Interpretation of 3D Structure and Motion Using Structured Lighting

Y. F. Wang and Arvind Pandey
Department of Computer Science
University of California
Santa Barbara, CA 93106
yfwang@happy.ucsb.edu
(805) 961-3866

1 INTRODUCTION

In this paper, we propose a new formulation for analyzing time varying image sequences using structured lighting. The goal is to estimate the structure and motion parameters of 3D objects in arbitrary motion from a sequence of structured light coded images. The surface of the moving objects may be either planar or curved, and the mode of motion can be arbitrary. We propose to adopt an active sensing principle in our analysis in which a light pattern is projected onto the object surface to "encode" the surface for analysis. A structured light coded image is partitioned into small windows, and each window is assumed to be a view of a piece of a plane, of a sphere or of a cylinder. And these windows can be processed in parallel. The distortion of the projected pattern observed in an image is related to the surface orientation and structure of the imaged object. Furthermore as the object moves, the light pattern on the surface changes accordingly. Changes of the observed light pattern over time are utilized to compute the motion of the imaged object.

2 ANALYSIS OF STRUCTURED LIGHT IMAGES

The analysis of structured light images is accomplished in three stages: (1) First, the orientation and the structure of the imaged surface are recovered using the distortion of the projected pattern, (2) second, change in the orientation of the surface over time is related to the rotational motion of the surface, and (3) third, change in the perceived surface pattern over time is related to the translational motion of the surface.

In the first stage processing, we have developed a general algorithm to infer the surface orientation and structure based on the distortion of the projection pattern for arbitrary free-form surfaces [Wang87]. This algorithm computes the tangent directions of the projected stripes at the stripe junctions in the image, and relate the orientation of the surface stripes to that of the underlying surfaces in space. The structure of the imaged surface is then recovered by integrating the inferred orientation map.

In the second and third stages, i.e. to compute the motion of the imaged 3D object from the input image sequence, our approach is to relate the changes in the surface orientation and the projected pattern over time to the motion of the underlying surface. In such an analysis, one has to be cautious that even though changes of the observed pattern are induced by the motion of the underlying object, they do not necessarily reflect faithfully the motion of the imaged object for the following two reasons:

1. Not all modes of motion induce changes of the observed pattern.
2. Furthermore, changes of the surface properties at the stripe junctions are due to more than one factor.

When a sequence of structure light images are analyzed, changes of the observed pattern are related to the motion of the imaged object. However, motion of the imaged object can induce changes both in the appearance and in the sampling position of the projected pattern. Changes of the observed pattern are really due to two factors: motion of the underlying object and possible changes of surface structure at different sampling locations. Change of the sampling positions on planar surface patches may not be critical at all points share the same surface orientation. However, it becomes significant for objects with curved surfaces as the observed change is now due to a combination of two factors. Without some a priori knowledge or assumption about the surface structure, changes of the surface pattern in an image sequence due to object motion and due to different sampling positions may not be separable. And the problem may become untractable.

To solve the problems mentioned above, we make special assumptions about the surfaces that may appear in the scene. A structured light coded image is partitioned into small windows, and each window is assumed to be a view of a piece of a plane, of a cylinder or of a sphere. Furthermore, we assume that we can identify the type of surfaces viewed inside each window. These windows can then be processed in parallel by appropriate analysis algorithms to estimate the structure and motion of the surface viewed in the windows. In the following paragraphs, we justify the above assumptions.

The assumption on the limited surface type may appear to be restrictive at the first glance, but it can be argued that even though the surface structures of 3D objects may be arbitrarily complex, the class of surfaces comprising the 3D objects we encounter frequently is usually restricted to a much smaller domain. First, man-made objects are usually of simple, regular shapes. Complex surface/volume structures can be constructed as hierarchies of simple components. In fact, most geometric modeling systems using CSG (constructive solid geometry) technique provide only a small number of simple building primitive solids. It has also been observed that 85% of industrial parts can be perfectly represented or can be well approximated by a small number of patches of planes, spheres and cylinders. As the surface characteristics of planar, cylindrical and spherical patches are maintained throughout the surface, changes due to different sampling positions at different time instants can be compensated for. This assumption also enables us to partition images into small localized windows and process these windows in parallel. Hence, in stead of considering objects of unrestricted
surface structures, we will limit our attention to a few simple classes of surfaces and build more complex object structures as hierarchies of simpler constructs.

Further, we assume that we can identify the type of surfaces inside each window so that appropriate algorithms can be selected to analyze their structure and motion inside each window. Such discrimination can be accomplished using the intrinsic properties of the surfaces. As is well known in differential geometry, planar, spherical, and cylindrical surfaces exhibit distinct principal curvature measurements, i.e., for planar surfaces $\kappa_1 = \kappa_2 = 0$, for spherical surfaces $\kappa_1 = \kappa_2 = \text{constant} \neq 0$, and for cylindrical surfaces $\kappa_1 \neq 0$ and $\kappa_2 = 0$, where $\kappa_1$ and $\kappa_2$ denote the two principal surface curvatures. This property can be used to discriminate the three elementary surface types under consideration.

We have developed algorithms [Wang89a] [Wang89b] which accomplish such analyses.

2.1 System Configuration

We replace one of the cameras in a passive stereo setup with a projection device (a 3Dmum slide projector). A parallel projection model is assumed here to simplify the mathematics involved. A grid pattern marked on a slide is projected to "encode" the object surfaces for analysis. The projection pattern consists of two sets of mutually orthogonal stripes. The encoded object surface structures are recorded using a laser camera. It should be noted that this analysis technique based on structured lighting is nondestructive and the second sets of stripes, on an object surface, result from the projection of a single grid code on the object (temporal correlation) and the other one is based on the observation of the movement of four surface patches— all belong to the same rigid object over two successive frames (spatial correlation). Of course, a hybrid technique based on both temporal and spatial correlations is also possible.

2.2 Planar Surfaces in Motion

2.2.1 Recovering the Orientation of a Planar Surface

If the stripe orientations, $\hat{L}_1$, and $\hat{L}_2$ (where 1 and 2 denote the first and second sets of stripes), on an object surface, $P_s$, are recovered, then the surface normal direction $\hat{n}$ can be readily computed using

$$\hat{n} = \frac{\hat{L}_1 \times \hat{L}_2}{|\hat{L}_1 \times \hat{L}_2|}$$

(1)

where $\times$ denotes the cross product. In the gradient space where we represent a plane as $\hat{n} = px + qy$, $p$ and $q$ are $(p, q) = (\partial g/\partial x, \partial g/\partial y) = (-Po_x/\lambda, -Po_y/\lambda)$.

Let the stripe orientations observed in the image plane $P_i$ be $\hat{L}_1$ and $\hat{L}_2$ and the stripe orientations in the grid plane are $\hat{L}_1'$ and $\hat{L}_2'$. Let $\hat{L}_1'$ and $\hat{L}_2'$ be two small lements to denote 2D measurements. Then $\hat{L}_1'$ and $\hat{L}_2'$ can be computed using the following equation:

$$c_x = (c(M_{11}, x)) \times (c(M_{21}, x))$$

$$c_y = (c(M_{12}, y)) \times (c(M_{22}, y))$$

(2)

where $M$ is a transform matrix which maps the local coordinate associated with $P_i$ to the viewer-centered coordinate associated with $P$. $M_{11}$ is a transform matrix which aligns an zero to an image coordinate $(x, y)$ to make $(x, y)$.

2.2.2 Inferring the Rotational Component

Once the orientation of a planar surface $P_s$ is recovered, the rotational component of the object motion can be inferred from the change of the object surface orientation over time. We represent the rotational motion as a revolution of an angle $\Omega$ about an axis of orientation $\hat{N} = (N_x, N_y, N_z)$ passing through the origin. Here, we do not consider the special cases where $\hat{N}$ coincides with $\hat{L}_1$, or parallel to it. To do this we assume that $\hat{N}_1$, $\hat{N}_2$, and $\hat{N}_3$ can be measured from the image plane. Hence, $\hat{L}_1$, $\hat{L}_2$, and $\hat{L}_3$ can be computed using Equations 1 and 2.

2.3 Computing the Translational Component

One should note that there are degenerate translational motions which do not induce changes in the observed pattern, and hence are difficult to analyze using structured light images. The translational component of a surface which is perpendicular to the plane (or along the plane normal direction) does induce changes in the observed pattern, and can be readily computed from a sequence of grid coded images. The translational component which is parallel to the plane does not induce changes in the observed pattern and has to be estimated using other information source (such as tracking the motion of certain surface features).

Here, we will concentrate on general motion. By general motion we mean that $\Omega \neq 0$ and $\hat{N}$ and $\hat{P}$ are not parallel. Under general motion, we will show that the position of the surface and the translational component can both be determined. Here we introduce two techniques to accomplish this analysis. One technique is based on the observation of the movement of a single planar surface patch over five image frames (temporal correlation) and the other one is based on the observation of the movement of four surface patches—all belong to the same rigid object over two successive frames (spatial correlation).

2.3.1 Recovering the Translational Component from a Sequence of Grid Coded Images

Let the stripe orientations observed in the image plane $P_i$ be $\hat{L}_1'$ and $\hat{L}_2'$ and the stripe orientations in the grid plane be $\hat{L}_1$ and $\hat{L}_2$. We use small lements to denote 2D measurements. Then $\hat{L}_1'$ and $\hat{L}_2'$ can be computed using the following equation:

$$c_x = (c(M_{11}, x)) \times (c(M_{21}, x))$$

$$c_y = (c(M_{12}, y)) \times (c(M_{22}, y))$$

(2)

where $M$ is a transform matrix which maps the local coordinate associated with $P_i$ to the viewer-centered coordinate associated with $P$. $M_{11}$ is a transform matrix which aligns an zero to an image coordinate $(x, y)$ to make $(x, y)$.

2.3.2 Temporal Correlation

Consider a sequence of $K + 1$ images taken at times $1, 2, \ldots , K$ and $\hat{L}_1'$ and $\hat{L}_2'$ can be computed from a sequence of grid coded images. The translational component which is parallel to the plane does not induce changes in the observed pattern and has to be estimated using other information source (such as tracking the motion of certain surface features).

2.3.3 Spatial Correlation

One should note that there are degenerate translational motions which do not induce changes in the observed pattern, and hence are difficult to analyze using structured light images. The translational component of a surface which is perpendicular to the plane (or along the plane normal direction) does induce changes in the observed pattern, and can be readily computed from a sequence of grid coded images. The translational component which is parallel to the plane does not induce changes in the observed pattern and has to be estimated using other information source (such as tracking the motion of certain surface features).

Here, we will concentrate on general motion. By general motion we mean that $\Omega \neq 0$ and $\hat{N}$ and $\hat{P}$ are not parallel. Under general motion, we will show that the position of the surface and the translational component can both be determined. Here we introduce two techniques to accomplish this analysis. One technique is based on the observation of the movement of a single planar surface patch over five image frames (temporal correlation) and the other one is based on the observation of the movement of four surface patches—all belong to the same rigid object over two successive frames (spatial correlation). Of course, a hybrid technique based on both temporal and spatial correlations is also possible.

Temporal correlation. Consider a sequence of $K + 1$ images taken at times $1, 2, \ldots , K$ and $\hat{L}_1'$ and $\hat{L}_2'$ can be computed from a sequence of grid coded images. The translational component which is parallel to the plane does not induce changes in the observed pattern and has to be estimated using other information source (such as tracking the motion of certain surface features).

2.3.4 Rotational Component

Once the orientation of a planar surface $P_s$ is recovered, the rotational component of the object motion can be inferred from the change of the object surface orientation over time. We represent the rotational motion as a revolution of an angle $\Omega$ about an axis of orientation $\hat{N} = (N_x, N_y, N_z)$ passing through the origin. Here, we do not consider the special cases where $\hat{N}$ coincides with $\hat{L}_1$, or parallel to it. To do this we assume that $\hat{N}_1$, $\hat{N}_2$, and $\hat{N}_3$ can be measured from the image plane. Hence, $\hat{L}_1$, $\hat{L}_2$, and $\hat{L}_3$ can be computed using Equations 1 and 2.

2.3.5 Spatial Correlation

One should note that there are degenerate translational motions which do not induce changes in the observed pattern, and hence are difficult to analyze using structured light images. The translational component of a surface which is perpendicular to the plane (or along the plane normal direction) does induce changes in the observed pattern, and can be readily computed from a sequence of grid coded images. The translational component which is parallel to the plane does not induce changes in the observed pattern and has to be estimated using other information source (such as tracking the motion of certain surface features).

Here, we will concentrate on general motion. By general motion we mean that $\Omega \neq 0$ and $\hat{N}$ and $\hat{P}$ are not parallel. Under general motion, we will show that the position of the surface and the translational component can both be determined. Here we introduce two techniques to accomplish this analysis. One technique is based on the observation of the movement of a single planar surface patch over five image frames (temporal correlation) and the other one is based on the observation of the movement of four surface patches—all belong to the same rigid object over two successive frames (spatial correlation). Of course, a hybrid technique based on both temporal and spatial correlations is also possible.
Hence,

\[
\begin{align*}
&\text{for } k = 1, \ldots, K \text{ can be obtained from the input image sequence. } P_k \text{ is derived from the calibration process, and } P^{(+)}_k \text{ and } P^{(-)}_k \text{ can be recovered using the technique introduced in Section 2.2.1. It is computed using } P_k \text{ over three successive frames (Section 2.2.2). Therefore, only } T_1, T_2, T_3, T_4 \text{ and } T_5 \text{ are unknown in Equation 9. Hence, information from five consecutive frames can theoretically determine these four unknowns.}
\end{align*}
\]

Notice that we can recover \( T \) in a general motion but only \( T_4 \) in a translational motion with structured lighting. The reason for that is the \( T_4 \) component stays the same in a translational only motion because \( P \), does not change. This component is therefore not observable in the structured light projection technique. However, if the motion is such that the rotation angle \( \Omega \) is not zero and the rotation axis \( \mathbf{N} \) is not parallel to \( P_k \), then \( T_4 \) changes with time. Hence, \( T_4 \) component also changes. The parallel component, \( T^{(\|)} \), at time \( t \), \( T^{(\|)}_t \), may become observable in the perpendicular component, \( T^{(\perp)}_t \), say, at time \( t + n \). Hence, the motion can be recovered in its entirety.

Spatial correlation. Another way to find the position of \( P_k \) and the translation component of the object motion is to use a number of surfaces of a single rigid object. Suppose a rigid object is made of planar surface patches and \( K \) of them are visible from two successive frames at times \( t \) and \( t' \). We simultaneously track the movement of \( K \) junctions, \( f^{(9)}, f^{(1)}, \ldots, f^{(K-1)} \) in the image plane, one on each surface patch over two successive frames at time \( t \) and \( t' \). Then it can be shown that if \( K \geq 4 \), the motion of the imaged object can be computed. This method requires lesser number of image frames to compute the motion parameters, and is discussed in greater detail in [Wang86].

### 2.3 Cylindrical Surfaces in Motion

#### 2.3.1 Recovering the Structural Parameters of a Cylinder

Given a plane curve — not a straight line — and through each point on the curve we draw a line perpendicular to the plane. The surface generated by these lines is called a cylinder. The lines are its rulings or generators, and the given curve its directrix. In the following discussion, we will concentrate on cylinders with circular directrix (the general definition of "cylinders"). For this particular type of cylinders, one can imagine that they are made of an infinite number of circular disks stacked on top of each other. The axis of such a cylinder is defined to be the line which pass through all the disk centers. We refer to the direction of the axis and the radius of the directrix as the structure parameters of a circular cylinder.

Finding the axis orientation of a cylinder. Refer to Figure 1, a stripe \( l \) in the grid plane is being projected onto a cylindrical surface, \( S \). The resulting curve, \( C_0 \), on the cylinder surface — when viewed from the image plane — appears as another curve \( c \), where small \( c \) denotes quantity measured in the 3D image plane. Notice that the projection of a stripe from \( P \) generates a sheet of light. This light sheet intersects the cylindrical surface in a section \( C_0 \), which is a conic of the same type as the directrix \( C_0 \), of the cylinder. Furthermore, the projection of the section \( C_0 \) onto the image plane creates, again, a conic of the same type as the section — hence the same type as the directrix of the cylinder. Denote the discriminants of the directrix \( C_0 \), a section \( C_0 \), and the projection of the section \( c \) as \( \Delta_4 \), \( \Delta_5 \), and \( \Delta_6 \), respectively. We have

\[
\Delta_3 = \Delta_5 \cos^2 \theta_p, \quad \Delta_4 = \Delta_6 \cos^2 \theta_p, \quad \Rightarrow \Delta_4 = \Delta_k [\cos \theta_p \cos \theta_p],
\]

where \( \theta_p \) is the angle between the light sheet from the grid plane (where the section \( C_0 \) lies) and the plane containing the directrix of the cylinder, and \( \theta_p \) is the angle between the image plane and the plane of the section. Denote the direction of the cylinder axis by \( \tilde{Z} = (Z_x, Z_y, Z_z) \) (for reasons which will become clear later), we can compute cosine of

\[
\cos \theta_p = \frac{\bar{M}_1^T (\tilde{P} - P_k) - \bar{Z}}{\bar{M}_1^T (\tilde{P} - P_k)}
\]

and

\[
\cos \theta_p = \frac{\bar{M}_1^T (\tilde{P} - P_k) - \bar{Z}}{\bar{M}_1^T (\tilde{P} - P_k)}
\]

for each point where

\[
\bar{M}_1^T (\tilde{P} - P_k) - \bar{Z} = 0
\]

Figure 1: Finding the structure parameters of a cylinder using active sensing.

Again, if the imaging system has been properly calibrated, \( M_k \) is known, so are \( \bar{P}_k \) and \( \bar{I}_k \), and \( \bar{P}_k \) is \((0, 0, -1)\) in a viewer-centered coordinate. Furthermore, \( \Delta_k \) is, in the case a circular directrix, \( -4 \). Hence, if \( \Delta_k \) is known, Equation 10 provides a constraint on \( \Delta_k \).

\[
\Delta_k \text{ can be computed in an image by approximating the projected stripe patterns with conic curves. Refer to Figure 1, under proper lighting condition, } q \text{ presents sharply contrasted feature (may be only partially complete) in an image. We can apply an edge or line detector to extract points along } c. \text{ Suppose that } K = 4 \text{ such points, } (x_k, y_k), k = 0, 1, \ldots, K, \text{ are extracted. Now, we know that they must lie on a conic of the form: } z^2 + bky + cz^2 + dxz + eyz + f = 0, \text{ where the leading coefficient is normalized to 1. The mean-square-error, } E, \text{ can be computed on the fitting: }
\]

\[
E(k, c, d, e, f) = \sum_{k=0}^{K} (x_k^2 + bky_k + cz_k^2 + dx_kz_k + ey_kz_k + f)^2
\]

For then the error to be minimized, \( \partial E/\partial b, \ldots, \partial E/\partial f \) should all be zero. Hence a linear system of equations can be derived to solve for \( b, c, d, e, f \) and \( \Delta_k \), \( \Delta_k \) can be computed.

Now, we have two sets of stripes of mutually orthogonal directions on \( P \). Projection of these two stripe sets generate sections \( C_1 \) and \( C_2 \) on a cylinder surface and conic curves \( c_1 \) and \( c_2 \) in the image plane, where \( l_1 \) and \( l_2 \) denote the first and second stripe sets. Therefore, two sets of constraints on \( \Delta_2 \), based on the measurements of \( \Delta_2 \) and \( \Delta_3 \) in the image plane, can be obtained. Furthermore, we know that \( \Delta_2 + \Delta_3 + \Delta_6 = 1 \). Hence, we can estimate \( \Delta_6 \).

In case the stripe projection does not result in a section (because the sheet of light happens to be parallel to the rulings of a cylinder), we need a different constraint. Since the two sets of stripes are mutually orthogonal, at most one set of stripes can intersect a cylinder along its rulings. The other set intersect the cylinder in sections. The constraint in Equation 10 holds for the stripe set which intersect the cylinder in a section. For the stripe which intersects the cylinder along its ruling, we have \( (\bar{M}_1^T (\tilde{P} - P_k) \cdot \bar{Z} = 0 \) as the constraint. The above equation states the fact that the axis of a cylinder, \( \tilde{Z} \), is parallel to the rulings of the cylinder. Hence it is perpendicular to the normal of the light sheet.

Finding the radius of the directrix. To estimate the radius of the directrix, we rely on the fact that every section \( C_0 \) of a circular cylinder is an ellipse. Furthermore, the length of the minor axis, \( \bar{A}_{\text{minor}} \), of any elliptical section \( C_0 \) is the same as the diameter of the directrix \( C_0 \). The direction of the minor axis, \( \bar{A}_{\text{minor}} \), is parallel to the line where
the planes of \( C_3 \) and \( C_4 \) intersect. We can compute the direction of the minor axis of a section, \( C_3 \), in space as

\[
\hat{a}_{\text{minor}} = (A_x, A_y, A_z) = \left( \frac{(M_y p_3^3) \times p_2^3}{| (M_y p_3^3) \times p_2^3 |} \right)
\]  

(13)

Refer to Figure 1 again, the projection of \( a_{\text{minor}} \) in an image is of direction \( a_{\text{minor}} = (A_x, A_y) \), the length of which, \( | a_{\text{minor}} | \), can be measured from the conic fitting in the image plane. It can be shown that \( | a_{\text{minor}} | = 2 \sqrt{A_x^2 + A_y^2} \) where

\[
t = \sqrt{\left( A_x^2 + B_x A_y + c A_y^2 + d A_x + e A_y + f \right) / \left( A_x^2 + B_x A_y + c A_y^2 + d A_x + e A_y + f \right)} + r
\]  

(14)

where \( r \) is the radius of the directrix of the cylinder, and

\[
i_p = \left( 2 c d - h e + 2 A_x A_y \right) / 2 A_x^y, \quad i_y = \left( 2 c - h d / A_x^y \right)
\]  

(15)

and \( \hat{i} = (i_x, i_y) \) is the center of the conic section in the image.

2.3.2 Inferring the Rotational Component

Once the orientation of the axis of a cylinder is recovered, the rotational component of the cylinder motion can be inferred from the change of the orientation of the cylinder axis over time. Using the same convention as in section 2.2.2 to represent the rotational motion and denote the axis orientation of the cylinder at time \( t \) as \( \hat{\theta}(t) \), then we have:

\[
\hat{\theta}(t+1) - \hat{\theta}(t) = \hat{\theta}(t) \hat{\theta}(t+1) - \hat{\theta}(t+1) = \hat{\theta}(t+1) - \hat{\theta}(t)
\]  

(16)

Furthermore, we know \( N_x^2 + N_y^2 + N_z^2 = 1 \). Hence, \( \hat{\theta}(t) \) estimated at three consecutive image frames at time \( t \), \( t+1 \), and \( t+2 \) (section 2.3.1) enable us to recover the direction of the rotation axis \( \hat{N} \). Once \( \hat{N} \) is obtained, the rotation angle \( \Omega \) and the rotation matrix \( R \) can also be computed using Equations 3 and 4.

2.3.3 Computing the Translational Component

Again, we will discuss only the general motion here. Our technique is based on the observation of the movement of a single cylindrical surface over five image frames. Refer to Figure 2 which shows a sequence of \( K + 1 \) images taken at times \( t, t+1, \ldots, t+K \) of a cylinder in general motion. A light pattern is projected onto the object surface and the movement of a particular grid junction \( J_i(i) \) is tracked in an image sequence and its image coordinates \((J_x(i), J_y(i)), (J_x(i)+K), (J_y(i)+K)\) at times \( t \) to \( t+K \) recorded. Now define

\[
J_x^{(t+K)} = f(t+K) + r \cos(\alpha(t+K)) + \sin(\alpha(t+K)) J_y^{(t+K)}
\]  

(18)

where \( J_x \) denotes the center of the circular disk where \( J_x \) lies. Again, these junctions, \( J_x(i), \ldots, J_x(i+K) \), result from the projection of a single junction, \( J_x \), from the grid plane. The spatial position of \( J_x \) (on the cylinder surface) changes with time because of the motion of the underlying object. However, all junctions must lie on the same line of projection (Figure 2), or

\[
J_x^{(t+K)} - J_x^{(t)} = t^K, J_y^{(t+K)} - J_y^{(t)}
\]  

(19)

Now we know that \( r \) and \( \hat{Z}_x \) (hence \( X_c \) and \( Y_c \)) can be estimated using the technique introduced in Section 2.3.1. We will show that \( \alpha^{(t+K)} \) and \( \hat{Z}_x^{(t+K)} \) can also be estimated from the input image sequence for \( k = 1, \ldots, K \). Hence \( F^{(t+K)} \) can be found from Equation 19. \( F^{(t+K)} - F^{(t)} \) constitutes a sequence of vectors which can be inferred from structured light images, and they will be used as bases for computing the translation of a cylinder.

For \( J_x^{(t+K)} - J_x^{(t)} \), the \( x \) and \( y \) components of \( J_x^{(t+K)} \) and \( J_x^{(t)} \) can be estimated from the input image sequence and the difference in the \( x \) component from:

\[
J_x^{(t+K)} - J_x^{(t)} = J_x^{(t)} + J_x^{(t+1)}
\]  

(20)

or

\[
J_x^{(t+K)} - J_x^{(t)} = (J_x^{(t+K)} - J_x^{(t)}) + (J_x^{(t+K)} - J_x^{(t+1)})
\]  

(21)

Furthermore, \( \alpha^{(t+K)} \) and \( \hat{Z}_x^{(t+K)} \) can be estimated from the projected image position, \( J_x \), of \( J_x \) onto the image plane to obtain \( J_x \), the coordinate of \( J_x \) can be expressed as

\[
\hat{Z}_x^{(t+K)} = J_x^{(t+K)} + r \cos(\alpha^{(t+K)}) J_y^{(t+K)} + r \sin(\alpha^{(t+K)}) J_x^{(t+K)}
\]  

(22)

where small letters \( J_x, J_y, J_x \) and \( J_y \) denote 2D image measurements of the corresponding quantities \( J_x, J_y, J_x \) and \( J_y \) in space. Similarly, we can find the cylinder axis in the image plane which is \( J_x^{(t+K)} = (J_x^{(t+K)} + J_x^{(t+K)}) / 2 \), where \( R_{c1} \) denotes the direction of a unit vector which is perpendicular to the image direction of \( J_x^{(t+K)} = (J_x^{(t+K)} + J_y^{(t+K)}) / 2 \) for \( k = 0, \ldots, K \). Now consider the distance from \( J_x^{(t+K)} \) to \( J_x^{(t+K)} \), it can be written as

\[
J_x^{(t+K)} - J_x^{(t)} = [ \cos(\alpha^{(t+K)}) J_x^{(t+K)} + \sin(\alpha^{(t+K)}) J_x^{(t+K)}] / \sqrt{J_x^{(t+K)} - J_x^{(t+K)}}
\]  

(23)

The particular choice of \( X_c, Y_c, Z_x \) ensures that \( J_y \), \( Z_c \), and \( J_z \) on the cylinder axis can be related to the angle \( \alpha \).

\[
\cos(\alpha^{(t+K)}) = \sqrt{J_x^{(t+K)} - J_x^{(t+K)}} / \sqrt{J_x^{(t+K)} - J_x^{(t+K)}}
\]  

(24)

Hence, \( F^{(t+K)} - F^{(t)} \) can be computed from Equation 19.

Now we can generate another sequence of points, \( J_x \), along the axis of a cylinder in the following manner: Refer to Figure 2, assume that at time \( t \) the position of the cylinder is specified by \( F^{(t)} = (J_x^{(t)}, J_y^{(t)}, J_z^{(t)}) \) which denotes a point on the axis \( J_x \) at time \( t \). Denote the rotation matrix by \( R \) and translation vector by \( T \). Then we can generate an "imaginary" sequence of points, \( F^{(t+1)}, \ldots, F^{(t+K)} \) - based on \( F^{(t)} \) - which lie on the axis from time \( t \) to \( t+K \).

\[
\begin{align*}
\text{at time } t+1 & : F^{(t+1)} = F^{(t)} + T \\
\text{at time } t+2 & : F^{(t+2)} = R F^{(t+1)} + T \\
\text{at time } \ldots & : \ldots \\
\text{at time } t+K & : F^{(t+K)} = R F^{(t+K-1)} + T \\
\end{align*}
\]  

(25)

Now we have two motion sequences along the axis of the cylinder: \( F^{(t)}, F^{(t+1)}, \ldots, F^{(t+K)} \) and \( F^{(t)}, F^{(t+1)}, \ldots, F^{(t+K)} \). The former sequence re-

Figure 2: A cylinder undergoes general motion

a local coordinate system associated with the cylinder: the origin of the local system lies on \( Z_x \), and \( X_c, Y_c \) are defined as follows:

\[
X_x = (-Z_d, Z_d, 0), \quad Y_x = (Z_d X_z, Z_d Y_z, -Z_d X_0 + Z_d Y_0)
\]  

(17)

It can be seen that \( X_x, Y_x, Z_x \) are orthogonal to one another. The position of \( J_x^{(t+K)} \) at time \( t+K \) can be expressed in this coordinate system as:
results from the structured light projection and the observed motion of the light pattern in an image sequence while the $F$ sequence results from the computation based on the unknown rotation and translation components of the object motion. When the motion of the light pattern is tracked, the $I$ sequence — or more precisely, the $(R^{(t+k)}, \omega^{(t+k)}, T^{(t+k)})$ sequence — is what we can infer from the image observation. The $F$ sequence is expressed in terms of the unknowns $r^{(t+k)}, \omega^{(t+k)}, T^{(t+k)}$. Now we should find a way to relate the $I$ sequence (observables) to the $F$ sequence (unknowns).

Notice that the $F$ sequence is an imaginary sequence, the exact position of $F(0)$ does not matter as long as it lies on the axis at time $t$. So let $F(0) = F(0)$. Furthermore, $\hat{r}$ and $\hat{\omega}$ can be estimated from Equation 22, which implies that only $\hat{I}^{(t+k)}$ is unknown. Now, even though $I$ and $F$ are two distinct sequences of points, $F^{(t+k)}$ and $F^{(t+k)}$ do have something in common — they both lie on the axis of the cylinder at time $t + k$. If we compute the distance from $F(0)$ (or equivalently, $F(0)$) to $Z^{(t+k)}$, this distance measurement can be written in two different forms, one based on $F^{(t+k)}$ and the other one on $F^{(t+k)}$ (since both lie on $Z^{(t+k)}$)

\[
\phi^{(t+k)} = \phi^{(t+k)} = (r^{(t+k)} - r^{(t+k)}) T^{(t+k)} T^{(t+k)} T^{(t+k)}
\]

or

\[
\phi^{(t+k)} = \phi^{(t+k)} = (r^{(t+k)} - r^{(t+k)}) T^{(t+k)} T^{(t+k)} T^{(t+k)}
\]

Now $F^{(t+k)} = F(0)$ is a quantity which we can infer from the input image sequence and $Z^{(t+k)}$ can be inferred using the technique introduced in Section 2.3.1. Only $F^{(t+k)}$ is expressed in terms of the unknowns $T_1, T_2, T_3$ and $I^{(t+k)}$, or

\[
F^{(t+k)} = F(0) + (R^{(t+k)} - I^{(t+k)}) + (R^{(t+k)} - I^{(t+k)}) + R^{(t+k)} - I^{(t+k)}
\]

Notice that Equation 28 is a vector equation with 3 scalar components in $x, y, z$. Hence, a minimum of three consecutive views are needed to provide six scalar equations to compute all the unknown parameters. Notice that Equations 27 and 28 are linear in terms of the unknowns and that we can recover $T$ in a general motion but only $T_1$ in a translational motion with cylinders. The reason is again that under general motion, the parallel component, $T^{(t+k)}$, at time $t + m$ may become observable in the perpendicular component, $T_1^{(t+k)}$, say, at time $t + n$. Hence, the motion can be recovered in its entirety.

2.4 Spherical Surfaces in Motion

In the case of the spheres, the structure parameter refers only to the radius of a sphere, as spheres do not have a dominant axis direction like the cylinders. We estimate the radius of a sphere from its encoding pattern. The equation and the center position of an ellipse in the image plane are estimated from the observation of an active circular section on the sphere surface, which is produced by the projected grid pattern. The position of the center of the sphere in the image is recovered next by identifying two sets of segments joining active circle centers from two stripe sets. The distance in space, between the center of the sphere and the center of the active circle which projects into the ellipse under consideration is estimated. The radius of the active circle is estimated from the conic fitting in the image plane, which is then used to compute the radius of the sphere. We track the motion of the center of the sphere in space to recover the motion parameters. The details can be found in [Wang98].

3 EXPERIMENTAL RESULTS

3.1 Results from synthetic images

The algorithms described in the previous sections have been tested on both synthetically generated image sequences and image sequences which depict real objects in motion taken using a digitizing camera. Synthetic image sequences which depict moving objects of planar, spherical and cylindrical surface structure were generated to test our algorithm. A particular example is shown in Figure 3 which depicts a moving cube. For synthetic images, the location and the orientation of the light source, the orientation of the grid plane and parameters of the stripe in the grid plane, the spacing between the stripes, and the surface structure of the test object can be manipulated. Synthetic image sequences were used to verify the correctness of the analysis algorithms under controlled conditions, to identify the possible sources of errors, and to determine how these errors affect various stages in the structure and motion computations. Experiments using real objects and projection setup were conducted later after we gained a better understanding of the numerical stability of the algorithm.

The analysis results on the moving cube (Figure 3) are tabulated in Table 1. The table consists of three parts: part (a) tabulates the computed structural parameters (the surface normal of the planar surfaces of the cube), Table 1(b), the computed angle of rotation $\theta$ and the axis of rotation $N$ are tabulated together with their actual values. As discussed in the previous sections, computing the translational motion requires information from at least five frames for planar surfaces, four frames for spherical surfaces, and three frames for cylindrical surfaces. Hence the translational motion was computed only when a sufficient number of image frames became available. In our experiment, only one estimate of the translational motion was obtained from the image sequences, while more than one estimate of the structural parameters and the rotational parameters were computed. In Table 1(c) the translation vector $(T_1, T_2, T_3)$ computed by our algorithm is shown along with the actual values.

In Table 1, only one surface of the cube was used in the computation of the structure and motion parameters. Hence, only temporal correlation information was used and five image frames (shown in Figure 3) were required. Table 1(a) also shows the location of the grid point from the grid junction used in the computation of the translational motion. The motion of this grid junction was tracked through all image frames. The change in the surface normal in successive image frames indicated a rotational motion of the cube, while the changing position of the grid junction indicated a possible translational motion. Both the actual and computed motion parameters as well as the corresponding image frame number are shown in Table 1(b) to judge an idea of the accuracy of the results. It can be seen that both the rotation angle and the translational parameters can be computed with a high degree of accuracy for planar surfaces.

We also experimented with synthetic images with cylindrical and spherical surfaces. In our study we observed that for planar and spherical surfaces, the structural parameters could be computed to within ±5% of the actual values. The error in computing the motion parameters was about ±10%. In the case of cylindrical surfaces, the error was within ±5% in the estimated axis orientation, but jumped up to ±15% for the computed radius, and to ±20% for the computed motion parameters. In general, the computation of the motion parameters showed larger error because the motion parameters were computed based on the values of the structural parameters in our algorithm. Hence, error in the computation of the structural parameters propagated, and at times magnified, in the computation of the motion parameters.

3.2 Results from real images

We have also tested our algorithms on camera-acquired image sequences of real objects. The imaged objects were placed on a table which was about 5m in front of the camera and about 5.5m from the projector. The distance from the camera to the projector (or the baseline separation) was about 1.5m. The focal axes of both the camera and the projector was set parallel to the ground and made an 18° angle in the plane parallel to the ground. Two sets of parallel stripes were used. Each made a 45° angle with respect to the ground and were mutually orthogonal. Imaged objects were moved manually on the supporting table so that the motion was precisely measured and recorded for comparison. At fixed locations, pairs of images were taken by projecting these two sets of parallel stripes. These pairs of structured light images were then superimposed on each other to generate the image sequences shown in Figures 4 and 5.

Since we adopted a parallel projection model in our formulation, no particular care was taken to calibrate the parameters of the camera. Instead, the imaged objects were placed far away (≈5m) — comparing to the focal length (≈20) of the camera—from the camera so that the parallel projection model was satisfied.

In Figure 4, a volleyball in motion is shown, while the set of images in Figure 5 show a toy robot in motion. The toy robot has a much more complex surface structure. The robot has parallel spherical surfaces, a cylindrical middle section, and a spherical head on top. The computed and the actual values of the structure and motion parameters for the
vocabulary are tabulated in Table 6, while those for the toy robot are shown in Table 7. In both Tables 6 and 7, we show the computed values in pixels and in centimeters. The image configuration was calibrated to obtain such a conversion factor. Note that in order to hold the objects steady on the supporting table, both the volleyball and the toy robot were translated horizontally only. There was no rotational motion component.

4 THE CONCLUDING REMARKS

In our experiments we use a simple algorithm to establish the correspondence of grid junctions in successive image frames, i.e. matching a grid junction in the current image frame to the closest grid junction in the previous image frame. We derive below the condition under which this simple scheme gives correct match.

Consider a small neighborhood around a projected stripe junction and approximate the surface structure in that neighborhood by its tangent plane. We decompose the translational motion into two components: $T_a$ and $T_r$. From the discussion in Section 2, we know that $T_a$ will not induce change in the observed pattern. In order for the match (the physically closest ones at adjacent frames) to be correctly established in the case of translational only motion, it must be the case that $T_a$ has to satisfy the constraint:

$$\frac{P_{a1}P_{a2}}{2P_{a1}P_{a2} \sin \theta} \leq T_a \leq \frac{P_{a1}P_{a2}}{2P_{a1}P_{a2} \sin \theta}$$

(29)

where $P_{a1}$ makes an angle $\theta$ with respect to the global $z$-axis and the stripes in the grid plane make an angle $\theta$ with respect to the $x-z$ plane, and $d$ is the spacing between stripes in the grid plane. This equation establishes the upper bound on how fast the imaged object can move between two snap shots (separated by 1 sec) in terms of the stripe spacing, and the orientations of the object surface and the grid plane. If the motion of the imaged surface satisfies the above constraint, then the grid correspondence problem can be simplified by matching a grid junction in the previous frame to the grid junction in the current frame which is physically closest.

Our experience indicates that the spacing between the grid lines should be at least four to five times smaller than the object feature size. This way, a sufficient number of features (grid junctions and lines) can be observed on the object surface to guide the analysis. Furthermore, it is not necessary to track the motion of the same grid junction over all the image frames to estimate the translation parameters. Different grid junctions can be tracked for different pairs of images to solve for the translational parameters.

It is obvious that this algorithm will fail if the surface patch being tracked becomes occluded or moves outside the field of view before a sufficient number of images can be acquired. This problem can generally be avoided by sampling the environment fast enough. Due to the low computation costs involved in this algorithm, the image sampling rate can be made quite high. Furthermore, tracking more than one surface patch on the same image object can also reduce the number of image frames needed in the analysis. Tracking multiple surfaces of the same object can also improve the reliability of the result. A different windows (surface patches) on the same object can be tracked in parallel, and the structure and motion parameters can be computed independently for each window. The results can be used to cross-verify each other.

References


Table 1: The estimated parameters for a sequence of 5 image frames of a cube in general motion using only temporal correlation.

(c)

Table 2: The estimated parameters for a sequence of 5 image frames of a cube in general motion using only temporal correlation, with random noise of ±2 pixels added in the image measurements.

(a)

(b)

(c)
Table 3: The estimated parameters for a sequence of 4 image frames for a volleyball in motion

<table>
<thead>
<tr>
<th>parameters</th>
<th>computed values</th>
<th>actual values</th>
</tr>
</thead>
<tbody>
<tr>
<td>volleyball radius</td>
<td>93 pixels (98&quot;)</td>
<td>100mm</td>
</tr>
<tr>
<td>rotational angle</td>
<td>0.0°</td>
<td>0.0°</td>
</tr>
<tr>
<td>translational parameters</td>
<td>[15pix,0,7pix]</td>
<td>[18.82mm,0.74mm]</td>
</tr>
</tbody>
</table>

Table 4: The estimated parameters for a sequence of 5 image frames for a toy robot in a motion

<table>
<thead>
<tr>
<th>parameters</th>
<th>computed values</th>
<th>actual values</th>
</tr>
</thead>
<tbody>
<tr>
<td>robot radius</td>
<td>94 pixels (99.7mm)</td>
<td>115mm</td>
</tr>
<tr>
<td>rotational angle</td>
<td>0.9°</td>
<td>0.9°</td>
</tr>
<tr>
<td>translational parameters (1st section)</td>
<td>[14.69mm,0.73mm]</td>
<td>[16mm,0mm,9mm]</td>
</tr>
<tr>
<td>translational parameters (2nd section)</td>
<td>[14.69mm,0.73mm]</td>
<td>[18mm,0mm,9mm]</td>
</tr>
</tbody>
</table>

Figure 3: The cube moving by (50,0,0), and rotating about (1,1,1) by 12.0° per frame.

Figure 4: The volleyball moving by (18mm,0mm,9mm), and no rotation motion.

Figure 5: The toy robot moving by (18mm,0mm,9mm), and no rotation motion.