

Computing Spatiotemporal Surface Flow #

by

**Mark Allmen
and
Charles R. Dyer**

**Computer Sciences Technical Report #935
May 1990**



Computing Spatiotemporal Surface Flow

Mark Allmen

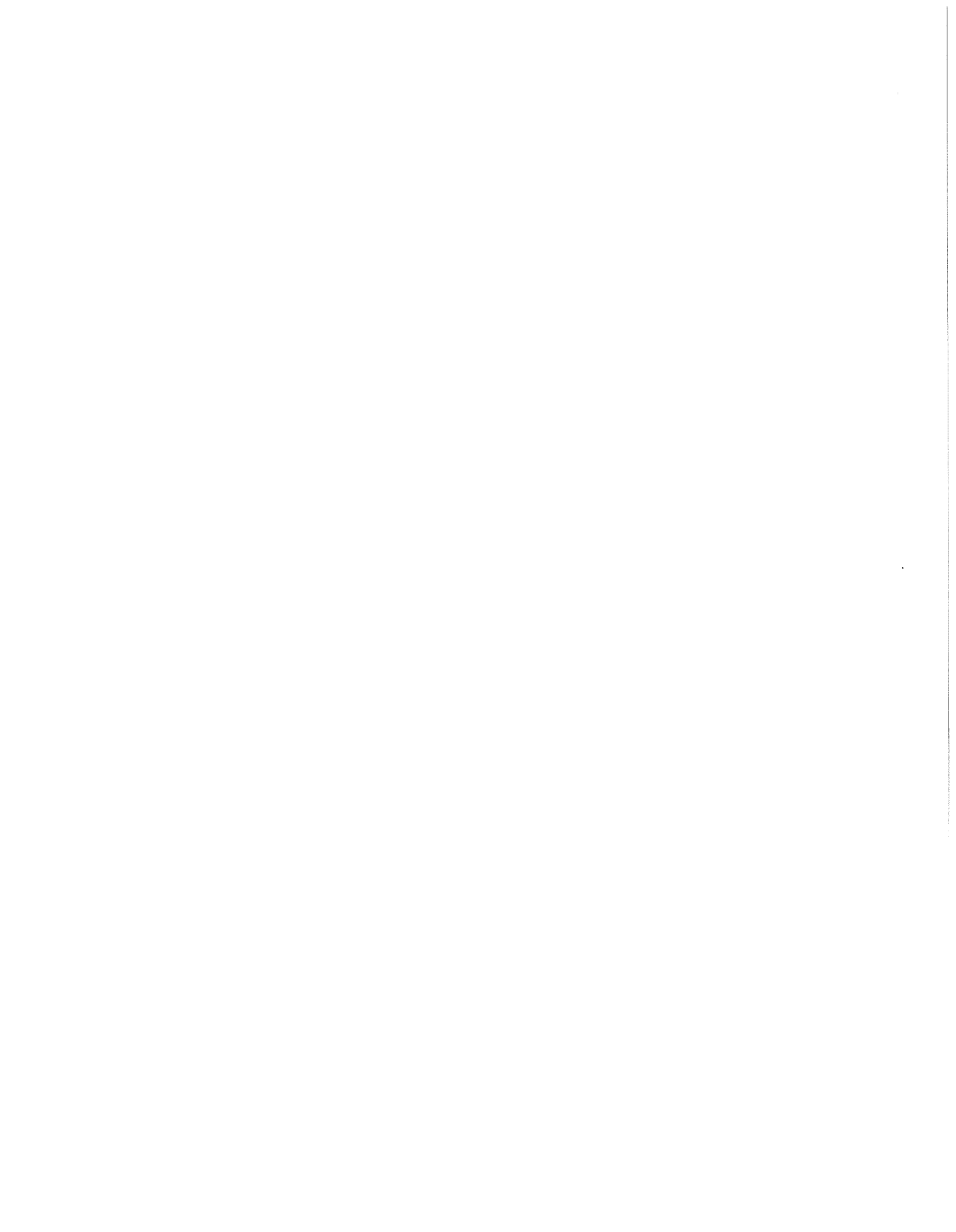
Charles R. Dyer

Computer Sciences Department
University of Wisconsin
Madison, Wisconsin 53706

Abstract

Spatiotemporal surface flow is the natural extension of optic flow to spatiotemporal surfaces. For every point on a surface, the instantaneous velocity of that point on the surface is recovered. This paper presents a method for making full use of the information available in a spatiotemporal surface. Using the low-level structure of spatiotemporal surfaces, translational and rotational motion parallel to the image plane is recovered. Rather than use the partial derivatives of the surface as done in most gradient-based optic flow methods, we use the gradient of a function on the spatiotemporal surface. It is observed that arc length of a contour does not change if that contour is moved in the direction of motion on the surface. Motivated by this observation, a function measuring arc length change is defined on a spatiotemporal surface. The direction of motion of a contour undergoing motion parallel to the image plane is shown to be perpendicular to the gradient of this function. It is also shown that this gradient approximates the direction of motion when object motion in the scene is not parallel to the image plane. This method is used to compute the spatiotemporal surface flow in sequences of edge images and gray level images.

The support of the National Science Foundation under Grant No. IRI-8802436 is gratefully acknowledged.



1. Introduction

The structure of spatiotemporal (ST) surfaces reveals a great deal of information about contours in a scene and their motion. As contours move, their projection into the image also moves. Over time, these projections sweep out spatiotemporal surfaces and volumes created by the surfaces. Thus, these surfaces and volumes are direct representations of object motion. If the images are edge maps, ST surfaces are 2-surfaces of edge points in a three-dimensional volume. Alternatively, gray level images can be used resulting in a four dimensional volume (x-y-time-gray level) containing 3-surfaces. In both cases, the local structure of the surfaces gives information about contours and the contours' motion.

The interpretation of ST surfaces must begin with the recovery of the motion that generated the surface. Figure 1a shows an ST surface formed by a circle rotating about a point on the circle. At any point on the surface there is a "twisting" shape as a result of the circular motion. The arrows on the surface indicate the motion of the contour at a point in space-time. If a contour is translating, as in Figure 1b, the direction of motion remains the same over time and a cylindrical ST surface results. The arrows show the *flow*, or local motion, of points on the contour that generated the ST surface. The flow at a point on an ST surface is called the *spatiotemporal surface flow*.

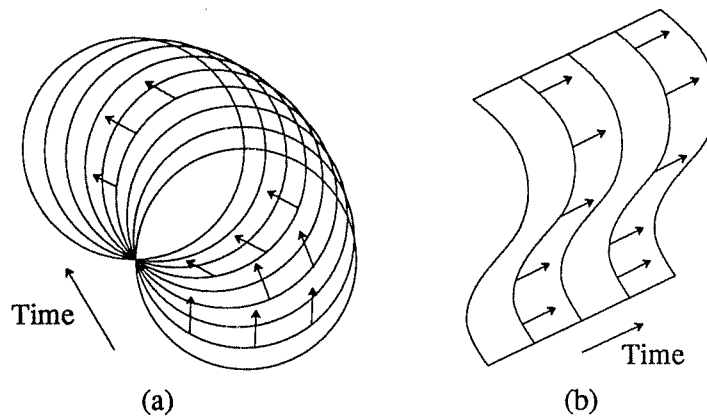


Figure 1. The spatiotemporal surfaces resulting from a circle rotating about a point on the circle (a) and a contour translating (b). The arrows show the local structure of the surfaces.

ST surfaces represent low-level motion, the motion of a point, as well as high-level motion, the motion of objects and camera motion. This can also be viewed as representing local and global motion. The motion of a point is an example of local motion. The motion of an object is more global but is limited to a region in an image. Camera motion manifests itself in ST surfaces by all points on all ST surfaces flowing in accordance with the camera motion. In this paper we are concerned with using local methods to recover low-level motion of points on ST surfaces, i.e., the ST surface flow. ST surface flow is a first, low-level step to motion understanding. The flow of an ST surface can be examined spatially and temporally about a point in order to recover information about the contours that generated the ST surface and how the flow changes over time. By combining this information over time and space we can begin to understand the global structure of ST surfaces and the motion they represent.

The examples in Figure 1 show that the structure of ST surfaces is a result of the motion of contours in time. One approach to quantifying this structure would be to use partial derivatives to examine how the surface changes. This is the approach taken in many optic flow algorithms [Agga88]. If $E(x, y, t)$ is the image intensity at point (x, y) in the image at time t , then assuming that the intensity of a point does not change as it moves, we have

$$E(x, y, t) = E(x + \delta x, y + \delta y, t + \delta t) \quad (1)$$

Approximating the right-hand side by a truncated Taylor series about the point (x, y, t) we obtain

$$E(x + \delta x, y + \delta y, t + \delta t) = E(x, y, t) + \delta x \frac{\partial E}{\partial x} + \delta y \frac{\partial E}{\partial y} + \delta t \frac{\partial E}{\partial t} \quad (2)$$

Substituting Eq. (1) in Eq. (2) and dividing through by δt we have

$$\frac{\delta x}{\delta t} \frac{\partial E}{\partial x} + \frac{\delta y}{\delta t} \frac{\partial E}{\partial y} + \frac{\partial E}{\partial t} = 0$$

In the limit as $\delta \rightarrow 0$

$$\frac{dx}{dt} \frac{\partial E}{\partial x} + \frac{dy}{dt} \frac{\partial E}{\partial y} + \frac{\partial E}{\partial t} = 0 \quad (3)$$

This is known as the optic flow constraint equation and, along with additional constraints [Agga88], has been widely used to compute optic flow.

The optic flow constraint equation contains two unknowns, $\frac{dx}{dt}$ and $\frac{dy}{dt}$, and is therefore insufficient to uniquely determine the motion. The system is under constrained since δE is affected by motion along both axes. Motion along the x axis affects $\frac{\partial E}{\partial y}$ and motion along the y axis affects $\frac{\partial E}{\partial x}$. If this were not the case, as assumed by Limb and Murphy [Limb75], the motion could be solved using just the simplified constraint equations

$$\frac{dx}{dt} = \frac{\partial E / \partial t}{\partial E / \partial x} \qquad \frac{dy}{dt} = \frac{\partial E / \partial t}{\partial E / \partial y}$$

Not only do the changes of the ST surface along x and y affect each other, but they do not have the uniformity we desire. For example, with the origin defined as the center of the image, consider an ST surface defined by $f(x, y, t) = (x+t)^2 + y^2$ such that each image has a paraboloid gray level structure and this structure translates along the x axis linearly with t . Since this surface results from a constant motion of a rigid gray level structure, the motion at every point in the image is the same. But the partial derivatives, $f_x(x, y, t) = 2(x+t)$, $f_t(x, y, t) = 2(x+t)$ and $f_y = 2y$, vary depending upon the point on the surface. Even though the surface was generated by a rigid structure undergoing constant translational motion, the *shape* of the surface, at least shape described by its partial derivatives, does not give the information necessary to recover the motion.

Rather than examining the shape of ST surfaces directly, we propose using a measure that does not vary with the parameters that define the surface. We use the gradient direction of a function on the surface rather than, say, the partial derivatives or gradient of the ST surface. This function measures the rate of change of the length of a contour as the contour moves along the surface. The directional derivative of this function gives information about the ST surface. Specifically, the direction of motion for a point is perpendicular to the gradient direction on the surface of this function.

Optic flow is equivalent to a temporal slice of the ST surface flow projected into the spatial plane. ST surfaces extend arbitrarily far back in time, so while any particular time slice gives the optic flow, there is a temporal coherence to the ST surface flow that does not exist when optic flow is independently and repeatedly computed at successive times. A complete description of ST surfaces includes global as well as local information. That is, not only is the flow at each point needed, but higher level descriptions such as long-range periodic motion are also desired [Allm90]. In order to recover more global

descriptions of ST surfaces, temporal as well as spatial coherence is needed. Optic flow uses spatial coherence, but the traditional approach which solves for the flow field at one instant of time, does not maintain temporal coherence. By recovering the ST surface flow of each point on an ST surface we can begin to understand the global structure of the ST surface and the physical behavior of the generating contours.

Related work in differential geometry and ST surfaces is presented in the next section. Section 3 shows that the direction of motion is perpendicular to the gradient of the arc length change function. This is shown using ST 2-surfaces composed from edge points and is then extended to ST 3-surfaces from a sequence of gray level images. Results are presented in Section 4. Finally, the relationship between our approach to computing ST surface flow and the gradient-based optic flow approach is discussed in Section 5. The behavior of our algorithm in cases where the aperture problem exists is also discussed in Section 5.

2. Related Work

Related work falls into two categories: related work with differential geometry and surface descriptors, and related work with ST surfaces. The former is concerned with using ideas from differential geometry to describe different types of surfaces. For example, if a range image is used to define a surface where the value at each pixel is the "height" of the surface, the curvature of the surface can be used for segmenting the image. There has been relatively little work using differential geometry for ST surface interpretation.

There have been numerous suggestions as to which measures are appropriate to describe a surface. Medioni and Navatia [Medi84] suggested using Gaussian curvature and the maximum principle curvature. Brady *et al.* [Brad85] used principle curvature directions and lines of curvature to describe surfaces. Besl and Jain [Besl86] suggested using Gaussian and mean curvature together as surface descriptors. Haralik *et al.* proposed the topographic primal sketch which labels each pixel in a range image as one of ten possible topographic labels. Gradients, Hessians, and first and second directional derivatives were used.

Since we are interested in the structure of ST surfaces and the structure is a result of the motion of contours, a measure that gives direction of motion is desirable. Many of these measures, with the

exception of lines of curvature and directional derivatives, are directionless. Lines of curvature of an ST surface give little information about the contour or the motion of the contour that generated the ST surface. In the restricted case where the motion is translational with constant speed, the lines of curvature associated with the minimum principle curvature correspond to the direction of motion. But this case is too restrictive to be of general interest. It will be shown that directional derivatives can be useful; not the directional derivative of an ST surface, rather the directional derivative of a function on an ST surface. Similar to lines of curvature, the directional derivative of an ST surface gives incomplete information about a contour and its motion except in very restrictive cases.

Goldgof, Lee and Huang [Gold88] used a curvature-based approach to analyze the motion of nonrigid surfaces. But their approach works with the actual 3D structure of the surfaces, i.e., the depth of each point of the surface has to be recovered *a priori*. We are concerned with recovering an interpretation of ST surfaces which can give information about motion in the scene without first recovering depth information.

Baker and Bolles [Boll87, Bake89] examined ST surfaces in order to recover camera motion but objects in the scene were stationary.

Jain [Jain88] described a spatiotemporal volume segmentation approach using the signs of the three principle curvatures of the ST 3-surface. While using curvature values is suitable for segmenting an ST volume, it does not give complete information about the structure, i.e., direction of motion, of the ST surfaces in the volume.

Peng and Medioni [Peng89] presented an algorithm which takes oriented slices of an ST volume. The features in the slices are then examined to recover the normal component of motion. Since only slices of the ST volume are used, there is no coherent treatment of the volume. Liou and Jain [Liou89] defined a motion detector based on the angle between the gradient direction in an ST volume and the time axis. Both these methods suffer from being able to detect only the normal component of motion even when the true motion is recoverable. Peng and Medioni addressed this problem with a second level of processing. For Liou and Jain this was not as great a problem since the goal was motion detection. Recovering only the normal component is sufficient if one is only interested in detecting where there was motion in an image sequence.

3. Local Spatiotemporal Surface Interpretation

In this section we show how to recover the instantaneous direction of motion of a point on the generating contour of an ST surface. We use a parameterization of the surface called the model parameterization which requires the motion of the generating contour. Since this parameterization assumes knowledge of the motion, we also show how the surface can be reparameterized as a Monge patch such that the results for the model parameterization still hold and this reparameterization is computable without knowing the motion.

3.1. Differential Geometry Preliminaries

Before presenting our results, we briefly review necessary definitions from differential geometry. See [DoCa76] , for example, for more details. Subscripts on mappings indicate the partial derivative with respect to the subscript. $\langle \mathbf{a}, \mathbf{b} \rangle_p$ indicates the inner product of vectors \mathbf{a} and \mathbf{b} evaluated at the point p . The inner product is also indicated by $\mathbf{a} \cdot \mathbf{b}$. Throughout this paper curves and surfaces are described parametrically rather than implicitly. For example, a parameterized circle is given by

$$\alpha(l) = (\cos(l), \sin(l)) \quad \alpha : (0, 2\pi) \rightarrow \mathbb{R}^2$$

Formally, a parameterized curve in \mathbb{R}^3 is a map $\alpha: I \rightarrow \mathbb{R}^3$ of an open interval $I = (a, b)$ of the real line \mathbb{R} into \mathbb{R}^3 .

Similarly, parameterized 2-surfaces and 3-surfaces are defined as, respectively, (see Figure 2):

$$\mathbf{X}: U \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3 \quad \mathbf{X}: U \subset \mathbb{R}^3 \rightarrow \mathbb{R}^4$$

For example,

$$\mathbf{X}(u, v) = (u, \sqrt{1 - (u^2 + v^2)}, v)$$

parameterizes the top half of a sphere.

The tangent plane at a point p on a surface S is denoted by $T_p(S)$. For a 3-surface this plane is a hyperplane.

For any parameterized surface, the first fundamental form measures the movement of a point on the surface given a movement of the point in parameter space. For example, let (u_0, v_0) be a point in parameter space and $\mathbf{X}(u_0, v_0)$ be the point mapped to the surface. As u_0 and v_0 change, the first fundamental

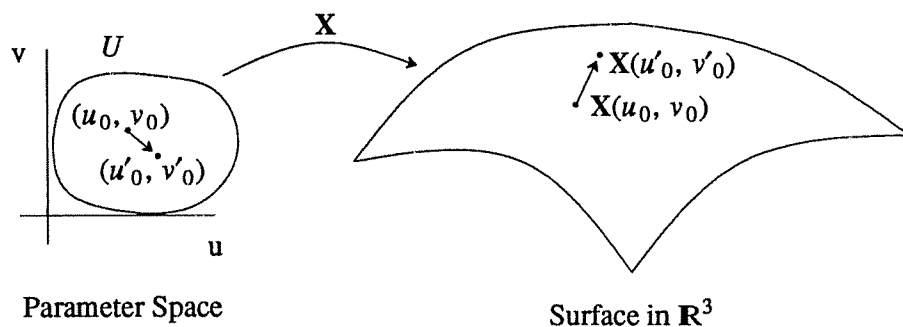


Figure 2. The definition of a surface. $\mathbf{X}(u', v')$ shows the change of the point $\mathbf{X}(p)$ on the surface for the change (u', v') of p in parameter space.

form measures how $\mathbf{X}(u_0, v_0)$ changes (see Figure 2). We express the first fundamental form in the basis $\{\mathbf{X}_u, \mathbf{X}_v\}$ associated with parameterization $\mathbf{X}(u, v)$ at p . For movement of (u', v') in parameter space, the first fundamental form at point p is

$$I_p = E (u')^2 + 2F u' v' + G (v')^2 \tag{4}$$

where

$$E = \langle \mathbf{X}_u, \mathbf{X}_u \rangle_p \quad F = \langle \mathbf{X}_u, \mathbf{X}_v \rangle_p \quad G = \langle \mathbf{X}_v, \mathbf{X}_v \rangle_p \tag{5}$$

For example, let $\mathbf{X}(u, v) = (\cos(u), \sin(u), v)$. Then

$$\mathbf{X}_u = (-\sin(u), \cos(u), 0) \quad \mathbf{X}_v = (0, 0, 1)$$

and

$$E = 1 \quad F = 0 \quad G = 1 \tag{6}$$

Substituting Eq. (6) in Eq. (5) gives

$$I_p = (u')^2 + (v')^2$$

Let $p = (0, 1)$ and the change in parameter space be $(u', v') = (0, 1)$. Then $I_p = 1$. So movement in parameter space results in movement of equal length on the surface. See Figure 3.

The first fundamental form is important because it can be used to answer metric questions about a surface. For example, the first fundamental form can be used to find the arc length of a curve on a surface.

The arc length s of a parameterized curve $\alpha: I \rightarrow S$ is given by

$$s(l) = \int_0^l |\alpha'(l)| dl = \int_0^l \sqrt{I(\alpha'(l))} dl$$

In particular, if $\alpha(l) = \mathbf{X}(u(l), v(l))$ is contained in a coordinate neighborhood corresponding to the parameterization $\mathbf{X}(u, v)$, we can compute the arc length of α between, say, 0 and l by

$$s(l) = \int_0^l \sqrt{E \left(\frac{du}{dl}\right)^2 + 2F \frac{du}{dl} \frac{dv}{dl} + G \left(\frac{dv}{dl}\right)^2} dl \quad (7)$$

Continuing the example, let $(u(l), v(l)) = (l, 1)$ parameterize a curve in the parameter space of the surface as shown in Figure 3. Consequently, $\mathbf{X}(u(l), v(l)) = \mathbf{X}(l, 1)$ is a curve that goes around the cylinder.

Then $\frac{du}{dl} = 1$ and $\frac{dv}{dl} = 0$. Substituting into Eq. (7) gives

$$\int_0^l dl = l$$

It is common for Eq. (7) to be rewritten as

$$\left[\frac{ds}{dl}\right]^2 = E \left[\frac{du}{dl}\right]^2 + 2F \left[\frac{du}{dl} \frac{dv}{dl}\right] + G \left[\frac{dv}{dl}\right]^2$$

giving the rate of change of arc length of α . In the example we have $\left[\frac{ds}{dl}\right]^2 = 1$, indicating that the

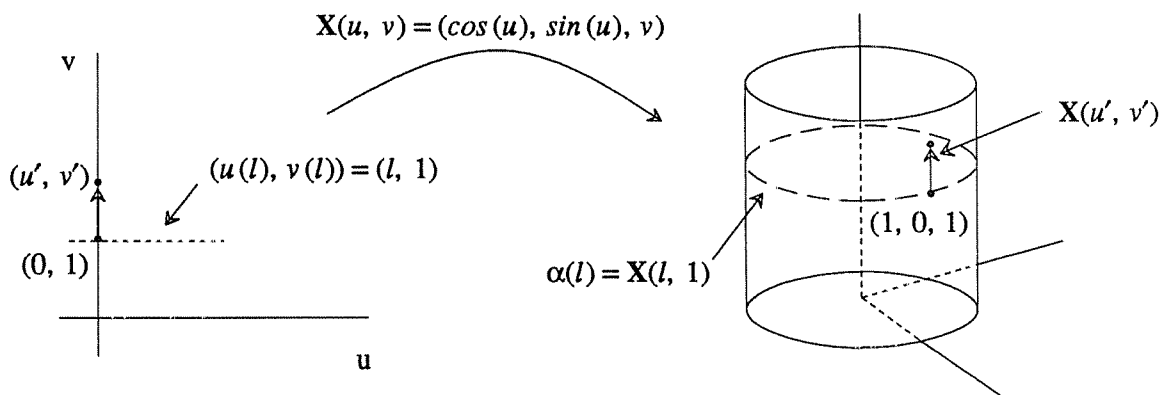


Figure 3. A change of (u', v') in parameter space results in a change of $\mathbf{X}(u', v')$ on the surface. A curve in parameter space mapped to a surface.

parameterizing variable l and the arc length vary at the same rate.

3.2. The Model Parameterization of Spatiotemporal Surfaces

ST surfaces are formed as a result of the movement of objects in the scene. The movement of objects results in movement of the contours that make up the projection of the object in the image plane. The resulting ST surfaces can be parameterized based on the type of motion the object and its contours undergo in the scene.

Let S be an ST surface recovered from an image sequence. (How this is done will be addressed later.) If a temporal slice is taken from this ST surface, the generating contour will be a curve in this slice. Let $\alpha(s) = (s, h(s))$ parameterize the generating contour. Consider the case where the contour is translating parallel to the image plane with speed $V^s(t)$ and $V^h(t)$ in the s and h directions, respectively. The resulting ST surfaces can be parameterized using s and time, t , as the two parameters. s parameterizes the contour while t parameterizes the motion. An ST surface generated under orthographic projection of a translating contour is given by

$$\mathbf{X}(s, t) = (s + V^s(t), h(s) + V^h(t), t) \quad (8)$$

Similarly, an ST surface generated as a result of a contour rotating parallel to the image plane is given by

$$\mathbf{X}(s, t) = (s \cos(\theta(t)) + h(s) \sin(\theta(t)), -s \sin(\theta(t)) + h(s) \cos(\theta(t)), t) \quad (9)$$

Again, s parameterizes the contour while t parameterizes the motion. This parameterization results from rotating the contour as a function of time. $\theta(t)$ determines the rate of rotation. Rotation is assumed to be around the origin since, without loss of generality, we can always reparameterize the contour so that the origin is defined as the center of rotation. $\theta(t)$, $V^s(t)$ and $V^h(t)$ can be arbitrarily complex in t . For example, they can contain quadratic terms to allow for acceleration.

Combining Eqs. (8) and (9) we arrive at the following parameterized ST surface for an arbitrary rigid contour undergoing translation and rotation in the image plane:

$$\mathbf{X}(s, t) = (s \cos(\theta(t)) + h(s) \sin(\theta(t)) + V^s(t), -s \sin(\theta(t)) + h(s) \cos(\theta(t)) + V^h(t), t) \quad (10)$$

Given these restricted types of motion, any recovered ST surface can be described by these model parameterizations. However, since these parameterizations require knowing the motion of the contour, it

is not possible to actually use them unless the motion is known *a priori*, which is generally not the case. Despite this, we can show properties of these ST surfaces that are invariant to the parameterization of the surface. For example, it can be shown with the parameterizations above that an ST surface generated from a contour undergoing constant translational motion has Gaussian curvature equal to 0 at every point on the ST surface. Thus, if an ST surface that was generated from a contour moving with constant translational motion is recovered from an image sequence, then the Gaussian curvature must be 0 on this ST surface. This is true no matter how the ST surface is parameterized. In the next section it will be shown how to recover the direction of motion of a point on a contour using the model parameterization above. It will then be shown how to reparameterize the surface such that it can be computed without knowing the motion of the contour.

3.3. Recovering Point Motion on Spatiotemporal Surfaces

The generating contour at time t is obtained by taking a temporal slice of the ST surface at time t . For any point on the contour we would like to recover the instantaneous motion of the point. First, consider a small segment of the contour which includes the point. Figure 4 shows the contour and segment at two times, t_0 and t_1 . The long dashed lines show the true movement of the contour segment. Obviously, the arc length of the segment in temporal slice t_0 is the same as in temporal slice t_1 . But suppose it is

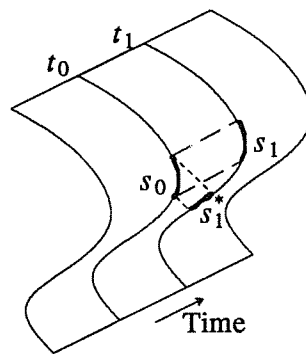


Figure 4. Arc length changes if the segment s_0 is moved in the wrong direction. s_1 preserves arc length, but s_1^* does not.

hypothesized that the motion of the contour is as shown by the short dashed lines. Since the contour is rigid, the end points of the segment undergo the same movement. The result of moving the end points in the same direction to the next temporal slice requires a change of arc length of the segment. *Thus, given any small segment of a contour in a temporal slice, the correct movement of the end points must result in no change of arc length in the next temporal slice.* It is this observation that motivates our method. However, the small segment will be replaced by a point and this intuitive observation will be made formal.

Using the model parameterization of ST surfaces, the generating contour is parameterized on the ST surface by keeping t constant in the parameter space of the surface. For example, for an ST surface generated by translational motion, the following curve on the surface is the same as the generating contour:

$$\alpha(l) = \mathbf{X}(u(l), v(l)) = \mathbf{X}(l, t_0) = \mathbf{X}(l + V^s(t_0), h(l) + V^h(t_0), t_0)$$

In general

$$u(l) = l \qquad v(l) = t_0$$

Earlier we defined the rate of change of arc length of a curve on a surface to be

$$\left[\frac{ds}{dl} \right]^2 = E \left[\frac{du}{dl} \right]^2 + 2F \left[\frac{du}{dl} \frac{dv}{dl} \right] + G \left[\frac{dv}{dl} \right]^2 \quad (11)$$

For the generating curve contour we know

$$\frac{du}{dl} = 1 \qquad \frac{dv}{dl} = 0 \quad (12)$$

Substituting Eq. (12) in Eq. (11), the square of the instantaneous change of arc length is given by

$$\left[\frac{ds}{dl} \right]^2 = E$$

E is defined at every point on an ST surface. For any given point, if one moves along the surface in the direction the contour is moving, then the change in E is 0. This fact will be proved below. Consequently, given a point on an ST surface, the direction of motion is the direction such that the change in E is 0. Equivalently, the direction of motion is the direction such that the directional derivative of E is 0. This direction will be shown to be perpendicular to the gradient direction in the tangent plane.

At this point we have an ST surface S and a function E defined at every point on the surface. The gradient direction of a differentiable function $f: S \rightarrow \mathbb{R}$ is a differential map $\nabla f: S \rightarrow \mathbb{R}^3$ which assigns to each point $p \in S$ a vector $\nabla f(p) \in T_p(S) \subset \mathbb{R}^3$. ∇f is given by [DoCa76]:

$$\nabla f = \frac{f_s G - f_t F}{EG - F^2} \mathbf{X}_s - \frac{f_t E - f_s F}{EG - F^2} \mathbf{X}_t \quad (13)$$

Since we are only concerned with the direction and not the magnitude of the gradient, the denominators can be eliminated in Eq. (13).

Let $f = E$. Recall that $E = \mathbf{X}_s \cdot \mathbf{X}_s$. Therefore, from Eq. (10), for any combination of translational and rotational motion we have

$$\begin{aligned} \mathbf{X}_s(x, t) &= (\cos(\theta(t)) + h_s(s, t) \sin(\theta(t)), -\sin(\theta(t)) + h_s(s, t) \cos(\theta(t)), 0) \\ E &= \mathbf{X}_s \cdot \mathbf{X}_s = 1 + h_s^2(s) \end{aligned}$$

The partial derivatives of E are then given as

$$f_s = E_s = 2 h_s(s) h_{ss}(s) \quad f_t = E_t = 2 h_s(s) h_{st}(s) = 0 \quad (14)$$

Substituting $E_t = 0$ in Eq. (13), the gradient of E on an ST surface is defined as

$$\nabla E = (E_s G) \mathbf{X}_s - (E_s F) \mathbf{X}_t$$

Note that \mathbf{X}_t is in the direction of motion. So if ∇E is perpendicular to the direction of motion, the inner product of ∇E and \mathbf{X}_t should be 0. This is in fact the case:

$$\begin{aligned} \nabla E \cdot \mathbf{X}_t &= \left[(E_s G) \mathbf{X}_s - (E_s F) \mathbf{X}_t \right] \cdot \mathbf{X}_t \\ &= \left[E_s (\mathbf{X}_t \cdot \mathbf{X}_t) \mathbf{X}_s - E_s (\mathbf{X}_s \cdot \mathbf{X}_t) \mathbf{X}_t - \right] \cdot \mathbf{X}_t \\ &= E_s (\mathbf{X}_t \cdot \mathbf{X}_t) (\mathbf{X}_t \cdot \mathbf{X}_s) - E_s (\mathbf{X}_s \cdot \mathbf{X}_t) (\mathbf{X}_t \cdot \mathbf{X}_t) \\ &= 0 \end{aligned}$$

To summarize, we have shown using the model parameterization of an ST surface, ∇E is perpendicular to the direction of motion in the tangent plane. Unfortunately, this parameterization cannot be used directly because it assumes prior knowledge of the motion of the contour. Consequently, we must reparameterize the surface such that E is the same but knowledge of the motion is not required.

3.4. The Monge Patch Parameterization of Spatiotemporal Surfaces

While the model parameterization used in the previous section is useful for showing properties of ST surfaces, it is not useful for parameterizing ST surfaces when the motion is not known. In this section ST surfaces recovered from an image sequence will be represented as a set of Monge patches. For every point on an ST surface, a patch will be recovered and parameterized.

The result of the previous section says that the direction of motion is perpendicular to the gradient of E in the tangent plane. This remains true when the surface is reparameterized so long as E in the reparameterization equals E in the model parameterization. The Monge patches recovered at each point on an ST surface will be parameterized such that E in a parameterized Monge patch equals E in the model parameterizations. Therefore the result of the previous section will apply.

Any parameterized surface can be represented locally by a function [DoCa76]. The map

$$\mathbf{X}(s, t) = (s, t, f(s, t))$$

is called a Monge patch and f is the "height" as a function of s and t . See Figure 5. See [Besl86] for a complete discussion of Monge patches.

For each point in an ST volume, a quadratic patch is fit using that point and the temporal and spatial neighboring edge points. This quadratic patch defines f . The partial derivatives of the patch are equal to:

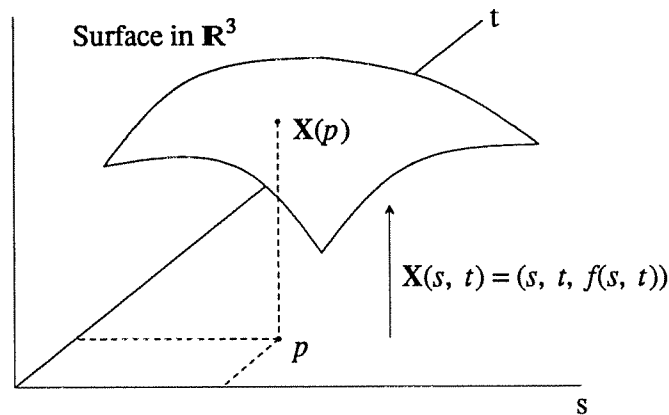


Figure 5. A surface defined locally as a height function.

$$\mathbf{X}_s = (1, 0, f_s(s, t)) \quad \mathbf{X}_t = (0, 1, f_t(s, t)) \quad (15)$$

Therefore

$$E = \mathbf{X}_s \cdot \mathbf{X}_s = 1 + f_s^2(s, t) \quad (16)$$

$$E_s = 2f_s(s, t)f_{ss}(s, t) \quad E_t = 2f_s(s, t)f_{st}(s, t) \quad (17)$$

$f_s(s, t)$, $f_t(s, t)$, $f_{ss}(s, t)$ and $f_{st}(s, t)$ are easily computable [Besl86].

We will now show that E in a Monge patch parameterized surface is equal to E in the model parameterization of that surface. Since the surface is the same and the function E on the surfaces are equal, the perpendicular direction to the gradient of E will be equal in the two parameterizations.

Recall that $E = \mathbf{X}_s \cdot \mathbf{X}_s$. Therefore $E = 1 + h_s^2(s)$ in the model parameterization and $E = 1 + f_s^2(s, t)$ in the Monge patch parameterization. To show that the E 's in the two parameterizations are equal we need to show that $h_s(s) = f_s(s, t)$. In the Monge patch parameterization, the s and t directions can be chosen arbitrarily as long as they are not parallel. Therefore we can choose them to correspond with s and t in the model parameterization. t is perpendicular to the image plane and s is perpendicular to t . $h_s(s)$ in the model parameterization is the slope of the generating contour. Let p be a point on the generating contour at time t_0 and $h_s(s)$ the slope of the generating contour at t_0 . Consider a Monge patch centered at p . The partial derivative of f with respect to s , $f_s(s, t)$, is the same as the derivative of the curve of intersection formed by the surface and a plane perpendicular to the t axis. This curve of intersection is simply the generating contour. Therefore $f_s(s, t)$ must equal $h_s(s)$ and E is the same in both parameterizations. Figure 6 shows the relative positions of the basis vectors in each parameterization.

Since the functions E defined on the ST surface are equal, and reparameterization of a surface does not affect the gradient of a function on the surface, we have shown that ∇E in the model parameterization must equal ∇E in the Monge patch parameterization. Earlier we showed that ∇E in the model parameterization is perpendicular, in the tangent plane, to the direction of motion, so ∇E in the Monge patch parameterization must also be perpendicular to the direction of motion.

3.4.1. An Algorithm

An exact procedure for recovering the motion of a point on a contour using the Monge patch parameterization can now be given:

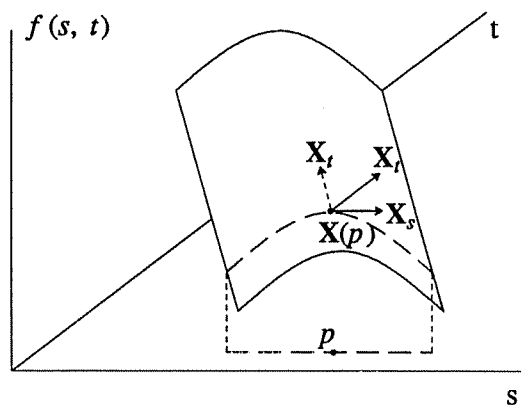


Figure 6. The basis vectors in the model and Monge patch parameterizations. The short dashed arrow indicates the model parameterization and the solid arrows indicate the Monge patch parameterization. X_s is the same in both. The long dashed line shows the generating contour which is translating left.

- Detect points on ST surfaces in a given ST volume using a spatiotemporal edge operator, e.g., three-dimensional zero-crossings of the Laplacian
- Fit a quadratic surface to the edge points [Besl86]
- Compute X_s , X_t and E , F and G of the patch (Eqs. (5), (15) and (16)) [Besl86]
- Compute E_s and E_t (Eq. (17))
- Compute ∇E on the patch (Eq. (13))
- Compute the direction in the tangent plane of the point that is perpendicular to the gradient direction

In order to find the direction perpendicular to the gradient, the following procedure is used. The cross product of the two basis vectors, X_s and X_t , gives a vector normal to the tangent plane. The cross product of this normal with the vector in the gradient direction gives a vector in the tangent plane which is perpendicular to the gradient direction.

3.5. Gray Level Images and Spatiotemporal 3-Surfaces

All the ideas presented for ST surfaces defined by the edge points in an ST volume can be extended to work with gray level images. Using edge maps resulted in a 2-surface embedded in 3-space. Using gray level images results in a 3-surface embedded in 4-space. The basic ideas and procedures are the same but

the dimension is increased. This results in slightly different definitions which are presented below. The "contour" is now two dimensional, parameterized by s and g . The gray level at each pixel defines the height of the contour. The model parameterization, again used to derive properties of the surface, for a contour undergoing translation and a contour undergoing rotation are, respectively,

$$\mathbf{X}(s, g, t) = (s + V^s(t), g + V^g(t), h(s, g), t) \quad (18)$$

$$\mathbf{X}(s, g, t) = (s \cos(\theta(t)) + g \sin(\theta(t)), -s \sin(\theta(t)) + g \cos(\theta(t)), h(s, g), t) \quad (19)$$

Combining Eqs. (18) and (19) gives the parameterization for a contour undergoing translation and rotation. The first fundamental form coefficients are

$$\begin{aligned} A &= \mathbf{X}_s \cdot \mathbf{X}_s & B &= \mathbf{X}_g \cdot \mathbf{X}_g & C &= \mathbf{X}_t \cdot \mathbf{X}_t \\ D &= \mathbf{X}_s \cdot \mathbf{X}_t & E &= \mathbf{X}_t \cdot \mathbf{X}_g & F &= \mathbf{X}_s \cdot \mathbf{X}_g \end{aligned}$$

The gradient direction of a function, f , on a 3-surface is given by:

$$\nabla f = \begin{bmatrix} B C - E^2 & E F - B D & D E - C F \\ E F - B D & A B - F^2 & D F - A E \\ D E - C F & D F - A E & A C - D^2 \end{bmatrix} \begin{bmatrix} f_s \\ f_g \\ f_t \end{bmatrix} \quad (20)$$

in the basis $\{\mathbf{X}_s, \mathbf{X}_g, \mathbf{X}_t\}$.

Since the contour is now parameterized by two variables, there are two functions that can be defined on the contour. One gradient direction is not sufficient to uniquely determine the direction of motion since there are an infinite number of vectors in 3-space perpendicular to the gradient direction. Therefore, the gradient directions of two functions are used. The cross product of the two gradient directions is perpendicular and in the direction of motion. These two functions are the analogs of E in the first fundamental form for a 2-surface and are defined as

$$f_1 = A \quad f_2 = B$$

where A and B are the coefficients from the first fundamental form for 3-surfaces. The gradient of each function is found separately. Using the model parameterization for translation and rotation of gray level images (the combination of Eqs. (18) and (19)) we find that A and B and their partial derivatives are equal to the following:

$$\begin{aligned}
A &= 1 + h_s^2(s, g) & B &= 1 + h_g^2(s, g) \\
A_s &= 2 h_s(s, g) h_{ss}(s, g) & B_s &= 2 h_g(s, g) h_{gs}(s, g) \\
A_g &= 2 h_s(s, g) h_{sg}(s, g) & B_g &= 2 h_g(s, g) h_{gg}(s, g) \\
A_t &= 2 h_s(s, g) h_{st}(s, g) = 0 & B_t &= 2 h_g(s, g) h_{gt}(s, g) = 0
\end{aligned}$$

Substituting A_s , A_g and A_t into Eq. (20) we get a vector in the basis $\{\mathbf{X}_s, \mathbf{X}_g, \mathbf{X}_t\}$ which is perpendicular to the direction of motion. This is proved by showing that the inner product of the resulting vector with \mathbf{X}_t is 0. A second vector perpendicular to the direction of motion is obtained by replacing A with B . The cross product of these two vectors then gives the direction of motion.

The procedure to compute the gradient directions and the direction of motion is very similar to the procedure when the ST surface is a 2-surface. A Monge patch and its partial derivatives are now defined as

$$\begin{aligned}
\mathbf{X}(s, g, t) &= (s, g, t, f(s, g, t)) \\
\mathbf{X}_s &= (1, 0, 0, f_s(s, g, t)) & \mathbf{X}_g &= (0, 1, 0, f_g(s, g, t)) & \mathbf{X}_t &= (0, 0, 1, f_t(s, g, t))
\end{aligned}$$

The quadratic surface fit described by Besl and Jain [Besl86] can easily be extended to describe 3-surfaces and used to compute the Monge patch partial derivatives. It is no longer necessary to detect ST surfaces using an edge operator. All the temporal and spatial neighbors of a point in the ST volume are used to fit the quadratic surface.

The gradient direction of the functions A and B each define a vector in 4-space. The direction of motion on the ST 3-surface is perpendicular to this vector. Note that we are only interested in the projection of this perpendicular direction into 3-space. But the perpendicular vector in 4-space must be found before the projection into 3-space. To find the perpendicular direction we use the same procedure as used for ST 2-surfaces. The cross product of the three basis vectors \mathbf{X}_s , \mathbf{X}_g and \mathbf{X}_t is computed. The resulting vector is perpendicular to the tangent hyperplane at the point. The cross product of this vector and the two gradient direction vectors gives a vector in the tangent hyperplane which is perpendicular to both gradient vectors. This final vector can then be projected into 3-space to give the direction of motion.

4. Results

Figures 7 to 12 show the results of our method on five image sequences. The sequences include motion of edge images in Figures 7 and 8, and gray level images in Figures 9 to 12. The motion in Figures 8 and 9 was produced by physically moving the objects in front of the camera. The motion in Figures 7, 10 and 11 was generated synthetically by transforming each image in a static sequence of frames. All gray level images in a sequence were smoothed using a $7 \times 7 \times 7$ Gaussian-weighted kernel. The kernel was convolved twice with each ST volume.

The results show a single temporal slice from the middle of each ST surface flow field. *Time* is into the page so the vectors shown are the result of projecting the ST surface flow vectors into the spatial plane. A vector showing no motion, i.e., no spatial component, is oriented straight into time and appears as a dot. The greater the speed of a point, the greater the spatial component of the vector and hence, the longer the vector appears in the spatial projection.

We found that the edge images worked better when there was a relatively large amount of movement between frames, i.e., greater than 1 pixel/frame, whereas the gray level images worked better with small movements, i.e., less than 1 pixel/frame.

With the gray level images there was a graceful degradation in results as the number of frames in the sequence was reduced. This was also the case with the edge images but a significantly longer sequence was required to achieve good results.

Flow vectors were not found for the edge images at the top and back of the object in Figure 7 and the top of the object in Figure 8 because an insufficient number of edge points were found in these regions. For each edge point, the edge points within a neighborhood around that point were found and used for fitting a Monge patch. When edge points were missing in some frames, or the neighborhood size was so large that edge points could not be found for each position in the neighborhood, a Monge patch was not fit to the data.

There are regions of incorrect results in the lower-left part of the left object and the top of the right object in Figure 8. These errors were caused because edge points from separate ST surfaces were recovered as one surface. In general, the problem of segmenting edge points into coherent ST surfaces is a difficult problem [Bake89a] which we have not focused on in this research. Gray level images are prefer-

able in this respect since the intermediate steps of detecting edge points and segmenting them into ST surfaces are avoided.

5. Related Issues

Since optic flow is equivalent to a temporal slice of the ST surface flow projected into the spatial plane, a comparison of the assumptions and robustness of both approaches is of interest. Specifically, we compare our method with gradient-based optic flow algorithms since they are the most similar to ours. Next, we show why our method approximates the correct ST surface flow when an ST surface is generated by object motion that is not parallel to the image plane or when orthographic projection is not assumed. The aperture problem and how our method deals with cases where it exists, is also briefly addressed.

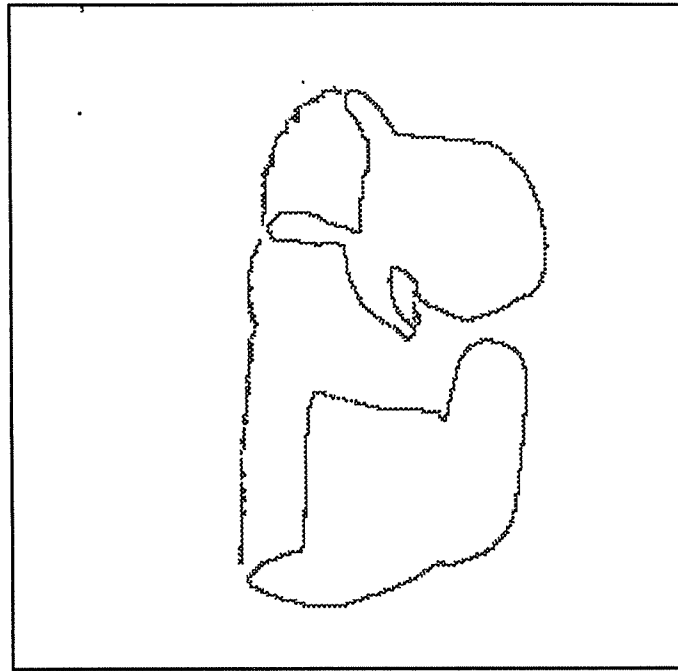
5.1. Comparison with Gradient-Based Optic Flow Algorithms

Gradient-based optic flow algorithms typically use Eq. (3), the optic flow constraint equation, along with additional assumptions in order to solve for the motion $\frac{dx}{dt}$ and $\frac{dy}{dt}$. These additional assumptions include [Agga88]:

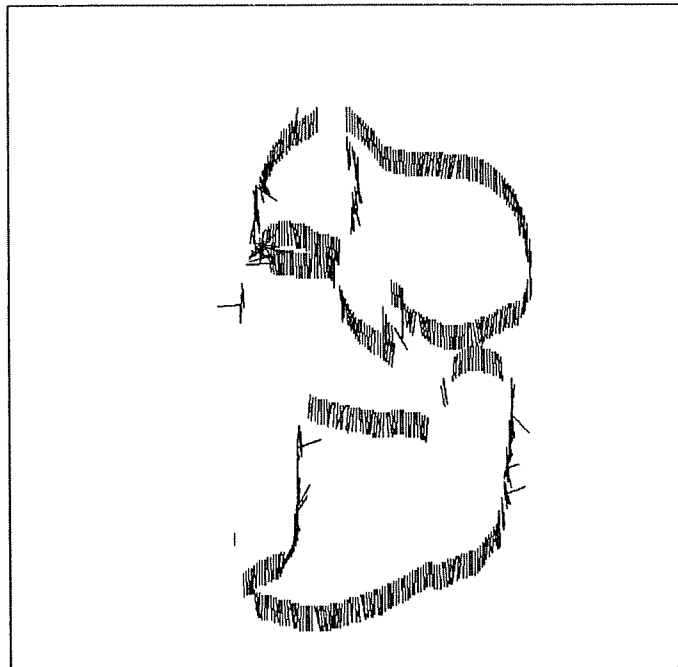
- optic flow is smooth and neighboring points have similar velocities
- optic flow is constant over some segment of the image
- objects in the scene undergo restricted motion

By examining what, if any, of these three assumption our method makes we can better understand the relationship between our method and gradient-based optic flow approaches.

With respect to the first assumption, our approach does not need to assume neighboring points have similar velocities because it solves for the velocity exactly. We implicitly assume that an ST surface is sufficiently smooth so that a quadratic surface can be fit and the partial derivatives recovered. If the ST surface is sufficiently busy over a small neighborhood, poor partial derivatives may result. But this assumption is also made by gradient-based optic flow approaches since the partial derivatives of the ST volume must be computed there as well. This assumption is not a severe limitation since higher order surfaces can always be used [Besl88].

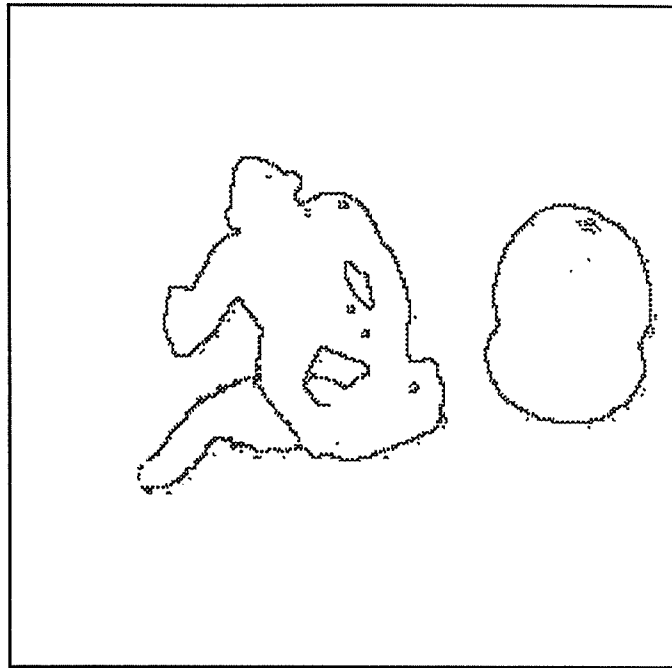


(a)

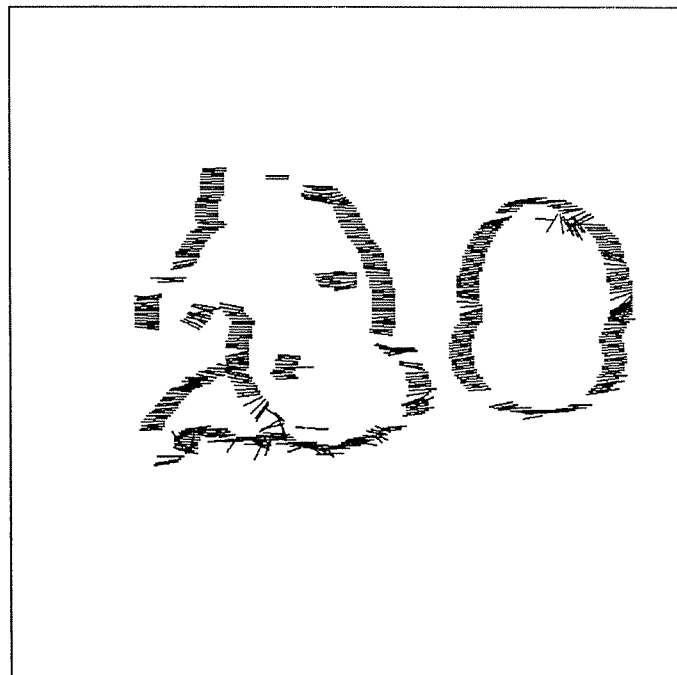


(b)

Figure 7. (a) Middle frame from a 9 frame sequence of a doll translating down at 2 pixels/frame. (b) ST surface flow for the middle frame. The final flow was smoothed using a 3x3 kernel. The motion was synthetically produced by translating each static image.

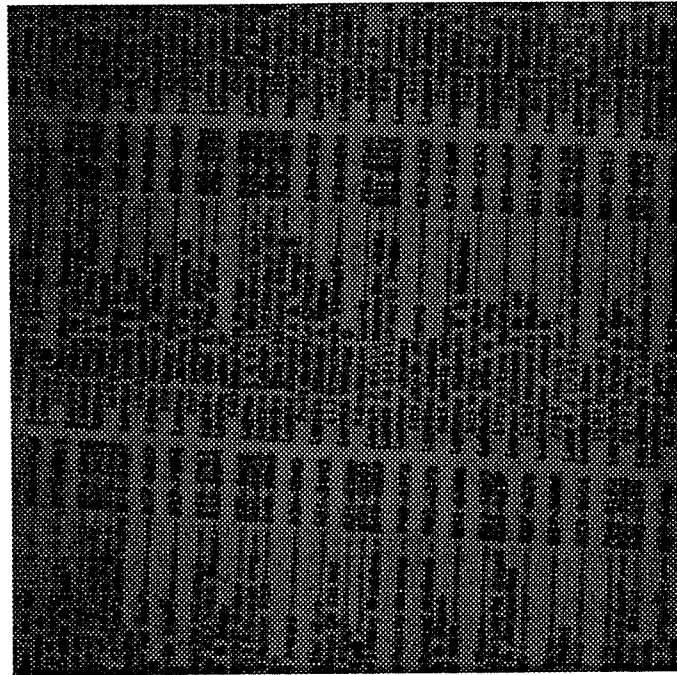


(a)

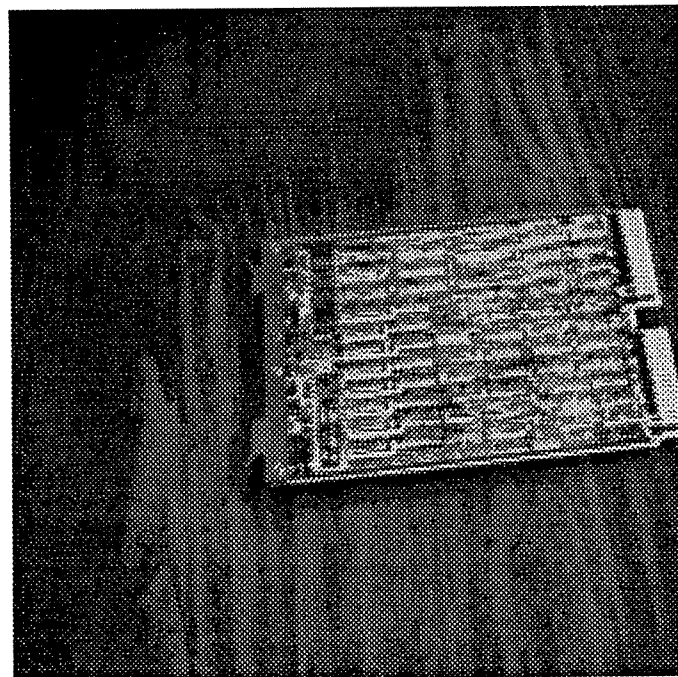


(b)

Figure 8. (a) Middle frame from a 13 frame sequence of a doll and an orange translating left at approximately 1.3 pixels/frame. (b) ST surface flow for the middle frame. The final flow was smoothed using a 3x3 kernel. The motion was produced by physically moving the objects in front of the camera.

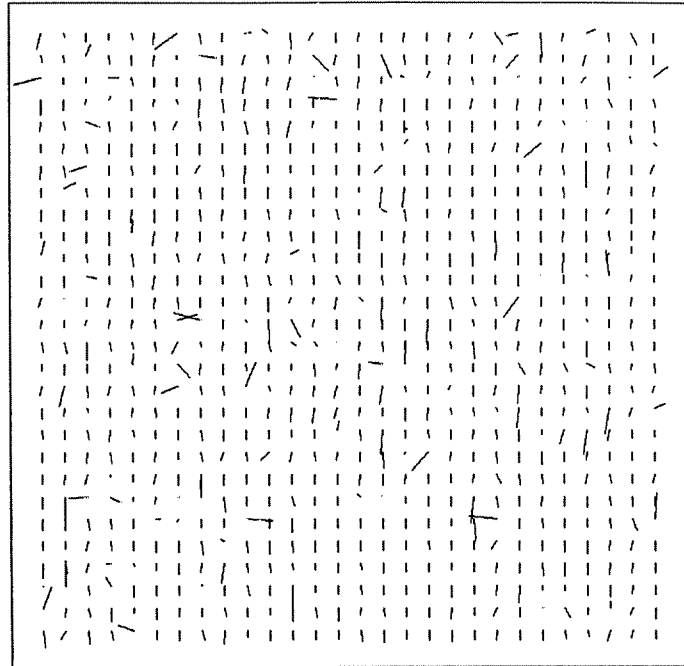


(a)

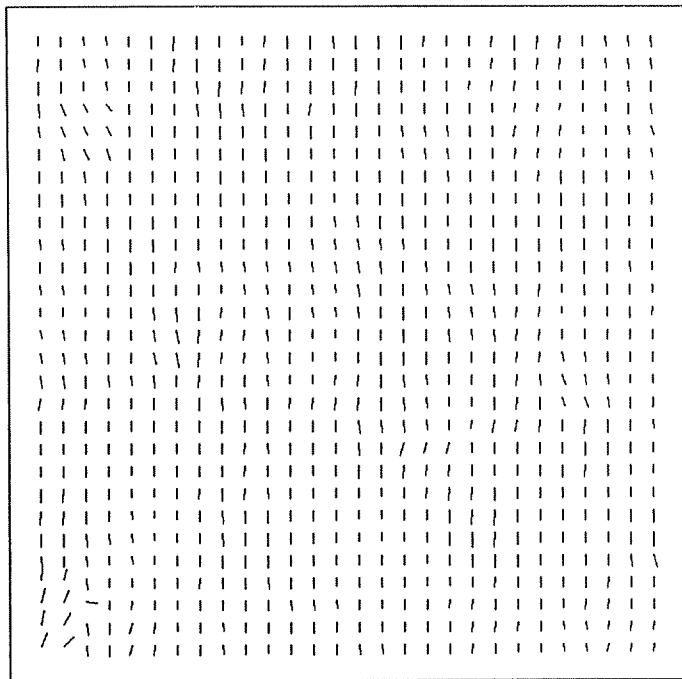


(b)

Figure 9. Two gray level images used in subsequent figures. (a) A page from a phone book; one frame from a 7 frame sequence in which no motion occurred. (b) A circuit board from a 7 frame sequence of images. The board was translated left by physically moving the object in front of the camera at approximately 1.3 pixels/frame.

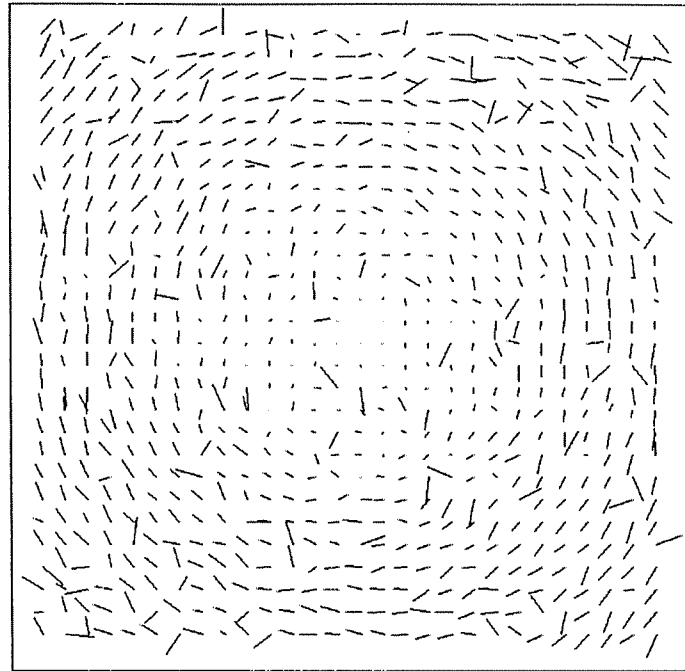


(a)

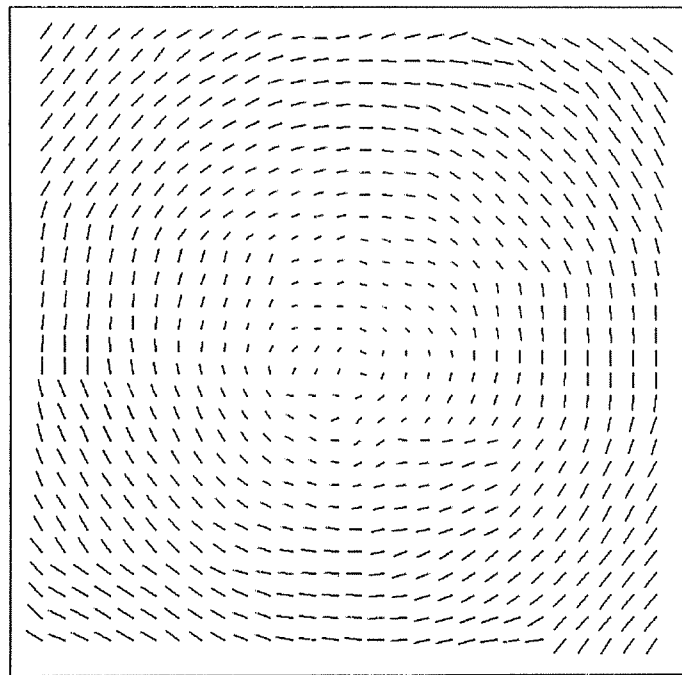


(b)

Figure 10. (a) ST surface flow for the middle frame of the phone book sequence, where motion was synthetically produced by translating each frame down 0.5 pixels/frame. (b) The flow in (a) smoothed using a 3x3 kernel.

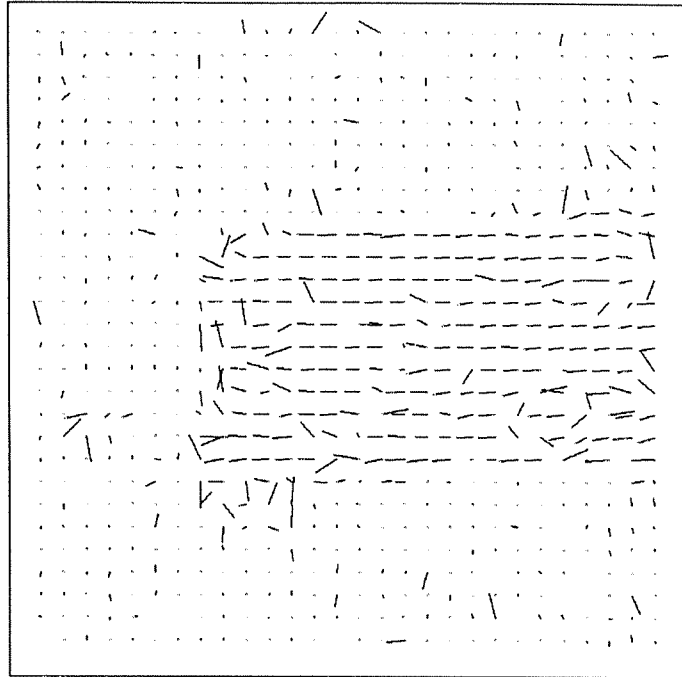


(a)

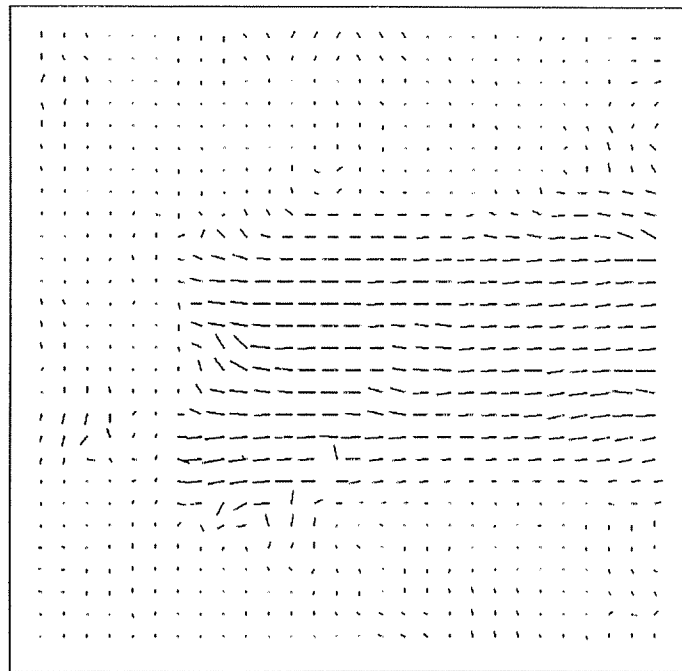


(b)

Figure 11. (a) ST surface flow for the middle frame of the phone book sequence, where motion was synthetically produced by rotating about the center at 0.008 radians/frame. (b) The flow in (a) smoothed using a 5x5 kernel.



(a)



(b)

Figure 12. (a) ST surface flow for the middle frame of the circuit board sequence. (b) The flow in (a) smoothed using a 3x3 kernel.

Since it is not necessary to assume neighboring points have similar velocities, we also do not assume that the optic flow is constant over an image region. Thus, our approach does not make the second assumption given above.

Our model parameterization assumes motion of a contour is a combination of translational and rotational motion parallel to the image plane, i.e., the motion is planar. However, it was shown by Schunck [Schu84] that the optic flow constraint equation is valid only for translation, i.e., it is even more restrictive than our model parameterization.

5.2. Approximating General Motion

The model parameterization assumes that motion in the scene is parallel to the image plane, i.e., there is no rotation in depth. Orthographic projection is also assumed. Therefore, no change in depth occurs. However, if these assumptions are violated, our method does give an approximation to the correct ST surface flow. In this section we show why this is true.

Recall the model parameterization for a contour translating parallel to the image plane is given by

$$\mathbf{X}(s, t) = (s + V^s(t), h(s) + V^h(t), t) \quad (8)$$

where $s + V^s(t)$ parameterizes the s component of the surface and $h(s) + V^h(t)$ parameterizes the h component of the surface. If an object rotates in depth or orthographic projection is not used, the resulting ST surface will not be of the form in Eq. (8). But if it can be shown that the parameterization in Eq. (8) approximates the true ST surface then our method will give an approximation to the true flow. The accuracy of the approximation will depend upon how good the parameterization in Eq. (8) approximates the true ST surface.

The ST surface resulting from a contour rotating in depth such that the projection rotates about the h axis is given by

$$\mathbf{X}(s, t) = (s \cos(\theta(t)), h(s), t) \quad (21)$$

A similar argument holds for rotation about the s axis. $\theta(t)$ measures the rotation of the contour as a function of time. To show that Eq. (21) is approximated by a parameterization of the form in Eq. (8) we need to show that $s \cos(\theta(t))$ can be approximated by $s + V^s(t)$. $\cos(\theta(t))$ is equal to $\sqrt{1 - \sin^2(\theta(t))}$. For small $\theta(t)$ this is approximately equal to $\sqrt{1 - \theta^2(t)}$. Further, $\sqrt{1 - \theta^2(t)}$ is approximately equal to

$1 - \theta^2(t)$ for small $\theta(t)$. Substituting this in Eq. (21) gives

$$\mathbf{X}(s, t) = (s(1 - \theta^2(t)), h(s), t) = (s - s\theta^2(t), h(s), t)$$

Viewing $s\theta^2(t)$ as the $V^s(t)$ term, we see that the model parameterization for a translating contour approximates the parameterization for a contour rotating in depth.

This result can be extended by showing that an ST surface generated by a contour moving in depth under perspective projection is also approximated by the model parameterization of a translating contour. The position in an image of a point (x, y, z) in the scene under perspective projection is $(f \frac{x}{z}, f \frac{y}{z})$ where f is the effective focal length. Applying this to s and $h(s)$ in the model parameterization gives the following parameterization of an ST surface resulting from a contour changing depth in the scene:

$$\mathbf{X}(s, t) = \left(\frac{fs}{z(s, t)}, \frac{fh(s)}{z(s, t)}, t \right) \quad (22)$$

$z(s, t)$ is the depth to a point on the contour which changes with time. Consider a point $p = \mathbf{X}(s_0, t_0)$ on an ST surface. Given a patch of the ST surface around p we will show that this patch can be approximately parameterized by the model parameterization for a translating contour. Approximating the first component of Eq. (22), $F(s, t) = \frac{fs}{z(s, t)}$, with a truncated Taylor series for two variables around the base point (a_1, a_2) we find

$$\begin{aligned} \frac{fs}{z(s, t)} &\approx F(a_1, a_2) + \left[(s - a_1) \frac{\partial F(a_1, a_2)}{\partial s} + (t - a_2) \frac{\partial F(a_1, a_2)}{\partial t} \right] \\ &\approx \frac{fa_1}{z(a_1, a_2)} + \left[(s - a_1) \frac{\partial F(a_1, a_2)}{\partial s} - (t - a_2) \frac{fs}{z^2(a_1, a_2)} z_t(a_1, a_2) \right] \end{aligned} \quad (23)$$

Let $a_1 = s_0$ and a_2 be such that $z(a_1, a_2) = f$. Substituting a_1 and a_2 in Eq. (23) gives

$$\frac{fs}{z(s, t)} \approx s_0 + \left[(s - s_0) \frac{\partial F(s_0, a_2)}{\partial s} - (t - a_2) \frac{s}{z(s_0, a_2)} z_t(s_0, a_2) \right] \quad (24)$$

At the point $p = \mathbf{X}(s_0, t_0)$ Eq. (24) equals

$$\frac{fs_0}{z(s_0, t_0)} \approx s_0 - (t_0 - a_2) \frac{s_0}{z(s_0, a_2)} z_t(s_0, a_2) \quad (25)$$

Viewing the second term of Eq. (25) as the velocity term of $s + V^s(t)$, we see that at p the first component of Eq. (22) is approximated by Eq. (25). A similar result can be shown for the second component of Eq. (22), $\frac{f h(s)}{z(s, t)}$. Therefore, the model parameterization for a translating contour approximates the ST surface generated by a contour moving in depth under perspective projection.

Figures 13 through 15 show the results of applying our method to a sequence of images rotating in depth (under orthographic projection) and to a sequence where the camera is zooming into the image (under perspective projection). The motion in the sequences was synthetically produced by transforming each image.

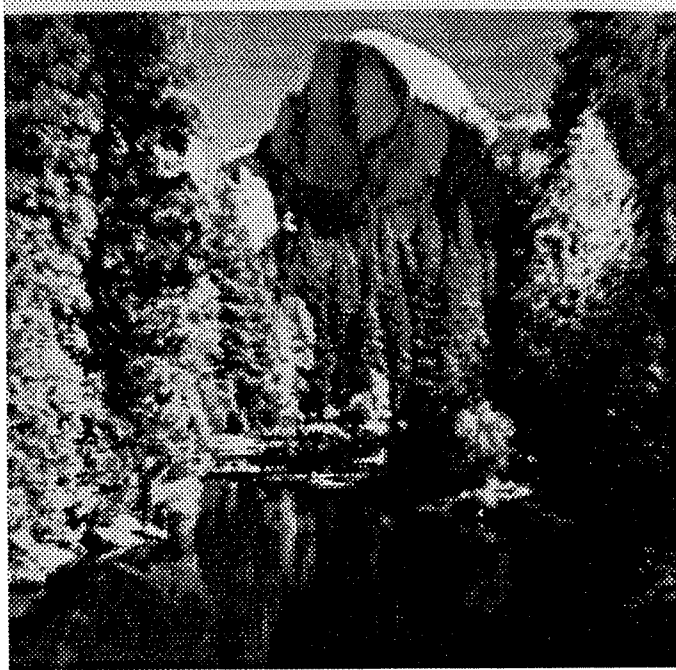
There are incorrect results in the lower right of Figure 14(b). This resulted because that section of the image sequence contains little variation in gray level (see Figure 13(a)). With little variation, the gradients were close to 0. Taking the cross product of the two gradient directions to get the direction of motion is noisy when the gradients are close to 0.

The top center of Figure 15(b) contains incorrect results which also resulted because of little variation in gray level. Although not visible in Figure 13(b), most of the sky region except the center contains clouds. In the cloud areas there was sufficient gray level variation for the correct result to be achieved, but in the non-cloud area there was not enough variation.

5.3. The Aperture Problem

Yamamoto [Yama89] showed that for translational and rotational motion parallel to the image plane, there is an aperture problem when the contour is a straight line and when there is rotation about the center of a circle. Consequently, our method should either be undefined or unable to recover the motion in these cases. In all other cases the true motion can be recovered even when a local method is used.

When using edge images, a straight line will force $E_s = 2 h_s(s, t)h_{ss}(s, t)$ to be 0. This results in $\nabla E = 0$. Since the gradient vector is 0, the perpendicular direction is undefined. A "straight line" in a gray level image occurs when the gradient direction of the image at neighboring points is the same. In this case, both functions defined on the ST surface will have all partials of second order equal to 0, i.e., $A_s = A_g = A_t = 0$ and similarly for B . This results in Eq. (20), the gradient equation, equaling 0.

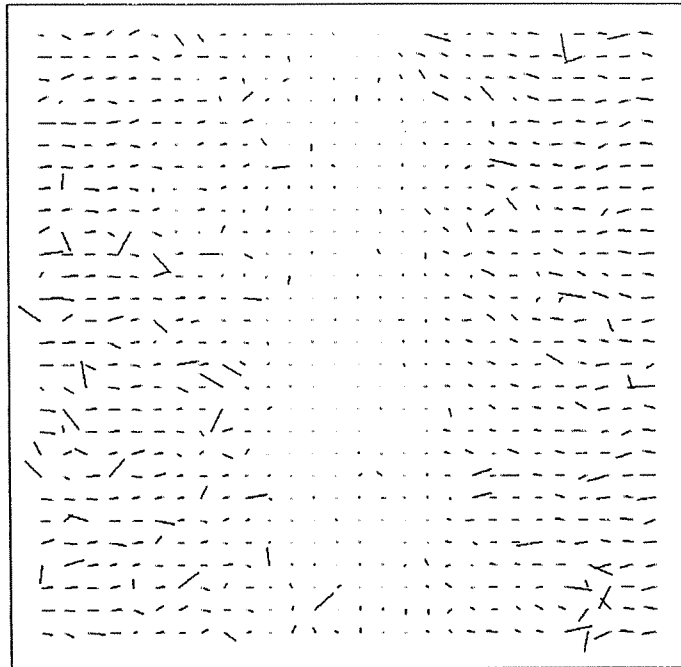


(a)

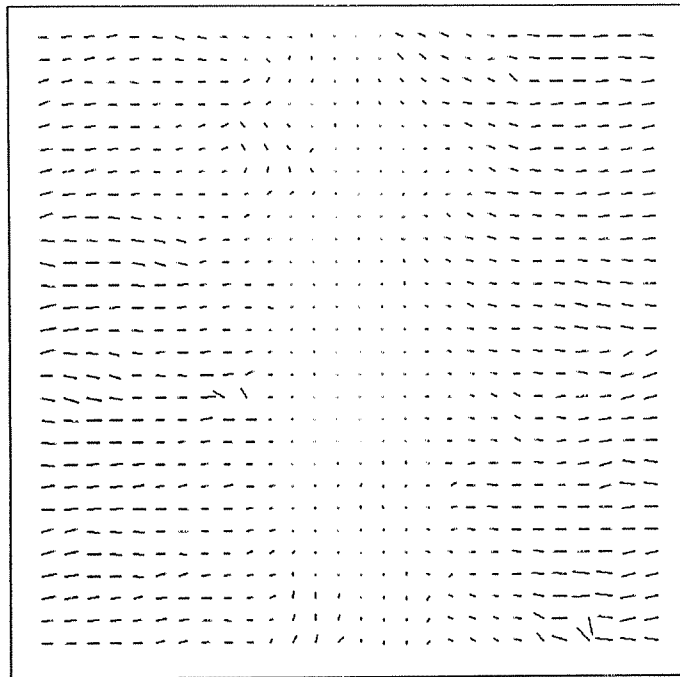


(b)

Figure 13. Two gray level images used in subsequent figures. (a) Halfdome; one frame from a 7 frame sequence of images in which no motion occurred. (b) Krakow; one frame from a 7 frame sequence of images in which no motion occurred.

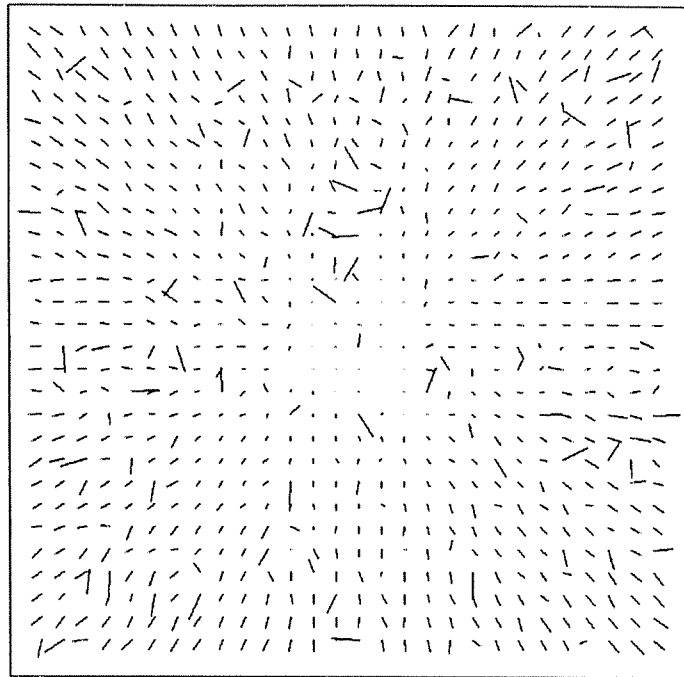


(a)

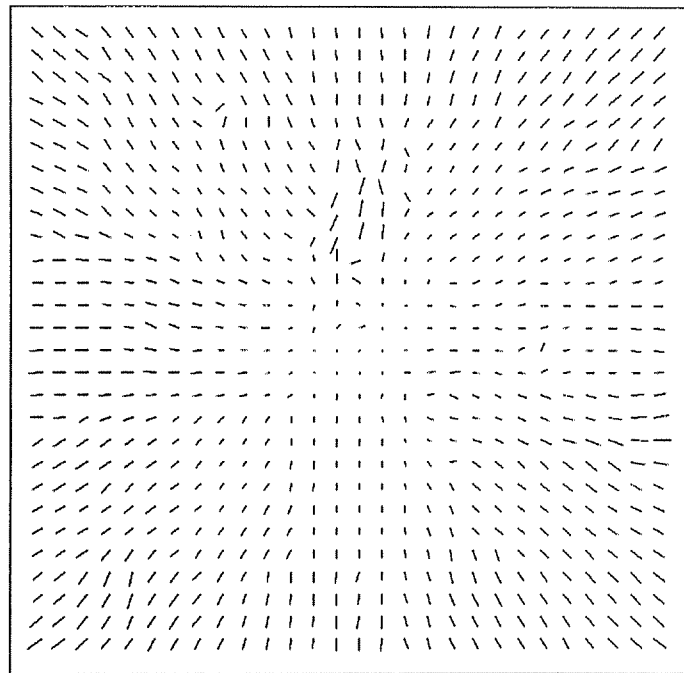


(b)

Figure 14. (a) ST surface flow for the middle frame of the Halfdome sequence, where motion was synthetically produced by rotating each frame about a vertical line through the middle of the image at 0.03 radians/frame. This resulted in movement of approximately 0.4 pixels/frame near the left and right edges. (b) The flow in (a) smoothed using a 3x3 kernel.



(a)



(b)

Figure 15. (a) ST surface flow for the middle frame of the Krakow sequence, where motion was synthetically produced by zooming into the center of each frame. The movement was approximately 0.6 pixels/frame near the edges. (b) The flow in (a) smoothed using a 3x3 kernel.

When there is circular motion about the center of an arc of a contour, there is no change in either the ST 2-surface or the ST 3-surface, so this will be interpreted as no motion. Similarly, if there is a translational component, only that component is recovered.

These are the only two cases where an aperture problem arises for a rigid contour in motion and our method, correctly, does not find a solution.

6. Concluding Remarks

A method was presented for recovering spatiotemporal surface flow, i.e., the instantaneous motion of all points on the generating contour of an ST surface. It was shown that the direction of motion of a point is perpendicular to the gradient direction of a function defined on an ST surface. This function measures the change in arc length on an ST surface. As a result, the motion can be solved for exactly. Because our formulation is well-posed, it is preferable to gradient-based optic flow methods which rely on the ill-posed optic flow constraint equation.

Our method has been implemented using the Monge patch parameterization of an ST surface and test results for five image sequences have been presented. The results show the successful computation of ST surface flow in both edge and gray level image sequences. The sequences using edge images required an intermediate step of segmenting edge points into ST surfaces. Because of this extra computation and possibility of errors, using gray level images is the preferable approach.

We have shown why our method approximates the correct flow in cases where object motion is not parallel to the image plane or when orthographic projection is not assumed. In the future, we plan to investigate ways to extend the ideas developed in this paper to include motion in depth and contour deformation so the flow in these cases can be computed exactly.

As in the gradient-based optic flow approach, our work assumes that any change in gray level through a sequence of images is due only to motion, i.e., every point in the scene remains the same intensity. We hope to remove, or at least relax, this assumption in future work. Consider an object rotating in depth. As it rotates, its surfaces change illumination. When this change is gradual, the ST surface will undergo a gradual rise or decline. Gray level variation will still exist, but will be at a higher or lower level depending upon the current illumination. By incorporating this rise and fall of the ST surface into our model parameterization we hope to eliminate the constant brightness assumption.

References

- [Agga88] Aggarwal, J. K. and N. Nandhakumar, On the computation of motion from sequences of images - A review, *Proc. IEEE* **76**, 1988, 917-935.
- [Allm90] Allmen, M. and Dyer, C. R., Cyclic motion detection using spatiotemporal surfaces and curves, *Proc. 10th Int. Conf. on Pattern Recognition*, 1990, to appear.
- [Bake89a] Baker, H. H., Building surfaces of evolution: the weaving wall, *Int. J. of Computer Vision* **3**, 1989, 51-71.
- [Bake89] Baker, H. H. and R. C. Bolles, Generalizing epipolar-plane image analysis on the spatiotemporal surface, *Int. J. of Computer Vision* **3**, 1989, 33-49.
- [Besl86] Besl, P. J. and R. C. Jain, Invariant surface characteristics for 3D object recognition in range images, *Computer Vision, Graphics, and Image Processing* **33**, 1986, 33-80.
- [Besl88] Besl, P. J. and R. C. Jain, Segmentation through variable-order surface fitting, *IEEE Trans. on Pattern Analysis and Machine Intelligence* **10**, 1988, 167-192.
- [Boll87] Bolles, R. C., H. H. Baker, and D. H. Marimont, Epipolar-plane image analysis: An approach to determining structure from motion, *Int. J. Computer Vision* **1**, 1987, 7-55.
- [Brad85] Brady, M., J. Ponce, A. Yuille, and H. Asada, Describing surfaces, *Proc. 2nd Int. Symp. on Robotics Research*, 1985, 5-16.
- [DoCa76] DoCarmo, M., *Differential Geometry of Curves and Surfaces*, Prentice-Hall, Englewood Cliffs, New Jersey, 1976.
- [Gold88] Goldgof, D. B., H. Lee, and T. S. Huang, Motion analysis of nonrigid surfaces, *Proc. Computer Vision and Pattern Recognition Conf.*, 1988, 375-380.
- [Jain88] Jain, R., Dynamic vision, *Proc. 9th Int. Conf. on Pattern Recognition*, 1988, 226-235.
- [Limb75] Limb, J. O. and J. A. Murphy, Estimating velocity of motion images in television signals, *Computer Graphics and Image Processing* **4**, 1975, 311-327.
- [Liou89] Liou, S. P. and R. C. Jain, Motion detection in spatio-temporal space, *Computer Vision, Graphics, and Image Processing* **45**, 1989, 227-250.
- [Medi84] Medioni, G. and R. Nevatia, Description of 3-D surfaces using curvature properties, *Proc. Image Understanding Workshop*, 1984, 291-299.
- [Peng89] Peng, S. L. and G. Medioni, Interpretation of image sequences by spatio-temporal analysis, *Proc. Computer Vision and Pattern Recognition*, 1989, 344-351.
- [Schu84] Schunck, B. G., The motion constraint equation for optical flow, *Proc. 7th Int. Conf. on Pattern Recognition*, 1984, 20-22.
- [Yama89] Yamamoto, M., A general aperture problem for direct estimation of 3-D motion parameters, *IEEE Trans. on Pattern Analysis and Machine Intelligence* **11**, 1989, 528-536.

