Shape and Motion from Linear Features*

Warren F. Gardner and **Daryl T. Lawton**
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280

Abstract

This paper introduces a technique for extracting structure and motion using directionally selective matches between linear features. A world-centered coordinate system is used to make these computations without the intermediate calculation of depth. In order to constrain the possible structure and motion configurations, we assume that the three-dimensional direction of gravity relative to each image frame is known. The direction of gravity, along with the directionally selective linear feature matches, form a set of quadratic equations which can be used to determine structure and motion.

1 Introduction

The extraction of environmental structure and motion from a sequence of two-dimensional images is a common problem in computer vision. Typically solutions to this problem are expressed in camera-centered coordinate systems where environmental geometry is specified by the depth along an image feature's ray of projection. Unfortunately, parameters computed from this camera-centered representation are dependent upon the depth to environmental features. This leads to erroneous results for objects located far from the camera.

The recently introduced *factorization method* [Tomasi and Kanade, 1990; Tomasi and Kanade, 1992; Boult and Brown, 1992] has attempted to overcome the disadvantages associated with a camera-centered representation. This method uses a world-centered coordinate system, along with an orthogonal projection assumption, in order to compute shape and motion without the intermediate calculation of depth. A matrix of image measurements is constructed by making point correspondences between image frames. The matrix is then factored into a shape matrix and a motion matrix using Singular Value Decomposition.

One problem with the factorization method is that it relies upon accurate point correspondences between image frames. This paper introduces a method of extracting shape and motion from directionally selective linear feature correspondences. This line-based algorithm is capable of reconstructing shape and motion without computing depth as an intermediate step. In addition to the orthogonality assumption, we assume that the three-dimensional direction of gravity is known relative to each image in a motion sequence.

The algorithm begins by searching for the orientation of one of the lines in the environment. This is a one dimensional search over 180^o, constrained by the projection of the line on one of the image planes. Each candidate line orientation, along with the position of gravity, forms a set of quadratic equations which constrain all the other lines, as well as the rotation between image frames. An error measure is computed from the derived line orientations and used to evaluate each shape and motion configuration. Once the line orientations and parameters of rotation have been derived, the relative positions of the lines can also be computed from simple linear equations.

The remainder of this section introduces the notation used throughout this paper. Section 2 shows how to derive line orientation and camera rotation from a sequence of two-dimensional images. Section 3 presents a set of linear equations which can be used to solve for the relative line positions. The algorithms presented in the paper are applied to synthetic data and the results are presented in Section 4. Finally, concluding remarks are given in Section 5.

1.1 Notation

The notation used throughout this paper is shown in Figure 1. An image frame at time f is delineated by unit vectors i_f, j_f, and k_f. A three-dimensional environmental line is represented by a unit vector d_s specifying the line direction, and a point on the line p_s. Line (d_s, p_s) is projected orthographically onto image frame f. The direction of the projected line is represented by its unit normal $\tilde{n}_{fs}$. $\tilde{p}_{fs}$ refers to the projection of p_s. The direction of gravity will be referred to as g_f. The two-dimensional parameters $\tilde{n}_{fs}$ and $\tilde{p}_{fs}$ as well as the three-dimensional parameter g_f are all expressed in the coordinate system of image frame f. All other parameters are specified relative to the world coordinate system. When $\tilde{n}_{fs}$ is specified in the world coordinate system it

*This research is supported by the Advanced Research Projects Agency of the Department of Defense and is monitored by the U. S. Army Topographic Engineering Center under contract No. DACA76-92-C-0016

Figure 2: Normals are determined by intersecting a plane with a circular cone

Since the line normals $\tilde{n}_{f1}$ were formed by orthographic projection, they must be perpendicular to the line d_1. Therefore, one constraint is that the vectors n_{f1} must lie within the plane perpendicular to d_1. An additional constraint is provided by the gravity vector g_f. The angle between $\tilde{n}_{f1}$ and g_f must be the same as the angle between n_{f1} and the direction of gravity in the world coordinate system (g_w). These two constraints can be used to solve for n_{f1}. Figure 2 shows the geometry of these two constraints. Each normal (n_{f1}) is determined by intersecting a plane with a circular cone. The plane is defined by d_1. The cone is constructed by rotating a vector about the direction of gravity at the appropriate angle. Since the origin of the cone lies within the plane, the intersection of the plane with the cone results in two lines. There are only two possible solutions since the normals are known to be unit vectors.

The constraints described above will now be examined in more detail. As stated earlier, the direction of gravity g_f relative to the line normals $\tilde{n}_{f1}$ is known. This results in the following relationship

$$n_{f1} \cdot g_w = \tilde{n}_{f1} \cdot g_f \tag{1}$$

where g_w is the direction of gravity in the world coordinate system. Letting $g_w = (0, -1, 0)$ we can simplify Equation 1

$$n_{f1_y} = -\tilde{n}_{f1} \cdot g_f \tag{2}$$

In addition to the angle constraint we know that n_{f1} lies within the plane defined by d_1. This constraint is expressed as

$$n_{f1} \cdot d_1 = 0 \tag{3}$$

Finally, we know that the magnitude of each normal vector (n_{f1}) equals one

$$\|n_{f1}\| = 1 \tag{4}$$

Equations 2, 3, and 4 can be combined into a single quadratic equation, resulting in two feasible solutions for each normal vector.

2.2 Additional Line Normals

The next step in the extraction of line orientation and rotation is to solve for the position within the world coordinate system of the rest of the line normals. This is

Figure 3: Normals are determined by intersecting two circular cones

accomplished by using the candidate line normals. The idea is essentially the same as in the previous section. Two constraints can be formulated from the given geometry. The first constraint is given by the gravity vector g_f, and is identical to the constraint presented in the previous section. The angle between $\tilde{n}_{fs}$ and g_f must be the same as the angle between n_{fs} and the direction of gravity in the world coordinate system. The second constraint is that the angle between an image normal vector $\tilde{n}_{fs}$ and the candidate image normal vector $\tilde{n}_{f1}$ must be the same as the angle between the associated world coordinate normal vectors n_{fs} and n_{f1}. These two constraints can be used to solve for all the additional normal vectors n_{fs}. The constraints are shown geometrically in Figure 3. The solution for a normal vector n_{fs} is essentially the result of intersecting two circular cones. One cone is the result of rotating a vector about the direction of gravity. The other cone results from rotating a vector about the candidate normal vector n_{f1}. The intersection of two circular cones which share the same origin is two lines. Once again, the normals are known to be unit vectors, resulting in two solutions.

The following equations result from the above analysis. The constraint resulting from the gravity vector g_f is identical to the one presented in Section 2. Therefore, from Equation 2 we can write

$$n_{fs_y} = -\tilde{n}_{fs} \cdot g_f \tag{5}$$

The second constraint relates the line normals n_{fs} to the candidate line normals n_{f1} as follows

$$n_{fs} \cdot n_{f1} = \tilde{n}_{fs} \cdot \tilde{n}_{f1} \tag{6}$$

Finally, we know that the magnitude of each normal vector (n_{fs}) equals one

$$\|n_{fs}\| = 1 \tag{7}$$

Equations 5, 6, and 7 can be combined into a single quadratic equation, resulting in two feasible solutions for each normal vector.

2.3 Parameter Estimation

Once the normal vectors (n_{fs}) have been derived, the process of estimating the line orientations and rotational parameters is trivial. The line orientations (d_s) are easily estimated from their associated normals (n_{fs}) using the following equation

$$d_s \cdot n_{fs} = 0 \tag{8}$$

d_s can be estimated with a minimum of two non-collinear normal vectors. When more vectors are available, d_s can be solved for using a linear least-squares technique. The rotational parameters are also easily obtained from the normal vectors n_{fs}. Three linear equations can be formulated for the three rotational parameters i_f, j_f, and k_f

$$\begin{aligned} i_f \cdot n_{fs} &= \tilde{n}_{fs_x} \\ j_f \cdot n_{fs} &= \tilde{n}_{fs_y} \\ k_f \cdot n_{fs} &= 0 \end{aligned}$$

There are also additional constraints available. One of these constraints is that the vectors must be orthonormal

$$\begin{aligned} i_f &= j_f \times k_f \\ j_f &= k_f \times i_f \\ k_f &= i_f \times j_f \\ \|i_f\| &= \|j_f\| = \|k_f\| = 1 \end{aligned}$$

Additional constraints can be derived from the relationship between the rotational vectors and gravity as was done in Sections 2.1 and 2.2. These constraints are

$$\begin{aligned} i_f \cdot g_w &= g_{f_x} \\ j_f \cdot g_w &= g_{f_y} \\ k_f \cdot g_w &= g_{f_z} \end{aligned}$$

Of course, all of the equations presented above are not independent, and all are not necessary. Currently we use the following subset of equations. Initially k_f is determined using a least squares formulation of

$$k_f \cdot n_{fs} = 0 \tag{9}$$

The technique presented in Section 2.1 is then used to solve for i_f with the following equations

$$i_f \cdot k_f = 0 \tag{10}$$

$$i_f \cdot g_w = g_{f_x} \tag{11}$$

Finally i_f and k_f are used to solve for j_f

$$j_f = k_f \times i_f \tag{12}$$

Equation 8 is used to solve for the line orientations d_s. Equations 9, 10, 11, and 12 are used to solve for the rotational parameters i_f, j_f, and k_f. The following section shows how to use these derived parameters to solve for the relative positions of the line segments, thus completing the spatial reconstruction.

3 Line Position

The final step in the line segment reconstruction is to solve for the line segment positions relative to the world

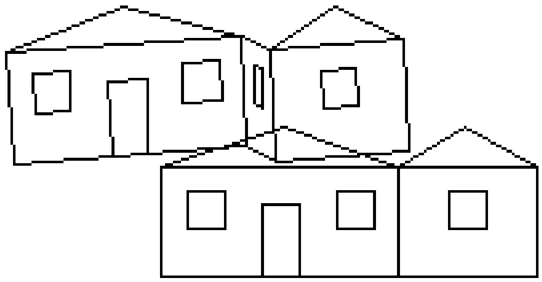

Figure 4: The first and last frames from a 20 image sequence

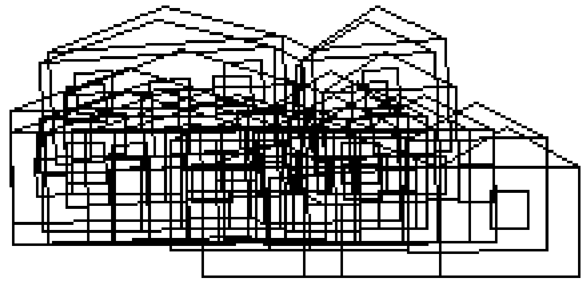

Figure 5: 10 image frames from a 20 image sequence

coordinate system. Initial assumptions about the position of the image frames relative to the world coordinate system are made, allowing a simple linear solution to the problem. The position of each line is represented by a point p_s which is chosen arbitrarily. The world coordinate system will be positioned at the center of image frame 1. The points $\tilde{p}_{1s}$ are then chosen arbitrarily $\tilde{p}_{1s} = (x_s, y_s)$. We assume that all the image planes intersect along line d_1. This means that the position of each image plane is given by $p_1 + \alpha_f d_s$ where α_f is a parametric scale factor.

Each line position $p_s = (x_s, y_s, z_s)$ consists of one unknown z_s. The solution for z_s is trivial. Each point p_s is constrained to lie within the planes perpendicular to n_{fs}. These planes are positioned by choosing some arbitrary point on the projection of each line, and then determining the position of that point within the world coordinate system. Let q be the point in world coordinates

$$q = p_1 + [i_f \; j_f] \cdot (\tilde{p}_{fs} - \tilde{p}_{f1}) \quad (13)$$

The equation of the plane is then written as

$$\begin{aligned} & n_{fs_x}(x_s - q_x - \alpha_f d_{s_x}) + \\ & \quad n_{fs_y}(y_s - q_y - \alpha_f d_{s_y}) + \\ & \quad n_{fs_z}(z_s - q_z - \alpha_f d_{s_z}) = 0 \end{aligned} \quad (14)$$

The two unknowns in this equation are z_s and α_f. α_f can be removed from the equation, and a least squares solution can be found for z_s.

4 Results

The algorithm presented in this paper was implemented and tested on several sequences of synthetic data. The first and last frames from a 20 image sequence are shown in Figure 4. Figure 5 shows 10 frames from the sequence (every other frame is displayed). This data was produced by random rotations and translations. The rotational parameters i_f and j_f associated with this sequence of motion are shown in Figures 6 and 7. The correct rotational values are displayed as solid lines, and the derived values are displayed as dotted lines. All errors are the result of perspective projection. Notice that

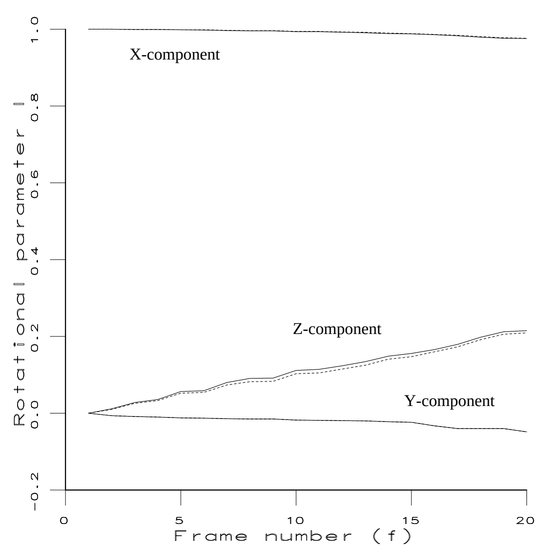

Figure 6: The components of i_f for a 20 frame sequence. The correct values are shown with solid lines, and the derived values are shown with dotted lines.

the Y-component of i_f is errorless. This is because this component is derived from the relationship between the image frames and the gravity vector (g_f) as shown in Equation 11. Thus the Y-component is unaffected by the perspective projection errors.

The derived line orientations and parameters of rotation were then used to reconstruct the line positions as discussed in Section 3. A top view of the original data is shown in Figure 8. The reconstructed data is shown in Figure 9. Once again the errors are the result of perspective projection.

5 Conclusion

The technique presented in this paper is an early attempt at constructing linear feature based depth-independent motion algorithms. The work has only been tested on synthetic data, and it is not clear what effect perspective

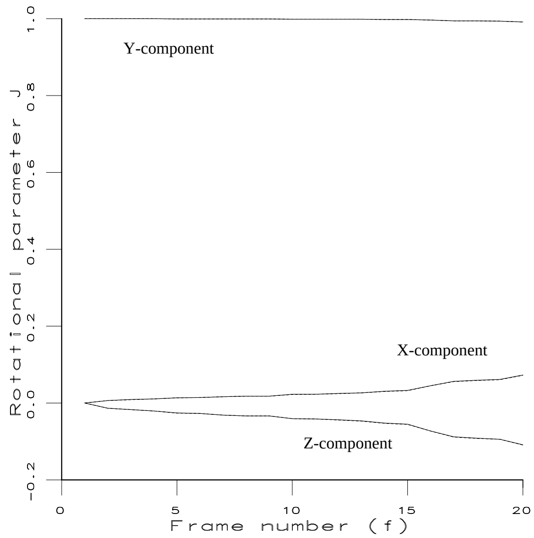

Figure 7: The components of j_f for a 20 frame sequence. The correct values are shown with solid lines, and the derived values are shown with dotted lines.

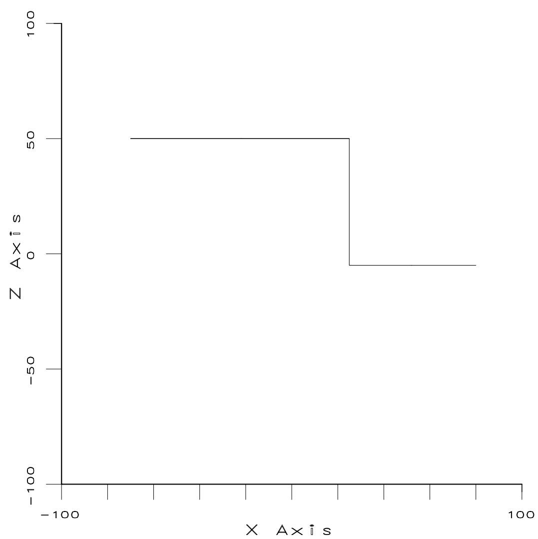

Figure 8: Top view of the house data

projection and other forms of noise will have. However, since the formulation involves linear least squares estimation, it appears that it will be robust. The ability to deal with occlusion is also straight-forward in this overconstrained system. Occluded line normals (n_{fs}) are null vectors and therefore have no effect on the least squares solution. Notice that the first frame shown in Figure 4 contains occluded lines.

One drawback of this method is that the three-dimensional direction of gravity is required. This measurement can be provided by a gravity sensor, but we would like to relax this restriction. One way to remove the gravity vector from the algorithm is to replace the di-

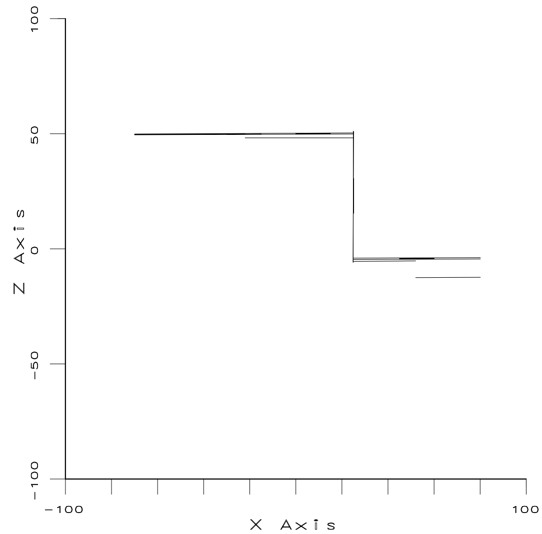

Figure 9: Top view of the reconstructed house

rection of gravity with another consistent direction. For example, for an object that consistently moves in one direction (such as a vehicle), the gravity vector can be replaced by a vector specifying this direction (the forward vehicle direction).

There are several areas for future work:

- Test this algorithm on noisy data and if necessary develop a more robust formulation that will work well in the presence of errors, including the errors introduced from perspective projection.
- Test the algorithm on real image sequences.
- Integrate this rotation based method with the translation based method discussed in [Lawton, 1982]. In this case the gravity vector is replaced by a direction of translation vector. The integration of these two methods will probably be accomplished through temporal filtering using the Kalman Filter.

References

[Boult and Brown, 1992] Terrance E. Boult and Lisa Gottesfeld Brown. Motion segmentation using singular value decomposition. In *Proceedings of the DARPA Image Understanding Workshop*, pages 495–506, 1992. San Diego, CA.

[Lawton, 1982] Daryl T. Lawton. Processing translational motion sequences. *Computer Vision, Graphics, and Image Processing*, 22:116–144, 1982.

[Tomasi and Kanade, 1990] Carlo Tomasi and Takeo Kanade. Shape and motion without depth. In *Proceedings of the DARPA Image Understanding Workshop*, pages 258–270, 1990. Pittsburgh, PA.

[Tomasi and Kanade, 1992] Carlo Tomasi and Takeo Kanade. The factorization method for the recovery of shape and motion from image streams. In *Proceedings of the DARPA Image Understanding Workshop*, pages 459–472, 1992. San Diego, CA.