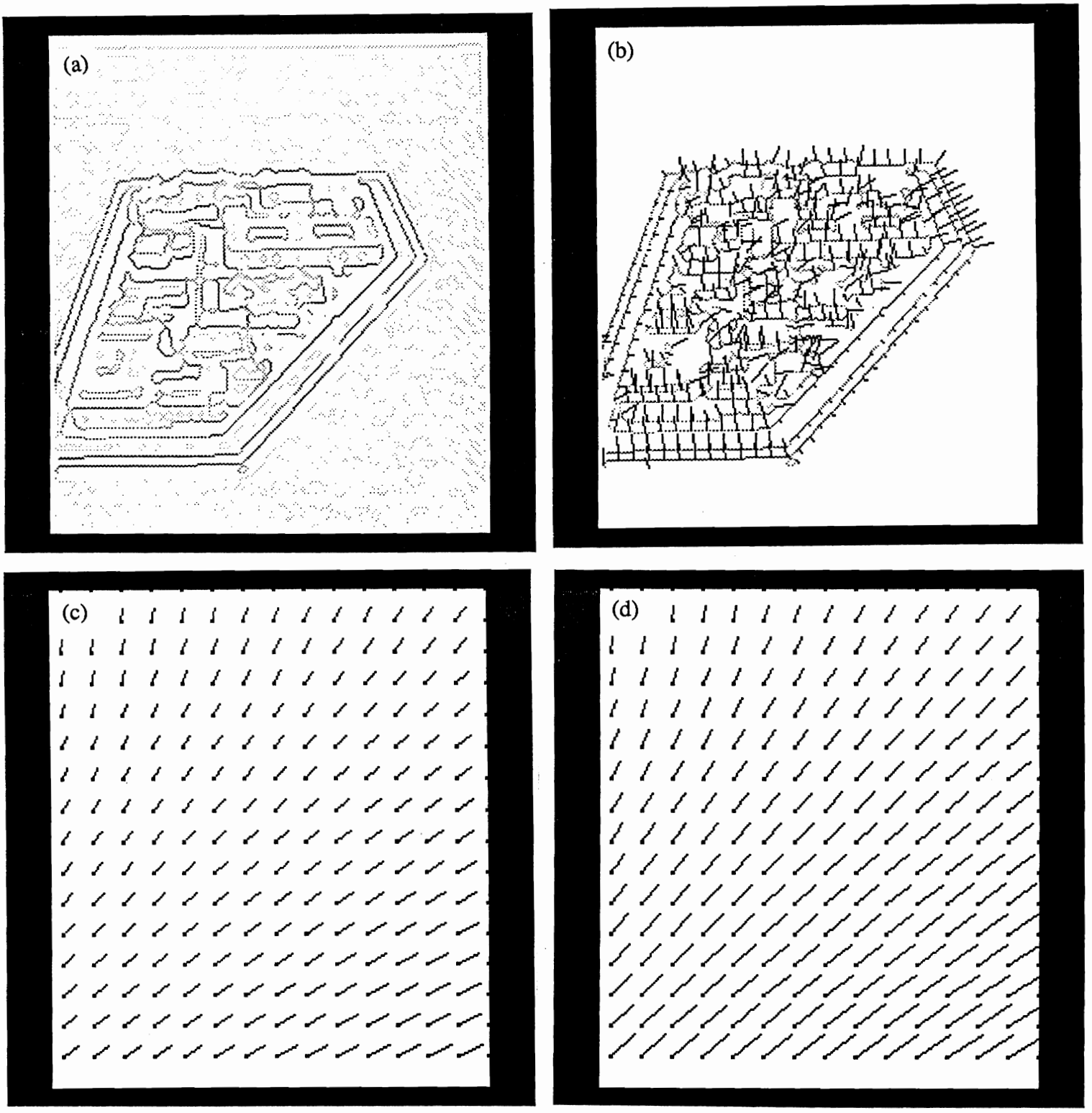


Figure 9: Output obtained from the iterative method. (a) zerocrossing contours at frame 1. (b) normal flows measured along contours. (c) full flows recovered and interpolated (after four iterations). (d) full flows recovered and interpolated at frame 4.

References

[1] R. Brockett, "Grammians, Generalized Inverses, and the Least Squares Approximation of Optical Flow", Proc. *IEEE Conf. Robotics and Automation*, Raleigh, 1987.

[2] D. Heeger, "A Model for the Extraction of Image Flow", Tech. Report, SRI Intl., Feb., 1986.

[3] E. C. Hildreth, "Computation Underlying the Measurement of Visual Motion", *Artificial Intelligence* **23**, No. 3, 309-355, 1984.

[4] B. K. P. Horn and B. G. Schunck, "Determining Optical Flow", *Artificial Intelligence* **17**, 185-204, 1981.

[5] J. J. Koenderink and A. J. van Doorn, "Invariant Properties of the Motion Parallax Field due to the Movement of Rigid Bodies Relative to an Observer", *Optica Acta* **22**, 773, 1975.

[6] H. Longuet-Higgins and K. Prazdny, "The Interpretation of a Moving Retinal Image", *Proc. Royal Soc. London* **B 208**, 385-397, 1980.

[7] D. Marr and E. C. Hildreth, "Theory of Edge Detection", *Proc. Royal Soc. London* **B 207**, 187-217, 1980.

[8] D. Marr and S. Ullman, "Directional Selectivity and its Use in Early Visual Processing", *Proc. Royal Soc. London* **B 211**, 151, 1981.

[9] A. M. Waxman, and S. Ullman, "Surface Structure and 3-D Motion From Image Flow: A Kinematic Analysis", *Intl. Journal of Robotics Research* **4**, 72-94, 1985.

[10] A. M. Waxman and K. Wohn, "Contour Evolution, Neighborhood Deformation and Global Image Flow: Planar Surfaces in Motion", *Intl. Journal of Robotics Research* **4**, 95-108, 1985.

[11] K. Wohn and J. Wu, "3-D Motion Recovery from Time-varying Optical Flows" *Proc. AAAI-86*, 670-675, August 1986.

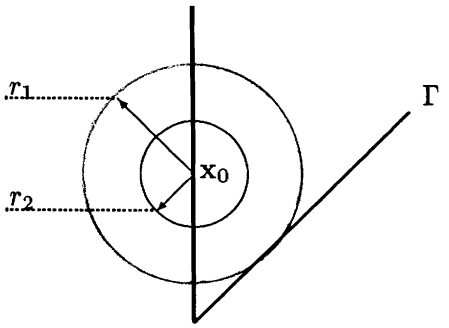
Appendix A Proof of Theorem 1

Let $B(r)$ be a circle centered at $\mathbf{x}_0$ with radius r. For any $\mathbf{x}_0$ on a smooth piece of contour, there exists a $r_1 > 0$ such that any $\mathbf{x} \in B(r_1) \cap \Gamma$ is connected to $\mathbf{x}_0$ by a smooth piece of contour. (Figure A.1.) Choose $0 < r_2 < r_1$ and $t_1 > 0$ such that

$$|T(t)\mathbf{x} - \mathbf{x}| \leq |\mathbf{x} - \mathbf{x}_0| - r_2, \qquad \forall(\mathbf{x}, t) \in (\Gamma - B(r_1) \cap \Gamma) \times [0, t_1].$$

So $T(t)\mathbf{x} \in B(r_2) \Longrightarrow \mathbf{x} \in B(r_1)$ for all $t \leq t_1$, *i.e.*, only those in $B(r_1)$ may be transformed into $B(r_2)$.

Choose $0 < t_2 \leq t_1$ such that $|T\mathbf{x}_0(t) - \mathbf{x}_0| \leq r_2$ for all $t \in [0, t_2]$. Then according to Equation (6), $\hat{\mathbf{x}}(t)$ obtained by the nearest point estimation procedure must satisfy $|\hat{\mathbf{x}}(t) - \mathbf{x}_0| < r_2$, i.e $\hat{\mathbf{x}} \in B(r_2)$. Therefore $T^{-1}(t)\hat{\mathbf{x}} \in B(r_1)$, and $T^{-1}\hat{\mathbf{x}}$ and $\mathbf{x}_0$ are locally connected by a smooth piece of contour. Equivalently, $\hat{\mathbf{x}}$ and $T\mathbf{x}_0$ are also locally smoothly connected. So Equation (15) holds. If we choose $0 < t_0 \leq t_2$ such that all transformation parameters $\ll 1$, and $\kappa y_0 \ll 1$, we have Equation (17) for $t \in [0, t_0]$. Replacing x_0, a_{21} and κ by the set of parameters defined by Equation (18) we get Equation (19). ■



1) Only those points inside $B(r_1)$ may be transformed into $B(r_2)$.
2) $\mathbf{x}_0$ is moving inside $B(r_2)$. So $\hat{\mathbf{x}}$ and $T\mathbf{x}_0$ are locally smoothly connected.

Figure A.1: Proof of Theorem 1

Appendix B Iterative Procedure — a Square Under 2-D Euclidean Motion

In this Appendix we will make an explicit computation of the iterative 2-D motion recovery procedure for a square undergoes (1) translation and (2) rotation.

Γ in this case is a 2 by 2 square bounded by the lines $x = 1, x = -1, y = 1$ and $y = -1$. (See Figure A.2).

The length λ of the Γ is 8. The 2-D transformation parameter vector and the coefficient vector are

$$\mathbf{c}(\mathbf{x}) = \begin{bmatrix} n_x \\ n_y \\ xn_y - yn_x \end{bmatrix}; \qquad \mathbf{p} = \begin{bmatrix} V_x \\ V_y \\ \Omega \end{bmatrix}$$

So the gramian

$$S = \int_\Gamma \mathbf{c}(\mathbf{x})\mathbf{c}^T(\mathbf{x})ds = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & \frac{8}{3} \end{bmatrix}$$

is non-singular. The least-squares solution for $\mathbf{p}$ is

$$\hat{\mathbf{p}} = \begin{bmatrix} \hat{V}_x \\ \hat{V}_y \\ \hat{\alpha} \end{bmatrix} = S^{-1} \int_\Gamma \begin{bmatrix} n_x \\ n_y \\ xn_y - yn_x \end{bmatrix} v_n ds$$

A.1 Pure translational case

Suppose the true parameter vector is

$$\mathbf{p}(T) = \begin{bmatrix} V_x \\ V_y \\ 0 \end{bmatrix}$$

We assume $V_x, V_y > 0$. Figure A.3 shows both the original square Γ and the transformed square Γ^0.

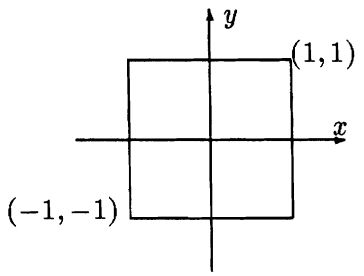


Figure A.2: Γ — a square.

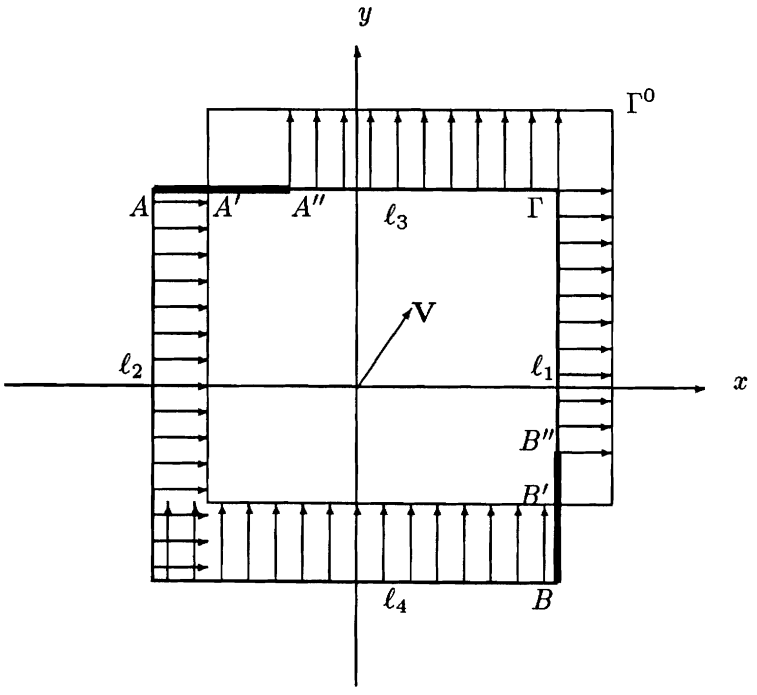


Figure A.3: Γ, Γ^0 and $\hat{v}_n(\mathbf{x})$

By the nearest point estimation method we estimate the normal displacements from Γ to Γ^0 as

:

$$\hat{v}_n^{(0)}(x,y) = \begin{cases} V_x & x = 1, \quad y \in (V_x + V_y - 1, 1) \\ 0 & x = 1, \quad y \in (-1, V_x + V_y - 1) \\ V_x & x = -1, \quad y \in (-1, 1) \\ V_y & y = 1, \quad x \in (V_x + V_y - 1, 1) \\ 0 & y = 1, \quad x \in (-1, V_x + V_y - 1) \\ V_y & y = -1, \quad x \in (-1, 1) \end{cases}$$

We can see that on part of the contour Γ (drawn as thin lines in Figure A.3) the normal displacement can be measured exactly, while on the other part of the contour (drawn as thick lines in in Figure A.3) the estimated normal displacement is wrong. We can see that the decomposition of $\Gamma = \Gamma_g^0 \cup \Gamma_b^0$ is

$$\begin{cases} \Gamma_b^0 &= \{\overline{AA''}, \overline{BB''}\} \\ \Gamma_g^0 &= \Gamma - \Gamma_b^0. \end{cases}$$

with

$$\begin{cases} \lambda_g^0 &= 4 - 2(V_x + V_y) \\ e^0(\mathbf{x}) &= e_g^0(\mathbf{x}) = 0, \qquad \mathbf{x} \in \Gamma_g \\ \lambda_b^0 &= 2(V_x + V_y) \\ e^0(\mathbf{x}) &= e_b^0(\mathbf{x}) = \begin{cases} -V_x & \mathbf{x} \in \{\overline{AA'}, \overline{A'A''}\} \\ -V_y & \mathbf{x} \in \{\overline{BB'}, \overline{B'B''}\}. \end{cases} \end{cases}$$

The relative length of Γ_b^0 is

$$\beta \equiv \frac{\lambda_b^0}{\lambda} = \frac{V_x + V_y}{4}.$$

β is proportional to the translation, and the amount of error in Γ_b is also proportional to the translation.

The transformation parameter vector estimated from $\hat{v}_n^{(0)}$ is

$$\mathbf{p}(\hat{T}_0) = \begin{bmatrix} \hat{V}_x^{(0)} \\ \hat{V}_y^{(0)} \\ 0 \end{bmatrix} = S^{-1} \int_\Gamma \begin{bmatrix} n_x \\ n_y \\ x n_y - y n_x \end{bmatrix} \hat{v}_n^{(0)} ds = \begin{bmatrix} (1-\beta)V_x \\ (1-\beta)V_y \\ 0 \end{bmatrix}$$

Using $\hat{T}_0$ we can generate Γ^1 as the dashed square in Figure A.4

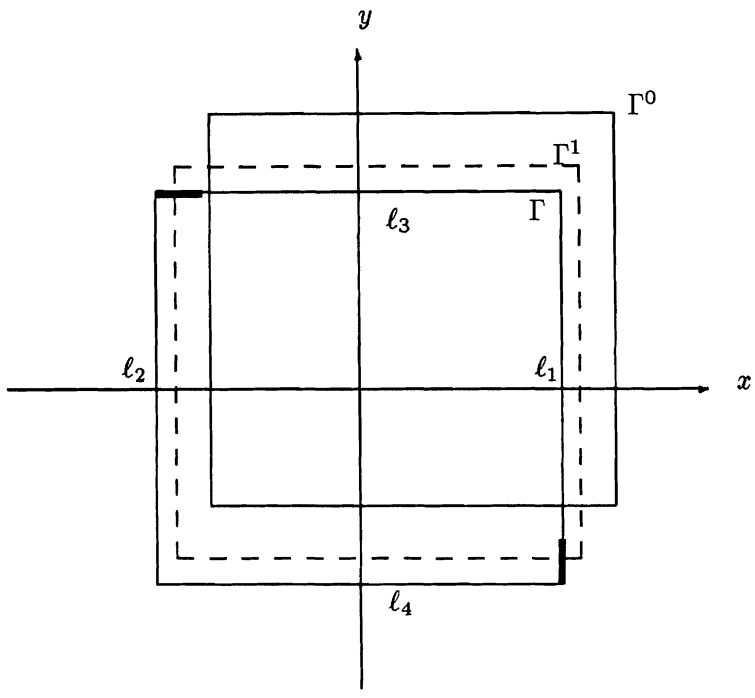


Figure A.4: Γ, Γ^1 and Γ^0

The procedure for calculating $\hat{v}_n^1$ and T_1 is same as that for calculating v_n^0 and T_0, except that this time we are working on Γ and Γ^1 instead of Γ and Γ^0. Since Γ^1 is much closer to Γ than Γ^0 is, we obtain a better estimate at this step

The exact transformation from Γ to Γ^1 is

$$\beta \begin{bmatrix} V_x \\ V_y \\ 0 \end{bmatrix}$$

So the relative length of Γ_b^1 is

$$\frac{\lambda_b^1}{\lambda} = \frac{\beta(V_x + V_y)}{4} = \beta^2$$

i.e. Γ_b^1 (the thick lines in Figure A.3) is shorter than Γ_b^0. $|e_b^1(\mathbf{x})|$ is also smaller than $|e_b^0(\mathbf{x})|$. We can write out the motion at this step immediately:

$$\mathbf{p}(\hat{T}_1) = \begin{bmatrix} (1-\beta^2)\beta V_x \\ (1-\beta^2)\beta V_y \\ 0 \end{bmatrix} = \begin{bmatrix} (\beta-\beta^3) V_x \\ (\beta-\beta^3) V_y \\ 0 \end{bmatrix}$$

Therefore

$$\mathbf{p}(\tilde{T}_1) = \mathbf{p}(\hat{T}_0\hat{T}_1) = \begin{bmatrix} (1-\beta^3)V_x \\ (1-\beta^3)V_y \\ 0 \end{bmatrix}$$

Continuing this iterative procedure we can get:

$$\mathbf{p}(\tilde{T}_i) = \begin{bmatrix} (1-\beta^{2^{i+1}-1})V_x \\ (1-\beta^{2^{i+1}-1})V_y \\ 0 \end{bmatrix}$$

So if $\beta = \frac{V_x+V_y}{4} < 1$, then

$$\lim_{i\to\infty} \mathbf{p}(\tilde{T}_i) = \begin{bmatrix} V_x \\ V_y \\ 0 \end{bmatrix} = \mathbf{p}(T).$$

converges to the true motion parameters quickly.

A.2 Pure rotational case

Suppose the true parameter vector is:

$$\mathbf{p}(T) = \begin{bmatrix} 0 \\ 0 \\ \alpha \end{bmatrix}$$

We assume $0 < \alpha \ll 1$.

Figure A.5 shows the both original square Γ and the rotated square Γ^0.

Since the shape is symmetric,

$$\int_\Gamma \begin{bmatrix} n_x \\ n_y \\ xn_y - yn_x \end{bmatrix} ds = \begin{bmatrix} 0 \\ 0 \\ 4\int_{\ell_1}(nn_y - yn_x)ds \end{bmatrix},$$

so we only need study $\int_{\ell_1}(xn_y - yn_x)ds$, where ℓ_1 is the right side of the square.

On ℓ_1 we have:

$$\hat{v}_n^0(\mathbf{x}) = \begin{cases} -\sin\alpha\,\cos\alpha\,y + \cos\alpha\,(1-\cos\alpha); & y \in (-a, 1) \\ \sin\alpha\,\cos\alpha\,y + \sin\alpha\,(1-\sin\alpha); & y \in (-1, -a) \end{cases}$$

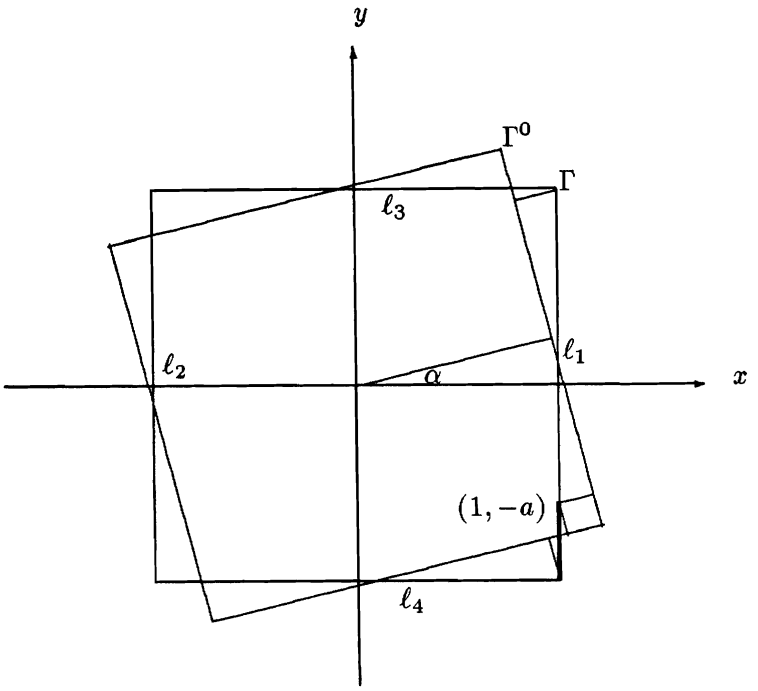


Figure A.5: Rotation case, Γ and Γ^0

where

$$a = \frac{\cos\alpha - \sin\alpha}{\cos\alpha + \sin\alpha}.$$

While the true normal displacement on ℓ_1 is

$$v_n(\mathbf{x}) = -\sin\alpha\, y - 1 + \cos\alpha\,.$$

So if we decompose $\ell_1 = \gamma_g \cup \gamma_b$, then

$$\begin{cases} \lambda_g^0 &= 1 + a \\ e_g(\mathbf{x}) &= \sin\alpha\,(1 - \cos\alpha\,)y + 1 - (\cos\alpha\,)^2 \\ \lambda_b^0 &= 1 - a \\ e_b(\mathbf{x}) &= \sin\alpha\,(1 + \cos\alpha\,)y + 1 - \cos\alpha + \sin\alpha\,(1 - \sin\alpha\,). \end{cases}$$

Substituting a into the above equations and expanding in powers of α we get

$$\begin{cases} \lambda_g^0 &= 2 - 2\alpha + O(\alpha^2) \\ e_g(\mathbf{x}) &= \alpha^2 + O(\alpha^3), \\ \lambda_b^0 &= 2\alpha + O(\alpha^2) \\ e_b(\mathbf{x}) &= (2y+1)\alpha + O(\alpha^2). \end{cases}$$

so the relative lengths are:

$$\begin{aligned} \frac{\lambda_g^0}{\lambda} &\approx 1-\alpha \\ \frac{\lambda_b^0}{\lambda} &\approx \alpha \end{aligned}$$

and average errors over Γ_b^0, Γ_b^0 respectively:

$$\begin{aligned} \bar{e}_g &\equiv \frac{\int_{\Gamma_g} |e(\mathbf{x})| ds}{\lambda_g^0} \approx \alpha^2, \\ \bar{e}_b &\equiv \frac{\int_{\Gamma_b} |e(\mathbf{x})| ds}{\lambda_b^0} \approx 3\alpha. \end{aligned}$$

Integrating $(xn_y - yn_x)\hat{v}_n^{(0)}(\mathbf{x})$ over ℓ_1,

$$\begin{aligned} \int_{\ell_1} (xn_y - yn_x) v_n^{(0)}(\mathbf{x}) ds &= -\int_{-1}^{-a} (ky^2 + by) dy - \int_{-a}^{1} (ky^2 + cy) dy \\ &= \frac{2}{3}ka^3 + \frac{1}{2}(b-c)(1-a^2) \end{aligned}$$

where

$$\begin{aligned} k &= \sin\alpha \cos\alpha, \\ b &= \sin\alpha (1 - \sin\alpha), \\ c &= \cos\alpha (1 - \cos\alpha). \end{aligned}$$

The least-squares solution for the rotation parameter α is

$$\begin{aligned} \hat{\alpha}_0 &= 4S^{-1} \int_{\ell_1} (xn_y - yn_x) v_n^{(0)}(\mathbf{x}) ds \\ &= ka^3 + \frac{3}{4}(b-c)(1-a^2). \end{aligned}$$

Substituting for k, a, b and $, c$ and expanding in powers of α,

$$\begin{aligned} \hat{\alpha}_0 &= \alpha - 3\alpha^2 + \frac{41}{6}\alpha^3 + \cdots \\ &= \alpha - 3\alpha^2 + O(\alpha^3). \end{aligned}$$

Again we can see the error E_0 is of second order in the transformation parameter α.

As the iteration procedure goes on, the rotations estimated are:

$$\begin{aligned} \hat{\alpha}_1 &= 3\alpha^2 - 3(3\alpha)^2 + \cdots \\ \tilde{\alpha}_1 &= \alpha - \tfrac{1}{3}(3\alpha)^4 + \cdots \end{aligned}$$

and

$$\tilde{\alpha}_i = \alpha - \frac{1}{3}(3\alpha)^{2^{i+1}}.$$

If $\alpha < \frac{1}{3}$,

$$\lim_{i \to \infty} \tilde{\alpha}_i = \alpha$$

converges to the true value of the rotation angle.

Appendix C On Convergence of the Iterative Method

In order to make calculation tractable, let us assume that the transformation T is an affine transformation $T\mathbf{x} = (I + A)\mathbf{x} + b$ with $A = \{a_{ij}\}$ and $b = [b_1, b_2]^T$. Denote

$$\begin{cases} T &= \{A, b\} \\ \mathbf{p}(T) &= [a_{11}, a_{12}, a_{21}, a_{22}, b_1, b_2]^T \end{cases} \tag{A.1}$$

where $\mathbf{p}(T) \in \Re^6$ is the parameter vector. Let $\|\mathbf{p}\|$ be the L_∞ norm of $\mathbf{p}$:

$$\|\mathbf{p}\| = \max\{a_{11}, a_{12}, a_{21}, a_{22}, b_1, b_2\}$$

By a straightforward calculation we have the following equations and inequalities for any two transformations $T_1 = T(\mathbf{p}_1) = \{A_1, b_1\}$ and $T_2 = T(\mathbf{p}_2) = \{A_2, b_2\}$:

$$\begin{cases} T_1T_2 &= \{A_1 + A_2 + A_1A_2,\ b_1 + b_2 + A_1b_2\} \\ T_1^{-1}T_2 &= \{(I + A_1)^{-1}(A_2 - A_1),\ (I + A_1)^{-1}(b_2 - b_1\} \\ T(\mathbf{p}_1 + \mathbf{p}_2) &= \{A_1 + A_2,\ b_1 + b_2\} \\ 1 + 2\|\mathbf{p}(T_1T_2)\| &\leq (1 + 2\|\mathbf{p}(T_1)\|)\ (1 + 2\|\mathbf{p}(T_2)\|) \end{cases} \tag{A.2}$$

Therfore, if we define

$$\tilde{T}_n = \hat{T}_0\hat{T}_2 \cdots \hat{T}_n.$$

Then

$$1 + 2\|\mathbf{p}(\tilde{T}_n)\| \leq \prod_{i=0}^{n}(1 + 2\|\mathbf{p}(\hat{T}_i)\|).$$

By the definition of T_i, $T = \hat{T}_0\hat{T}_1 \cdots \hat{T}_{i-1}T_i = \tilde{T}_{i-1}\hat{T}_i^{-1}T_i$, so

$$\tilde{T}_{i-1} = T\ T_i^{-1}\hat{T}_i. \tag{A.3}$$

Let

$$\mathbf{E}_i = \mathbf{p}(\hat{T}_i) - \mathbf{p}(T_i)$$

be the error vector at the ith step of the iterative procedure, we have

Theorem 2 *The iterative procedure converges to the true transformation if and only if* $\lim_{i\to\infty} E_i = 0$.

Proof : Let $\{A_E, b_E\}$ be the transformation corresponding to $\mathbf{E}_i$ according to Equation(A.1). Suppose $T_i = \{A, b\}$, then $\hat{T}_i = \{A + A_E, b + b_E\}$. By Equation (A.2),

$$T_i^{-1}\hat{T}_i = \{(I + A)^{-1}A_E, (I + A)^{-1}b_E\} \tag{A.4}$$

By using Equations (A.3) and (A.4), we have

$$\lim_{i\to\infty} \mathbf{E}_i = 0 \iff \lim_{i\to\infty} T_i^{-1}\hat{T}_i = T_I \iff \lim_{i\to\infty} \tilde{T}_i = T.$$

■

In showing the convergence, we shall prove $\mathbf{E}_i$ goes zero as $i \to \infty$ for the entire contour segment under consideration. Recalling that Corollary 1 in Section 3 is valid for each point, we need some sort of global anlysis. The question we are posing is the following: given a contour segment, determine the minimal motion under which the iteration converges. The argument proceeds much in the same philosophy as the proof procedure for the uniform convergence of a infinite series of functions within a certain interval.

Suppose the contour has at least one corner or intersection. It is clear that there is no uniform p such that for all $\mathbf{x} \in \Gamma - N$ the first condition is satisfied. However if the contour is smooth enough, for example — all derivatives at its smooth points are uniformly bounded — then there is a uniform p such that the second condition holds.

Definition 3 *(Uniformity)*
A contour is uniform if there exists a constant $p > 0$ such that for any $T \in \{T|\mathbf{p}(T) \leq p_\Gamma\}$, the following two conditions are satisfied.

1. *For all $\mathbf{x}, \mathbf{x}' \in \Gamma - N$ which are locally smoothly connected, $T\mathbf{x}$ and $T\mathbf{x}'$ are also locally smoothly connected.*

2. *For all $\mathbf{x} \in \Gamma - N$, if $T\mathbf{x}$ and $\hat{\mathbf{x}}$ are locally smoothly connected, then*

$$e(\mathbf{x}, T) \leq k_\Gamma \, \|\mathbf{p}(T)\|^2.$$

where k is a constant independent of $\mathbf{x}$.

For this type of contours, large error can occur only at those points $\mathbf{x}$ which are not locally smoothly connected to $T^{-1}\hat{\mathbf{x}}$. For small enough T, this situation can only occur near a corner or an intersection.

For a given contour Γ under a given transformation T, define the maximum displacement as

$$d_\Gamma(T) = \sup_{\mathbf{x}\in\Gamma} |T\mathbf{x} - \mathbf{x}|. \tag{A.5}$$

A straightforward calculation shows that there exists a constant $D_\Gamma > 0$ depending only on Γ such that for any T,

$$d_\Gamma(T) \leq D_\Gamma |\mathbf{p}(T)| \tag{A.6}$$

Define

$$B_{\mathbf{x}}(r) = \{\mathbf{x}' \in \Gamma \mid |\mathbf{x} - \mathbf{x}'| < r\}.$$

For every $\mathbf{x} \in \Gamma - N$, let $r' \in \{r'\}_{\mathbf{x}}$ be such that for every $\mathbf{x}' \in B_{\mathbf{x}}(r')$, $\mathbf{x}$ and $\mathbf{x}'$ are locally smoothly connected. Define

$$\rho_{\mathbf{x}} = \sup_{r'\in\{r'\}_{\mathbf{x}}} r'. \tag{A.7}$$

Then $B_{\mathbf{x}}(\rho_{\mathbf{x}})$ is the maximal open ball in which all points are locally smoothly connected to $\mathbf{x}$.

We would like to show that if the point under consideration is sufficiently far away from singular points, $T\mathbf{x}$ and $\hat{\mathbf{x}}$ are locally connected.

Lemma 1 *Let Γ be a uniform contour and let $\mathbf{x} \in \Gamma - N$. If $\rho_{\mathbf{x}} > 2d_\Gamma(T)$, then the true correspondence $T\mathbf{x}$ and the estimate $\hat{\mathbf{x}}$ are locally smoothly connected.*

Proof : By Equation (6)

$$|\mathbf{x} - T^{-1}\hat{\mathbf{x}}| \leq |\mathbf{x} - \hat{\mathbf{x}}| + |\hat{\mathbf{x}} - T^{-1}\hat{\mathbf{x}}| \leq 2d_\Gamma(T) < \rho_{\mathbf{x}}.$$

So $\mathbf{x}$ and $T^{-1}\hat{\mathbf{x}}$ are locally amoothly connected. By Definition 3, $T\mathbf{x}$ and $\hat{\mathbf{x}}$ are also locally smoothly connected. ■

Definition 4 *(n-junction)*

A point $w \in \Gamma$ is a n-junction if $w \in N$ and there are n curves starting at w.

By this definition, an intersection of two curves is a 4-junction, a corner is a 2-junction and an end point is a 1-junction.

Define J to be the set $J = \{n\text{-junction};\ n \geq 2\}$.

Define $C_{\mathbf{x}}(r)$ to be the boundary of $B_{\mathbf{x}}(r)$,

$$C_{\mathbf{x}}(r) = \{\mathbf{x}' \in \Gamma \mid |\mathbf{x} - \mathbf{x}'| = r\}.$$

Define $\check{\rho}$ for a subset Γ' of Γ

$$\check{\rho}(\Gamma') = \min_{\mathbf{x} \in \Gamma'} \rho_{\mathbf{x}}. \tag{A.8}$$

where $\rho_{\mathbf{x}}$ is as defined in Equation (A.7).

Denote the length of Γ' as $\lambda(\Gamma')$.

Let $w \in J$ be a n-junction. It is clear that there exists $r_w > 0$ such that for any $0 < r \leq r_w$, $C_w(r)$ contains exactly n points. We also have

$$\begin{cases} \lambda(B_w(r_1)) \leq \lambda(B_w(r_2)), & r_1 \leq r_2 \\ \lim_{r \to 0} \check{\rho}(C_w(r)) = \lim_{r \to 0} \lambda(B_w(r)) = 0 \\ \check{\rho}(C_w(r)) \leq r < 2r \leq nr \leq \lambda(B_w(r)). \end{cases} \tag{A.9}$$

Definition 5 *(Separability)*

We say an n-junction $w \in J$ is separable if there exists r_w such that

1.
$$\check{\rho}(C_w(r_1)) \leq \check{\rho}(C_w(r_2)), \qquad r_1 \leq r_2 \leq r_w \tag{A.10}$$

2. *There exists ℓ_w,*
$$\frac{\lambda(B_w(r))}{\check{\rho}(C_w(r))} \leq \ell_w, \qquad r \leq r_w. \tag{A.11}$$

The first condition prevents $\check{\rho}(C_w(r))$ from having infinitely many oscillations as $r \to 0$, and the second condition means that we can use rays to separate branches.

Examples:

1. A n-junction all of whose branches are rays is separable, since $\check{\rho}(C(r)) = r \tan \alpha$, where α is the smallest angle between any two lines, and
$$\frac{\lambda(B(r))}{\check{\rho}(C(r))} = \frac{nr}{r \tan \alpha} = \frac{n}{\tan \alpha}.$$

2. A 2-junction at the origin consisting of two branches $\{y = 0, x \geq 0\}$ and $\{y = x^2, x \geq 0\}$ is not separable, since

$$\frac{\lambda(B(r))}{\check{\rho}(C(r))} \geq \frac{2r}{r^2} = \frac{2}{r} \stackrel{r \to 0}{\longrightarrow} \infty.$$

3. A 2-junction at the origin consisting of two branches $\{x = 0, y \geq 0\}$ and $\{y = x \sin \frac{1}{x}, x \geq 0\}$ is not separable, since the the latter branch has infinitely many oscillations near 0.

Theorem 3 *Let Γ be a uniform contour all of whose junctions are separable. Then there exists $p > 0$ such that for all $T \in \{T \mid \|\mathbf{p}(T)\| \leq p\}$, we can decompose Γ as $\Gamma = \Gamma_g(T) \cap \Gamma_b(T)$ with*

$$\begin{cases} \lambda(\Gamma_g(T)) = 1 - k_1\|\mathbf{p}(T)\| \\ e(\mathbf{x}) \leq k_2\|\mathbf{p}(T)\|^2; \quad \mathbf{x} \in \Gamma_g(T) \\ \lambda(\Gamma_b(T)) = k_1\|\mathbf{p}(T)\| \\ e(\mathbf{x}) \leq k_3\|\mathbf{p}(T)\|; \quad \mathbf{x} \in \Gamma_b(T) \end{cases} \tag{A.12}$$

where k_1, k_2 and k_3 are constants.

Proof : Since all junctions of Γ are separable, we can choose r_0 for all $w \in J$ such that for all $r \leq r_0$, *(i)* if w is a n-junction, then $C_w(r)$ contains n points; and *(ii)* Equations (A.10) and (A.11) hold. Let

$$\Gamma' = \cup_{w \in J} B_w(r_0),$$

and

$$d = \check{\rho}(\Gamma - \Gamma') \tag{A.13}$$

Let D_Γ be the constant as in Equation(A.6). Choose $p > 0$ such that

$$p = \frac{d}{2D_\Gamma}. \tag{A.14}$$

Then for any $T \in \{T \mid \|\mathbf{p}(T)\| \leq p\}$, choose for every $w \in J$ a $r_w(T)$ such that

$$\check{\rho}(C(r_w(T))) = 2D_\Gamma\|\mathbf{p}(T)\|.$$

Let

$$\begin{cases} \Gamma_b(T) = \cup_{w \in J} B_w(r_w(T)) \\ \Gamma_g(T) = \Gamma - \Gamma_b. \end{cases}$$

We shall show that $\check{\rho}(\Gamma_g(T)) \geq 2d_\Gamma(T)$. Since $\Gamma_g(T) = (\Gamma - \Gamma') \cup \Gamma''$, where

$$\Gamma'' = \cup_{w \in J}(B_w(r_0) - B_w(r_w)),$$

by Equations (A.13) and (A.14), $\check{\rho}(\Gamma - \Gamma') = 2D_\Gamma p \geq 2D_\Gamma \|\mathbf{p}(T)\|$. By Equations (A.10) $\check{\rho}(\Gamma'') \geq \check{\rho}(C(r_w(T))) = 2D_\Gamma \|\mathbf{p}(T)\|$. So

$$\check{\rho}(\Gamma_g(T)) \geq 2D_\Gamma \|\mathbf{p}(T)\| \geq 2d_\Gamma(T).$$

Therefore by Lemma 1, for every $\mathbf{x} \in \Gamma_g(T)$, $T\mathbf{x}$ and $\hat{\mathbf{x}}$ are locally smoothly connected. Since Γ is uniform, there exists $k_2 = k_\Gamma$ such that

$$e(\mathbf{x}) \leq k_\Gamma \|\mathbf{p}(T)\|^2;\ \mathbf{x} \in \Gamma_g(T)$$

The length of $\Gamma_b(T)$ can be estimated by using Equation (A.9):

$$\lambda(\Gamma_b(T)) \leq \sum_{w \in J} \ell_w \rho_w(r_w(T)) = 2D_\Gamma \|\mathbf{p}(T)\| \sum_{w \in J} \ell_w.$$

i.e. $k_1 = 2D_\Gamma \sum_{w \in J} \ell_w$. It is obvious that $k_3 = 2D_\Gamma$. Thus we obtain Equation (A.12). ■

Corollary 2 *Let Γ be a uniform contour all of whose junctions are separable. Then there exists $p > 0$ such that, if at i^{th} step of the iterative procedure we have $\|\mathbf{p}(T_i)\| < p$, then the iterative procedure converges to the true transformation.*

Proof : By Theorem 3 there exists p' such that for all $\|\mathbf{p}(T_i)\| < p'$ we can decompose $\Gamma = \Gamma_g^i \cap \Gamma_b^i$ with Equation (A.12) holding. Therefore by the error formula (Equation (27)) the error $\mathbf{E}_i$ of $\mathbf{p}(T_i)$ after the least-squares procedure satisfies the inequality $\mathbf{E}_i \leq k \|\mathbf{p}(T_i)\|^2$.

Choose $p = \min\{\frac{1}{8}, \frac{1}{8k}, p'\}$, then for all $\|\mathbf{p}(T_i)\| < p$, we have

$$\mathbf{E}_i \leq k\|\mathbf{p}(T_i)\|^2 \leq \frac{1}{8}\|\mathbf{p}(T_i)\|. \tag{A.15}$$

As in Equation (A.4), we can calculate

$$T_{i+1} = T_i \hat{T}_i^{-1} = \{-(I + A + A_E)^{-1} A_E,\ -(I + A + A_E)^{-1} b_E\}$$

So

$$\|\mathbf{p}(T_{i+1})\| \leq \frac{3}{8}\|\mathbf{p}(T_i)\| \leq p. \tag{A.16}$$

Therefore if we let i_0 be the smallest i such that Equations (A.15) and (A.16) hold, then by induction, the equations are true for all $i > i_0$. So for all $i > i_0$.

$$\begin{aligned} 0 \leq \mathbf{E}_i &\leq \frac{1}{8}\|\mathbf{P}(T_i)\| \leq \frac{1}{8}\frac{3}{8}\|\mathbf{P}(T_{i-1})\| \leq \cdots \\ &\leq \frac{1}{8}\left(\frac{3}{8}\right)^{i-i_0-1} \|\mathbf{P}(T_{i_0+1})\| \\ &\leq \frac{1}{8}\left(\frac{3}{8}\right)^{i-i_0} p. \end{aligned}$$

Thus $\lim_{i\to\infty} \mathbf{E}_i = 0$. By Theorem 2, the iterative procedure converges to the true transformation. ■