1988

Time-Varying Images: The Effect of Finite Resolution on Uniqueness

Chia-Hoang Lee

Report Number:
88-801

Lee, Chia-Hoang, "Time-Varying Images: The Effect of Finite Resolution on Uniqueness" (1988).
Department of Computer Science Technical Reports. Paper 683.
<https://docs.lib.purdue.edu/cstech/683>

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries.
Please contact epubs@purdue.edu for additional information.

TIME-VARYING IMAGES: THE EFFECT
OF FINITE RESOLUTION ON UNIQUENESS

Chia-Hoang Lee

CSD-TR-801
August 1988

**Time-Varying Images:
The Effect of Finite Resolution on Uniqueness**

Chia-Hoang Lee
Department of Computer Science
Purdue University
West Lafayette, Indiana 47907

Abstract

The classical paper [1] establishes the result about the uniqueness of motion parameters in image sequences: "Given the image correspondences of eight points in general position, the motion parameters are unique." In this correspondence, we use examples to illustrate that the theorem does not hold if finite resolution is taken into account. It also brings out robustness issue of any possible algorithm. In fact, we show: Given the image correspondence of eight points in two views. If 10% error in the motion parameters is not acceptable, then no robust algorithm uses slant, tilt, or Eulerian angles can be found under the worst case analysis. Furthermore, we suggest rotational matrix should be used to test the robustness of any potential motion algorithm.

The support of the National Science Foundation under grant IRI-8702053 is greatly acknowledged, as is the help of Georgia in the preparation of this paper.

1. Introduction

Analysis of time-varying images is a very important task in such fields as robotic vision and object tracking. Despite great advances in this research area, practical implementations is still far from reality and remains elusive. Thus the timely and urgent task is to implement or to search for correct and robust algorithms for motion analysis. To date not only few literatures address this question, but also there is no general understanding as to why the developed methods are not robust.

In general, approaches employed in time-varying image analysis can be grouped into featured-based and flow-based methods. In the feature-based method each frame of the sequence is segmented first, and the feature points are marked. Next, the correspondence of these featured points between the two frames is established. Lastly, the motion parameters and object structure are derived. The second step is often called the correspondence problem, and the third step is called the structure from motion problem. The discussion throughout this paper is related to the structure from motion problem.

Two different computational schemes can be found among existing analyses for the structure from motion problem. For instance, [2,3,4] rely on the solution of nonlinear equations using iterative searches. Other methods like [1,5] rely on the solution of linear equations and the singular value decomposition of a 3×3 matrix. In solving nonlinear equations iteratively, the search is enormous unless a good initial guess is given. [2,3,4] give neither the details for the implementation of their algorithms nor the experimental results clearly.

On the other hand, [1] which relies on solving linear equations, gives a clear report on experimental simulations aside from theoretical analysis. However, the results suggest the difficulties of this technique to become a robust algorithm because of its sensitivity to the data. In addition to experimental results, [1] also addresses a condition for having unique recovery of motion parameters for time-varying image analysis. They state: *Given seven or more image point correspondences in two views, the motion parameters are uniquely determined if the seven object points do not lie on two planes with one plane passing through the origin or on a cone containing the origin.* Longuet-Higgins [6] enumerates the configurations that defeat the 8-point algorithm (i.e., cases in which the motion parameters are not unique).

The purpose of this paper is twofold: (i) The theory in [1] does not consider the possible effect of finite resolution in the digital image. In fact, the finite resolution requirement is unavoidable for any practical application. We will use one example (more could be created) to illustrate that the uniqueness theorem does not hold if finite resolution is taken into account. (ii) We will use the same example to address the robustness property of any potential motion algorithm and to reveal one reason why the experiments in [1] are so sensitive to noise.

2. Imaging Geometry and the Problem

In this section we will discuss parameters of imaging geometry, some terminology, the structure from motion problem, and the objective of the task.

Camera Parameters:

A pin-hole model instead of an actual camera will be used. The purpose is to avoid calibration procedure and issues of focusing. The pin-hole is assumed at the origin of the $x - y - z$ coordinate system and z -axis is along the optical axis. The image plane is at $z = 1$ and perpendicular to z -axis. The field of view of this pin-hole model is 60° . Figure 1 sketches the imaging geometry.

It is straightforward to deduce from above parameters that the image plane has dimension $\frac{2}{\sqrt{3}} \times \frac{2}{\sqrt{3}}$ square. This image plane will be sampled into 512×512 screen pixels. Thus the spatial resolution is: $1 \text{ pixel} = \frac{1}{256\sqrt{3}}$.

Figure 2 depicts portion of the image plane and shows how sampling is performed. Each square represents a pixel position and described by a coordinate of two integers. The origin of the coordinate system is registered to the center of the pixel (0,0), and the axes are aligned with the grid orientations. In other words, any point (floating representation) lying inside a pixel square is considered to be the same pixel. As an illustration, any image point (X, Y) where

$$-\frac{1}{512\sqrt{3}} \leq X \leq \frac{1}{512\sqrt{3}} \quad \text{and} \quad -\frac{1}{512\sqrt{3}} \leq Y \leq \frac{1}{512\sqrt{3}}$$

will correspond to the same pixel position (0, 0). In case one needs floating number representation for a pixel to perform computations, the center of the pixel will be used. Notice Figure 1 also illustrates two object point with the same pixel position on the screen.

The following algorithm converts a floating number to its pixel coordinate. We assume the conversion from a floating number to integer number is performed by truncation.

Algorithm: (Floating number to screen coordinate)

```
float  X
int    SX (* Screen coordinates *)
if     X ≥ 0 then
      SX = X * 256√3 + 0.5
else
      SX = X * 256√3 - 0.5
```

Motion Problem:

Consider a particular point P on an object. Let

(x, y, z) = object-space coordinates of a point P before motion

(x', y', z') = object-space coordinates of P after motion

$(X, Y, 1)$ = image-space coordinates of P before motion

$(X', Y', 1)$ = image-space coordinates of P after motion

This mapping $(X, Y, 1) \rightarrow (X', Y', 1)$ for a particular point is called an image point correspondence.

It is well known that any 3-D rigid body motion is equivalent to a rotation by an angle θ around an axis through the origin with directional cosines (n_1, n_2, n_3) followed by a translation $T = (t_x, t_y, t_z)^t$,

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R \begin{bmatrix} x \\ y \\ z \end{bmatrix} + T \quad (1)$$

where R is a 3×3 orthogonal matrix.

$$R = \begin{bmatrix} n_1^2 + (1 - n_1^2)\cos\theta & n_1 n_2 (1 - \cos\theta) - n_3 \sin\theta & n_1 n_3 (1 - \cos\theta) + n_2 \sin\theta \\ n_1 n_2 (1 - \cos\theta) + n_3 \sin\theta & n_2^2 + (1 - n_2^2)\cos\theta & n_2 n_3 (1 - \cos\theta) - n_1 \sin\theta \\ n_1 n_3 (1 - \cos\theta) - n_2 \sin\theta & n_2 n_3 (1 - \cos\theta) + n_1 \sin\theta & n_3^2 + (1 - n_3^2)\cos\theta \end{bmatrix}$$

From (1), we could rewrite it into

$$z' \begin{bmatrix} X' \\ Y' \\ 1 \end{bmatrix} = z R \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} + T \quad (2)$$

where

$$X' = \frac{x'}{z'} ; Y' = \frac{y'}{z'} ; X = \frac{x}{z} ; Y = \frac{y}{z}$$

Note that all these numbers are floating numbers so far. We will call an image with floating point coordinates as digital picture and call an image with integer (pixel)

coordinates as digital image. To obtain their screen coordinates, one has to convert (X, Y) and (X', Y') to integers as described in the algorithm above.

Let $(SX_i, SY_i, 1)$ and $(SX'_i, SY'_i, 1)$ be the screen coordinates of $(X_i, Y_i, 1)$ and $(X'_i, Y'_i, 1)$, respectively. Now given N image point correspondences

$$(SX_i, SY_i, 1) \longleftrightarrow (SX'_i, SY'_i, 1); \quad i = 1, \dots, N$$

determine R, T and (X_i, Y_i, Z_i) , $i = 1, 2, \dots, N$. Note that all the existing literatures do not distinguish $(SX_i, SY_i, 1)$ from $(X_i, Y_i, 1)$.

Motion Parameters:

The motion parameters described above consist of θ : rotational angle, (n_1, n_2, n_3) : directional cosines of the rotational axis, and (t_x, t_y, t_z) : translational vector. The rotational axis may also be described in terms of slant and tilt. Slant, ranging from zero to 90° , is the angle between rotational axis and optical axis. Tilt, ranging from zero to 360° , is the angle between the horizontal axis (x -axis) and the projection of rotational axis on the image plane. Since directional cosines and (tilt, slant) are used in the literatures, we will include both of them, denoted by RA and RB, as rotational parameters. RA will denote $(\theta, \tau, \sigma) = (\text{rotational angle, tilt, slant})$ while RB will denote $(\theta, \alpha, \beta, \gamma) = (\text{rotational angle, } \text{acos}(n_1), \text{acos}(n_2), \text{acos}(n_3))$.

As for the translational vector (t_x, t_y, t_z) , we will normalize it so that $t_z = 1$ if $t_z \neq 0$. This can be done since the solution to (2) is up to a scalar.

Objective:

Our task is (i) to investigate the effect of finite resolution on uniqueness of motion parameters, (ii) to bring out the robustness issue of any possible motion algorithm.

3. Example

A rotation with parameters $RA = (10^\circ, 10^\circ, 10^\circ)$ is applied to the following eight points a_1 through a_8 followed by a translation $(2, 2, 8)$. The position of a_i after the motion is denoted by b_i . The screen coordinates for a_i, b_i are listed besides and these serve as observable inputs. Figure 3 depicts the two input images.

	Screen Coordinates
$a_1 = (0.030770 \ 1.323791 \ 2.539637)$	$\rightarrow (5, 231)$
$a_2 = (-0.146403 \ 0.151709 \ 2.506726)$	$\rightarrow (-26, 27)$
$a_3 = (0.962006 \ 0.344183 \ 2.503714)$	$\rightarrow (170, 61)$
$a_4 = (0.289919 \ 0.577317 \ 3.516234)$	$\rightarrow (37, 73)$
$a_5 = (0.371317 \ 0.841253 \ 3.648379)$	$\rightarrow (45, 102)$

$$\begin{aligned}
 a_6 &= (-0.279064 \ -0.532165 \ 3.363153) \rightarrow (-37, -70) \\
 a_7 &= (-0.684855 \ 1.065062 \ 2.787762) \rightarrow (-109, 169) \\
 a_8 &= (0.052545 \ -0.240079 \ 3.994399) \rightarrow (6, -27) \\
 \\
 b_1 &= (0.073834 \ 1.484693 \ 3.578298) \rightarrow (9, 184) \\
 b_2 &= (0.099363 \ 0.301052 \ 3.510543) \rightarrow (13, 38) \\
 b_3 &= (1.158502 \ 0.680327 \ 3.510367) \rightarrow (146, 86) \\
 b_4 &= (0.464369 \ 0.765326 \ 4.531251) \rightarrow (45, 75) \\
 b_5 &= (0.500482 \ 1.035318 \ 4.671074) \rightarrow (48, 98) \\
 b_6 &= (0.092230 \ -0.420184 \ 4.346317) \rightarrow (9, -43) \\
 b_7 &= (-0.585077 \ 1.100200 \ 3.820426) \rightarrow (-68, 128) \\
 b_8 &= (0.373941 \ -0.094259 \ 4.985191) \rightarrow (33, -8)
 \end{aligned}$$

The numbers in $a_1 \sim a_8$ and $b_1 \sim b_8$ presented above has been divided by 8 (for the purpose of comparison) from the original data because the last component of translational vector is 8. It is straightforward to check that any four of these eight points are noncoplanar. Thus it satisfies the assumption in the uniqueness theorem of [1].

The following two sets of solutions consisting of z_i and motion parameters, aside from the one we actually used, can also interpret the 8-image correspondences.

(I) Let the following z_i be the depth of a_i :

$$\begin{array}{ll}
 z_1 = 2.561739 & z_5 = 3.379525 \\
 z_2 = 2.512362 & z_6 = 3.369261 \\
 z_3 = 2.392679 & z_7 = 2.790867 \\
 z_4 = 3.541643 & z_8 = 3.893868
 \end{array}$$

$$RA = (10.024160, -10.877623, 9.363212)$$

$$RB = (10, 100, 91, 10)$$

$$\text{Translation} = (0.281830, 0.246854, 1)$$

$$R = \begin{bmatrix} 0.985125 & -0.171818 & -0.002938 \\ 0.171669 & 0.984749 & -0.028272 \\ 0.007750 & -0.028272 & 0.9995956 \end{bmatrix}$$

(II) Let the following z_i be the depth of a_i :

$$\begin{array}{ll} z_1 = 2.582580 & z_5 = 3.173150 \\ z_2 = 2.632322 & z_6 = 3.320907 \\ z_3 = 2.722661 & z_7 = 2.788329 \\ z_4 = 3.308908 & z_8 = 3.690951 \end{array}$$

$$RA = (10.870609, -35.339671, 17.536793)$$

$$RB = (10, 124, 96, 35)$$

$$\text{Translation} = (0.362555, 0.311039, 1)$$

$$R = \begin{bmatrix} 0.983140 & -0.180595 & -0.0028663 \\ 0.179057 & 0.982601 & -0.049337 \\ 0.037075 & 0.043373 & 0.998371 \end{bmatrix}$$

The meaning of the solutions listed above involve the following steps. (1) Use screen coordinates of a_i as input, (2) Convert screen coordinates into floating representations, (3) Take z_i as the depth of a_i , (4) Apply the rotation parameters and translation vector to the object points constructed from (2) and (3) to obtain space coordinates of objects after motion, (5) Take the projection of these new coordinates in (4) and convert them into screen coordinates, (6) These screen coordinates in (5) actually coincide with the screen coordinate of the second input image i.e. b_i .

4. Counterexamples or Not

The above example, in fact, does not violate the uniqueness theorem in [1]. In the case of finite resolution, points within 0.5 pixel of the center of the picture element (pixel) are regarded as coincident. Conceptually, one could create many pairs of input images (in terms of infinite resolution) having the same two finite resolution input images. Thereafter one could recover motion parameters for each pair of input images and presumably anticipate many solutions. However, owing to the sensitivity of the existing algorithms, it is not a straightforward task to create the above examples. In fact, these examples are part of our efforts to investigate the robustness of motion algorithms.

The above example actually raises the issue of robustness of any potential motion algorithm. The error in tilt might be as large as 400%, and the error in slant might be 70%. However, the use of relative error is quite misleading because small angles will inevitably cause large relative errors. If absolute error is used, then the error in tilt might 40° where the range of tilt is 360°, thus a 10% error. The example shows error from 0% to 10%. Since these solutions are all accurate with regard to finite resolution of the image, there is no basis to favor one over the other. If a 10% error in the motion

parameters is not acceptable, then no robust algorithm uses slant, tilt or Eulerian angles can be found. This also points out the difficulties encountered in [1] and any existing algorithm.

5. Robustness Issue

The sensitivity behavior exhibited by the above example is a worst case analysis of any potential algorithm for minimal input. By minimal input, we mean the number of available feature points is eight. If more points are available, the sensitivity in general would be attenuated. However it is also clear that the more number of points a technique requires, the less application it has. Recently, Barron et al [8] propose an approach to noise sensitivity analysis. The basic idea is to examine error amplification factors: given a certain size input error what is the size of the output error? This approach would be also suited for the analysis of Huang and Tsai's algorithm. See [8] for the details. The computational aspect of the Huang and Tsai's algorithm involves two major steps. The first step involves solving a linear system of eight equations. The second step involves the singular value decomposition of a matrix. It is well known that the sensitivity of either of these two steps depends on the condition of a matrix. Therefore, the behavior of the eight-point algorithm depends on the data and can be predicted. The above example above, however, does not exclude the possible existence of a scheme which could compute motion parameters reliably in some other metric system. For example, [8] suggest that an average case error analysis rather than worst case analysis would be a more appropriate type of analysis. In fact, the following observation suggests measurements in terms of l_2 norm might be a good and useful criterion.

If one examines the rotation matrix then there is a closeness between the actual matrix

$$R = \begin{bmatrix} 0.98525 & -0.17093 & 0.00779 \\ 0.17108 & 0.98482 & -0.02924 \\ -0.00267 & 0.03014 & 0.99954 \end{bmatrix}$$

and the derived matrix,

In fact, if we evaluate $||\hat{R} - R||$ in l_2 norm, we find they are very close to each other. Note that the following holds for any x :

$$\frac{||\hat{R}x - Rx||}{||x||} \leq ||\hat{R} - R||$$

The geometric meaning is that the angle between Rx and $\hat{R}x$ (for every x) cannot exceed $2 \sin^{-1} \left(\frac{||R - \hat{R}||}{2} \right)$. This fact can be seen in Figure 4 where $||R - \hat{R}||$ is the length defined by Rx and $\hat{R}x$. Compute $||\hat{R} - R||$, we obtain 0.022 and 0.064 respectively. This means that the angle between Rx and $\hat{R}x$ (for all x) cannot exceed 2° in one case and 6° in another case.

To further illustrate the difficulty of achieving the robustness in the worst case analysis, two more examples with different sets of motion parameters are provided. The set of motion parameters thus covers a wide spectrum in our examples. The basic configuration of the eight-point object at the first time frame are the same as that in the example of section 3.

Example A: The motion parameter are 10 ° of tilt, 30 ° of slant, and 20 ° of rotational angle.

$A_1 = (5, 231)$	$B_1 = (4, 129)$	
$A_2 = (-26, 27)$	$B_2 = (26, -8)$	
$A_3 = (170, 61)$	$B_3 = (152, 57)$	
$A_4 = (37, 73)$	$B_4 = (54, 29)$	$\rightarrow (55, 28)$
$A_5 = (45, 102)$	$B_5 = (52, 52)$	$\rightarrow (54, 50)$
$A_6 = (-37, -70)$	$B_6 = (46, -87)$	
$A_7 = (-109, 169)$	$B_7 = (-63, 67)$	$\rightarrow (-63, 66)$
$A_8 = (6, -27)$	$B_8 = (55, -56)$	

The above A_i 's and B_i 's represent the two input images and B_i 's are perturbed by one or two pixels as listed to their right hand side, then the following solution would be observed.

$$RA = (22.45, -12.34, 40.25)$$

$$\text{Translation} = (0.462, 0.5060, 1)$$

The error in slant is about 10% and the error in tilt is about 5%.

Example B: The motion parameters are 20 ° of tilt, 30 ° of slant, and 40 ° of rotational angle.

$A_1 = (5, 231)$	$B_1 = (10, 71)$	$\rightarrow (11, 70)$
$A_2 = (-26, 27)$	$B_2 = (73, -50)$	
$A_3 = (170, 61)$	$B_3 = (175, 51)$	$\rightarrow (176, 51)$
$A_4 = (37, 73)$	$B_4 = (88, -8)$	$\rightarrow (89, -8)$
$A_5 = (45, 102)$	$B_5 = (80, 13)$	$\rightarrow (81, 13)$
$A_6 = (-37, -70)$	$B_6 = (120, -126)$	$\rightarrow (122, -126)$
$A_7 = (-109, 169)$	$B_7 = (-36, -6)$	$\rightarrow (-38, -6)$
$A_8 = (6, -27)$	$B_8 = (117, -92)$	

The above A_i 's and B_i 's represent the two input images. Furthermore, B_i 's in the second image are perturbed by one or two pixels as listed to their right hand side, then the following solution would be observed.

$$RA = (43.75, 1.419, 41.04)$$

$$\text{Translation} = (0.38, 0.78, 1)$$

It is clear that the error in slant is about 10% and the error in tilt is about 20°, thus a 5% error out of the range of 360°. The error in translation would be quite unacceptable.

6. Concluding Remarks

From the viewpoints of sampling, there is a 0.5 pixels tolerance for every screen coordinate. With this tolerance, we are able to find three different solutions which clearly demonstrate the effect of finite resolution on uniqueness of motion parameters. From these solutions, we see that the robustness of an algorithm strongly depends on the criterion used. This reveals one source of the difficulties to obtain small-error solution encountered in [1]. In fact, we show that it is not possible to find a robust algorithm if 10% error is not acceptable and angles are used as output under the worst case analysis. Furthermore, we suggest rotational matrix instead of angles should be used to test the robustness of any motion algorithm. However, this does not mean a motion algorithm will then become a robust one if the rotational matrix is used as output. The challenge of searching for robust algorithm remains.

7. References

1. R.Y. Tsai and T.S. Huang, "Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces", *IEEE Trans. Pattern Anal. Machine Intelligence*, Vol PAMI-6, No. 1, Jan. 1984.
2. J.W. Roach and J.K. Aggarwal, "Determining the Movement of Objects from a Sequence of Images", *IEEE Trans. Pattern Anal. Machine Intelligence*, Vol. PAMI-2, Nov. 1979.
3. S. Ullman, *The Interpretation of Visual Motion*, Cambridge, MA, MIT Press, 1979.
4. H.H. Hangel and B. Neumann, "On 3-D Reconstruction from Two Perspective Views", in *Proc. IJCAI 81*, Vol. II, Aug. 1981.
5. H.C. Longuet-Higgins, "A Computer Algorithm for Reconstructing a Scene from Two Projections", *Nature*, 293, 1981.
6. H.C. Longuet-Higgins, "The Reconstruction of a Scene from Two Projections – Configurations that Defeat the 8-Point Algorithm", *Proc. of the First Conf. on Artificial Intelligence Applications*, Dec. 1984
7. Strang, Gilbert, *Linear Algebra and Its Applications 1980*, Academic Press, p. 228, p. 304 New York.
8. J. Barron et al, *The Feasibility of Motion and Structure Computations 1988*, *Second International Conference on Computer Vision*, pp 651- 657.

Figure 1.

Figure 2. Sampling

Figure 3. Squares represent the first image while dots represent the second image

Figure 4