infrastructures

MDPI

Article

Tracking the 6-DOF Flight Trajectory of Windborne Debris Using Stereophotogrammetry

Chaman Lal Sabharwal [1,*] and Yanlin Guo [2]

[1] Computer Science Department, Missouri University of Science and Technology, Rolla, MO 65409, USA
[2] Department of Civil & Environmental Engineering, Colorado State University, Fort Collins, CO 80523-1372, USA; yanlin.guo@colostate.edu
* Correspondence: chaman@mst.edu

Received: 16 September 2019; Accepted: 22 October 2019; Published: 24 October 2019

Abstract: Numerous post-windstorm investigations have reported that windborne debris can cause costly damage to the envelope of buildings in urban areas under strong winds (e.g., during hurricanes or tornados). Thus, understanding the physics of debris flight is of critical importance. Previously developed numerical models describing debris flight physics have not been validated for the complex urban flow environment; such a validation requires experimentally measuring the debris flight trajectory in wind tunnel tests. In this context, this paper proposes a debris measurement algorithm using stereophotogrammetry. This algorithm aims to determine the six-degree-of-freedom (6-DOF) trajectory and velocity of flying debris, addressing the research gap, i.e., the lack of an algorithm/software for measuring three-rotational-DOF using stereophotogrammetry. This is a civil engineering problem, but computer graphics is the foundation to solve it. This paper focuses on the theoretical development of the algorithm. The developed algorithm can be readily implemented in modern wind tunnel experiments.

Keywords: frame; coordinate system; differential; trajectory; motion; debris

1. Introduction

Windborne debris damage to the cladding and façades of buildings has been identified as a major contributor to damage in urban areas after windstorms (e.g., hurricanes, tornados, thunderstorms) in numerous studies [1–11]. To physically model debris-induced damage, one needs to understand debris flight behavior. In recent years, there have been a number of experimental and numerical studies investigating the mechanics of debris flight. Tachikawa [12] pioneered fundamental research on the two-dimensional (2D) trajectory of a flying plate in a uniform flow through both wind tunnel experiments and numerical simulation. Following Tachikawa's work, Lin et al. [13,14] conducted wind tunnel studies to investigate the trajectories of compact-, rod-, and plate-type debris in a uniform wind field and provided experimental support for the validation of numerical simulations using Tachikawa's equations. However, the numerical models developed in the literature have not been validated for the complex 3D urban wind field. Researchers have found that a complex turbulent flow structure in an urban environment can dramatically alter the typical flight distance/speed and lead to unusual debris trajectory, through both experimental and full-scale investigations [15–18]. Thus, experimental validation of the numerical debris flight model for an urban built environment is critical for understanding complex debris flight behavior and predicting the debris impact location, energy/momentum, and ultimately the damage state of the building envelope components. Such a validation first requires the measurement of the debris flight trajectory in a 3D field in wind tunnel tests. In the hardware aspect, the high-speed cameras that have become more accessible in modern wind tunnels can be used for the measurement; however, to the authors' knowledge, there is no off-the-shelf algorithm/software for extracting the entire six-degree-of-freedom (6-DOF) trajectory of flying debris from the images. The existing literature on trajectory measuring of windborne debris using high-speed cameras either only measures the three-translational-DOF (without the three-rotational-DOF) [19] or qualitatively analyzes the different flight modes of the debris instead of resolving the coordinates of the full trajectory [17]. In this context, this paper proposes a debris trajectory measuring algorithm that can reconstruct all 6-DOF displacements and the velocity of debris in a 3D field from pairs of stereoscopic 2D images captured by two high-speed, high-resolution cameras. The concept of using pairs of stereoscopic 2D images to track debris is similar to stereoscopic particle image velocimetry (PIV), which is a non-intrusive laser optical technique for measurement of flow velocity. However, it is distinct from stereoscopic PIV in the following three aspects: (i) the proposed method tracks a single piece of debris, while PIV tracks the patterns of a group of flow particles; (ii) the proposed method does not rely on a laser sheet to illuminate the debris, and thus can track the debris flight with larger lateral motion, while PIV can only track particle motions within the thickness of a laser sheet; and (iii) the proposed method is developed to track the 6-DOF trajectory of debris, while PIV can only measure the 3-DOF motions of flow particles. In wind engineering, three types of debris are typically concerned, i.e., compact, rod, and thin plate. In this paper, we focus on the thin plate. The theoretical development of the three-translational-DOF of compact debris has been included in this discussion as the foundation for the algorithm for thin plates. Furthermore, this algorithm can be extended to consider rod-like debris in the future.

The paper is organized as follows. Section 2 includes the problem formulation and notation. Section 3 focuses on the reconstruction of the 6-DOF coordinates of debris from stereopairs of camera images. Section 4 derives the differential operator for velocity reconstruction. Section 5 introduces procedures for the reconstruction of time histories of debris motion trajectory, including both displacements and velocity. Section 6 includes concluding remarks and a discussion of future directions.

2. Background

Problem Formulation and Notation

In this problem (depicted in Figure 1), first, a background grid is utilized to calibrate the corresponding pixel locations for two high-speed cameras; it can be removed once the calibration is completed. Then the two cameras are used to measure the 2D coordinates of the projection of flying debris on the grid wall by lights emanating from the camera locations. Using the stereopair of 2D coordinates (i.e., d_1, h_1, d_2, h_2 measured by the two cameras) of the debris, we try to determine both the position (three-translational-DOF, x, y, z) and orientation (three-rotational-DOF) of the debris, given the relative locations of the two cameras to the grid wall (i.e., d and l).

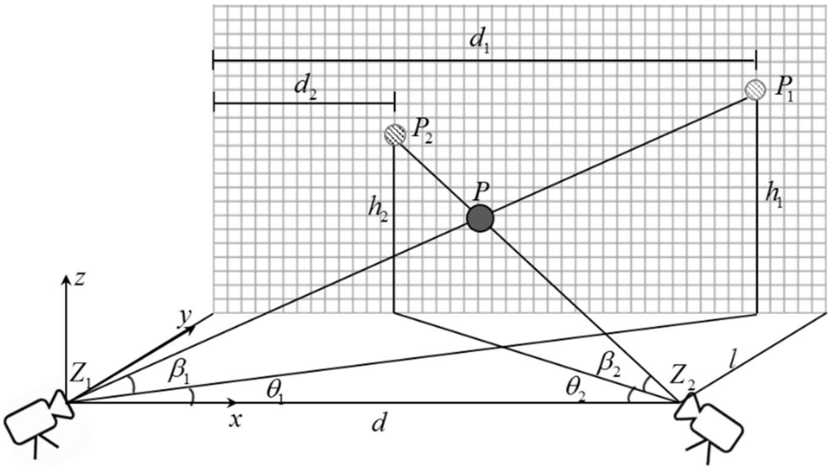

Figure 1. The setup of parameters for the trajectory tracking problem (adapted from [19]).

This problem will be solved in two steps: First, for a single compact piece of debris, considered as a point object (only three-translation-DOF is considered), then for rigid body thin plate debris. The former lays the theoretical foundation for the latter.

The notation used in the problem is defined as follows. The notation of points and matrices are in upper case letters, the position vectors are in bold lowercase letters, and vectors are column vectors by default. For example, $\mathbf{z}_j$ is the position vector of point Z_j, while $\mathbf{p}_j$ is the position vector of point P_j. The list of variables used in Figure 1 is given below.

Z_1, Z_2	–	points at the lens positions of two cameras
P	–	position of a point debris
P_1, P_2	–	projected locations of the debris on the grid wall seen from two cameras (i.e., a stereopair), respectively
d	–	distance between cameras
l	–	distance between camera lens and the wall along the y-direction; the wall is parallel to the z-x plane
d_1, d_2	–	distances of points on the wall along the x-axis from the y-axis
h_1, h_2	–	heights of points on the wall along the z-axis from the x-y plane
β_1, β_2	–	angles that Z_1P_1, Z_2P_2 rays make with the x-y plane
θ_1, θ_2	–	angles between the x-axis and the projections of Z_1P_1, Z_2P_2 rays on the x-y plane

We need to determine the 3D position of the debris, i.e., x, y, and z, in terms of these parameters.

3. Debris Position and Spatial Orientation (6-DOF) in 3D Space

3.1. The 2D Stereopairs' Positions

In Figure 1, Z_1 is the origin and Z_2 is the origin offset by d units along the x-axis: $\mathbf{z}_2 = \mathbf{z}_1 + [d, 0, 0]^T$. The position vector for point P_1 on the wall is

$$\mathbf{p}_1 = [h_1 \cot(\beta_1)\cos(\theta_1), h_1 \cot(\beta_1)\sin(\theta_1), h_1]^T, \tag{1}$$

with a distance from Z_1 of:

$$t_1 = h_1 \cos\mathrm{ec}(\beta_1) = |C_1P_1|. \tag{2}$$

Alternatively, $\mathbf{p}_1 = [d_1, d_1 \tan(\theta_1), h_1]$, $t_1 = \sqrt{{d_1}^2 \sec(\theta_1)^2 + {h_1}^2}$. Similarly, for point P_2 on the wall:

$$\mathbf{p}_2 = [d_1, 0, 0]^T + [-h_2 \cot(\beta_2)\cos(\theta_2), h_2 \cot(\beta_2)\sin(\theta_2), h_2]^T, \tag{3}$$

with a distance from Z_2 of:

$$t_2 = h_2 \cos\mathrm{ec}(\beta_2) = |Z_2P_2|. \tag{4}$$

Alternatively,

$$\begin{aligned} \mathbf{p}_2 &= [d, 0, 0]^T + [-(d - d_2), (d - d_2)\tan(\theta_2), h_2]^T \\ &= [d_2, (d - d_2)\tan(\theta_2), h_2]^T \end{aligned} \tag{5}$$

$$t_2 = \sqrt{(d - d_2)^2 \cos\mathrm{ec}(\theta_2)^2 + h_2^2}. \tag{6}$$

In the simplest terms, the vectors along Z_1P_1 and Z_2P_2 are normalized to unit vectors, $\mathbf{d}_1$ and $\mathbf{d}_2$, where

$$\mathbf{d}_1 = [\cos(\beta_1)\cos(\theta_1), \cos(\beta_1)\sin(\theta_1), \sin(\beta_1)]^T \tag{7}$$

$$\mathbf{d}_2 = [-\cos(\beta_2)\cos(\theta_2), \cos(\beta_2)\sin(\theta_2), \sin(\beta_2)]^T. \tag{8}$$

Then the position vectors for the stereo-pair P_1 and P_2 are

$$\mathbf{p}_1 = \mathbf{z}_1 + t_1\mathbf{d}_1,\ \mathbf{p}_2 = \mathbf{z}_2 + t_2\mathbf{d}_2. \tag{9}$$

This is valid even if the cameras are not at the same distance from the wall.

3.2. The Relationship between the 3D Spatial Position of Debris and the 2D Stereopairs' Positions

Consider compact debris or a point object. The camera beams intersect at the true location of the debris in 3D space with position vector: $\mathbf{z}_1 + s_1\mathbf{d}_1 = \mathbf{z}_2 + s_2\mathbf{d}_2$. This leads to the following equations:

$$s_1\mathbf{d}_1 - s_2\mathbf{d}_2 = \mathbf{z}_2 - \mathbf{z}_1 \tag{10}$$

$$[\mathbf{d}_1, -\mathbf{d}_2]\begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \mathbf{z}_2 - \mathbf{z}_1. \tag{11}$$

This is an overdetermined system of three equations involving two unknowns, s_1 and s_2. Let $M = [\mathbf{d}_1 \ -\mathbf{d}_2]$, which is a 3×2 matrix. Since a 3×2 matrix does not have an inverse, we exploit the generalize inverse, denoted by M^+, to determine the least squares error solution. Now $M^T = \begin{bmatrix} \mathbf{d}_1^T \\ -\mathbf{d}_2^T \end{bmatrix}$ it is a 2×3 matrix, so we create a 2×2 matrix, M^TM:

$$M^TM = \begin{bmatrix} \mathbf{d}_1^T \\ -\mathbf{d}_2^T \end{bmatrix}\left[\mathbf{d}_1^T, -\mathbf{d}_2^T\right] = \begin{bmatrix} \mathbf{d}_1^T\mathbf{d}_1 & -\mathbf{d}_1^T\mathbf{d}_2 \\ -\mathbf{d}_2^T\mathbf{d}_1 & \mathbf{d}_2^T\mathbf{d}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{d}_1 \cdot \mathbf{d}_1 & -\mathbf{d}_1 \cdot \mathbf{d}_2 \\ -\mathbf{d}_1 \cdot \mathbf{d}_2 & \mathbf{d}_2 \cdot \mathbf{d}_2 \end{bmatrix}. \tag{12}$$

Since $\mathbf{d}_1$ and $\mathbf{d}_2$ are unit vectors, let $k = \mathbf{d}_1 \cdot \mathbf{d}_2 = \cos(\theta)$, where θ is the angle between $\mathbf{d}_1$ and $\mathbf{d}_2$. Then,

$$M^TM = \begin{bmatrix} 1 & -k \\ -k & 1 \end{bmatrix}, \ \det(M^TM) = 1 - k^2$$

$$M^+ = (M^TM)^{-1}M^T = \frac{\begin{bmatrix} 1 & k \\ k & 1 \end{bmatrix}}{1-k^2}\begin{bmatrix} \mathbf{d}_1^T \\ -\mathbf{d}_2^T \end{bmatrix}, \ M^+ = \frac{\begin{bmatrix} \mathbf{d}_1^T - k\mathbf{d}_2^T \\ k\mathbf{d}_1^T - \mathbf{d}_2^T \end{bmatrix}}{1-k^2} \tag{13}$$

Then,

$$\begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = [\mathbf{d}_1, -\mathbf{d}_2]^+(\mathbf{z}_2 - \mathbf{z}_1) = \frac{\begin{bmatrix} \mathbf{d}_1^T - k\mathbf{d}_2^T \\ k\mathbf{d}_1^T - \mathbf{d}_2^T \end{bmatrix}}{1-k^2}(\mathbf{z}_2 - \mathbf{z}_1). \tag{14}$$

This form is useful when the camera direction changes and the location is fixed; otherwise, it can be further simplified to

$$\begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \frac{\begin{bmatrix} (\mathbf{d}_1^T - k\mathbf{d}_2^T)(\mathbf{z}_2 - \mathbf{z}_1) \\ (k\mathbf{d}_1^T - \mathbf{d}_2^T)(\mathbf{z}_2 - \mathbf{z}_1) \end{bmatrix}}{1-k^2}. \tag{15}$$

We computed vector forms $\mathbf{z}_1 + t_1\mathbf{d}_1$, $\mathbf{z}_2 + t_2\mathbf{d}_2$ for input points on the wall. Now we have computed $\begin{bmatrix} s_1 \\ s_2 \end{bmatrix}$ for the corresponding object in 3D space. The actual object is at $\mathbf{z}_1 + s_1\mathbf{d}_1 = \mathbf{z}_2 + s_2\mathbf{d}_2$ for these s_1 and s_2. This suggests that the object size at $\mathbf{z}_1 + t_1\mathbf{d}_1$ can be scaled by $\frac{s_1}{t_1}$, with s_1 computed above, along direction $\mathbf{d}_1$ to get the actual size at $\mathbf{z}_1 + s_1\mathbf{d}_1$ in 3D space. Similarly, for camera 2, the object size at $\mathbf{z}_2 + t_2\mathbf{d}_2$ is scaled by $\frac{s_2}{t_2}$, (s_2 computed above), along the direction $\mathbf{d}_{2.}$ The rationale for the scaling of an object at P_1 by a scale factor of $\frac{s_1}{t_1}$ along the line Z_1P_1 is as follows:

$$\mathbf{z}_1 + \frac{s_1}{t_1}(Z_1P_1) = \mathbf{z}_1 + \frac{s_1}{t_1}(\mathbf{p}_1 - \mathbf{z}_1) = \mathbf{z}_1 + \frac{s_1}{t_1}(\mathbf{z}_1 + t_1\mathbf{d}_1 - \mathbf{z}_1) = \mathbf{z}_1 + \frac{s_1}{t_1}(t_1\mathbf{d}_1) = \mathbf{z}_1 + s_1\mathbf{d}_1. \tag{16}$$

Similarly, for camera 2,

$$\mathbf{z}_2 + \frac{s_2}{t_2}(Z_2P_2) = \mathbf{z}_2 + s_2\mathbf{d}_2. \tag{17}$$

3.3. Explicit Expressions of Debris Position in 3D Space

Once s_1 is determined for 3D position, we can compute the desired parameters x, y, and z as in Figure 1. Now $s_1\mathbf{d}_1$ becomes

$$s_1\mathbf{d}_1 = [z\cot(\beta_1)\cos(\theta_1), z\cot(\beta_1)\sin(\theta_1), z]^T. \tag{18}$$

Since $\mathbf{d}_1$ is a unit vector, this implies that

$$s_1 = z\cos\mathrm{ec}(\beta_1),\ z = s_1\sin(\beta_1),\ r_1 = s_1\cos(\beta_1)$$

$$x = r_1\cos(\theta_1) = s_1\cos(\beta_1)\cos(\theta_1),\ y = r_1\sin(\theta_1) = s_1\cos(\beta_1)\sin(\theta_1). \tag{19}$$

Similarly,

$$s_2 = z\cos\mathrm{ec}(\beta_2), z = s_2\sin(\beta_2), r_2 = s_2\cos(\beta_2)$$

$$x = d_1 - r_2\cos(\theta_2) = d_1 - s_2\cos(\beta_2)\cos(\theta_2),\ y = r_2\sin(\theta_2) = s_2\cos(\beta_2)\sin(\theta_2). \tag{20}$$

3.4. Cartesian Position and Orientation for Thin Plate Debris

Now we derive the position and orientation for rigid body thin plate debris, based on the 3D positions of the point objects constructed from the stereo images in the previous section. In Figure 2, let C be the centroid (also called the pivot) of the rigid body plate with vertices Q_1, Q_2, Q_3, Q_4, Q_5, Q_6, Q_7, and Q_8. Assume that the vertices of the debris are identifiable in the stereo images using marker tracking. The 3D positions of the vertices can then be determined using the expressions in the previous section based on the stereo measurements. The problem now becomes using the 3D positions of vertices to determine the orientation (three-rotational-DOF) of the rigid body.

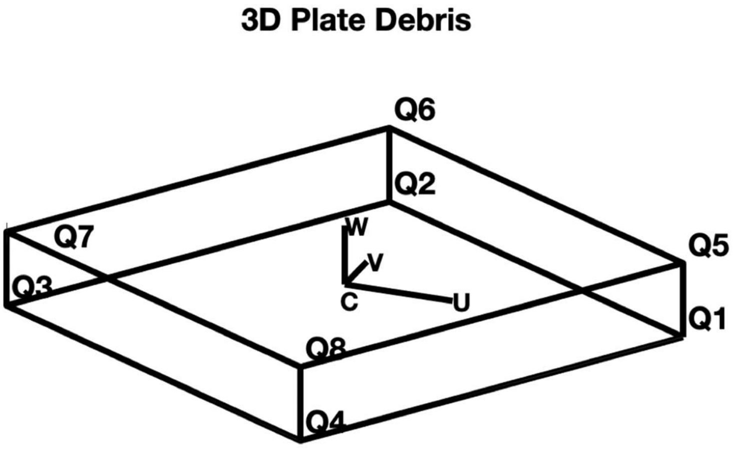

Figure 2. Rectangular plate representing debris.

Firstly, a local coordinate system needs to be defined using any three non-collinear points on the debris. Here we use the points C, Q_1, and Q_2 to create a local coordinate system centered at C and axis directions with unit vectors $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$, where $\mathbf{u}$ is along CQ_1, $\mathbf{v}$ is orthogonal to CQ_1 and in the plane of C, Q_1, and Q_2, with a projection of CQ_2 on $\mathbf{v}$ positive, and $\mathbf{w}$ is naturally $\mathbf{u}\times\mathbf{v}$ so that $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$ form a right-handed system of orthonormal vectors: $\mathbf{u} = \frac{CQ_1}{|CQ_1|}$, $\mathbf{v} = \frac{CQ_2-\mathbf{u}\cdot CQ_2\mathbf{u}}{|CQ_2-\mathbf{u}\cdot CQ_2\mathbf{u}|}$, $\mathbf{w} = \mathbf{u}\times\mathbf{v}$ (equivalently $\mathbf{w} = \frac{\mathbf{u}\times CQ_2}{|\mathbf{u}\times CQ_2|}$, $\mathbf{v} = \mathbf{w}\times\mathbf{u}$).

The matrix $R = \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{w} \end{bmatrix}$ of axis unit vectors is the rotation matrix and is just the 3D local coordinate system at C. For simplicity in the notation for time history of the plate trajectory determined from vertices reconstructed using stereopairs, the 3D plate is represented as a frame $F(t)$ at time t that contains both the position and the orientation for the plate. Since at each time point of the trajectory, there is a frame, the frame sequence will later be denoted by F_k, $k = 1, 2, \ldots, N$. This is similar to frames in a robotic system of N- links.

The matrix $F = \begin{bmatrix} R & \mathbf{c} \\ \mathbf{0} & 1 \end{bmatrix}$ represents the local coordinate system centered at C. The frame can be written interchangeably as

$$F = \begin{bmatrix} R & \mathbf{c} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{w} & \mathbf{c} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} u_1 & v_1 & w_1 & c_1 \\ u_2 & v_2 & w_2 & c_2 \\ u_3 & v_3 & w_3 & c_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{21}$$

This matrix F transforms $\begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ to $\begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{w} & \mathbf{c} \\ 0 & 0 & 0 & 1 \end{bmatrix}$. The *inverse* of F transforms the coordinate system $\begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{w} & \mathbf{c} \\ 0 & 0 & 0 & 1 \end{bmatrix}$, located at the centroid C of the debris, to the universal coordinate system $\begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$. The inverse of F can be written as follows:

$$F^{-1} = \begin{bmatrix} \mathbf{u^T} & -\mathbf{u^T c} \\ \mathbf{v^T} & -\mathbf{v^T c} \\ \mathbf{w^T} & -\mathbf{w^T c} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} = \begin{bmatrix} u_1 & v_1 & w_1 & -\mathbf{u^T c} \\ u_2 & v_2 & w_2 & -\mathbf{v^T c} \\ u_3 & v_3 & w_3 & -\mathbf{w^T c} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} u_1 & v_1 & w_1 & -(u_1c_1 + u_2c_2 + u_3c_3) \\ u_2 & v_2 & w_2 & -(v_1c_1 + v_2c_2 + v_3c_3) \\ u_3 & v_3 & w_3 & -(w_1c_1 + w_2c_2 + w_3c_3) \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{22}$$

3.5. Three-Degree-of-Freedom (3-DOF) Frames in Universal Coordinate System

Here we describe the relationship between the local coordinate system and the axis of rotation and the amount of rotation in a universal coordinate system. The local coordinate system at the centroid of the plate is a frame

$$F = \begin{bmatrix} R & \mathbf{c} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{w} & \mathbf{c} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{23}$$

where R is the rotation matrix about an axis, which is a composite of rotations about the x, y, and z axes. If $\mathbf{n}$ is the unit vector and θ is the angle amount of counter clockwise rotation about $\mathbf{n}$, the axis through the origin, then [20]

$$R = \mathbf{nn}^T + (\mathbf{I} - \mathbf{nn}^T)\cos(\theta) + \mathbf{n} \times \mathbf{I}\sin(\theta). \tag{24}$$

This is directly related to the matrix representation, $R = \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{w} \end{bmatrix}$. The angle θ and vector $\mathbf{n}$ are given by

$$\theta = \cos^{-1}(\frac{u_1 + v_2 + w_3 - 1}{2}) \tag{25}$$

$$\mathbf{n} = \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix} = \begin{bmatrix} v_3 - w_2 \\ w_1 - u_3 \\ u_2 - v_1 \end{bmatrix} / 2\sin(\theta). \tag{26}$$

The rotation vector and amount of rotation about $\mathbf{n}$ is denoted by $\mathbf{n}\theta$, giving the 3-DOF Euler angles, components of rotation about the x, y, and z axes in the universal coordinate system.

3.6. Differential Operators for 6-DOF Motion

To facilitate the determination of the velocity, differential operators for 6-DOF motion with respect to the universal coordinate system are derived in this section. Let $dF = F_{k+1} - F_k$ or explicitly 3-DOF translational and 3-DOF rotational differentials are

$$dF = \begin{bmatrix} dR & d\mathbf{c} \\ \mathbf{0} & 0 \end{bmatrix}, \tag{27}$$

where dR and $d\mathbf{c}$ are differential changes in rotation (R) and translation ($\mathbf{c}$). The differential $d\mathbf{c}$ is a linear differential in the x, y, and z coordinates of $\mathbf{c}$; it is simple and trivial. However, differential change, dR, in the angle of rotation is not straightforward because the differential of the rotation matrix is a matrix operator. Recall that matrix R [21] is

$$R = \mathbf{nn}^T + (\mathbf{I} - \mathbf{nn}^T)\cos(\theta) + \mathbf{n} \times \mathbf{I}\sin(\theta). \tag{28}$$

Differentiating with respect to θ, we have

$$dR/d\theta = -(\mathbf{I} - \mathbf{nn}^T)\sin(\theta) + \mathbf{n} \times \mathbf{I}\cos(\theta). \tag{29}$$

This derivative of the rotation matrix can be expressed as a matrix multiplication, the differential operator [21]. This is accomplished by cross-multiplying Equation (28) by $\mathbf{n}$ as follows:

$$\mathbf{n} \times R = \mathbf{n} \times (\mathbf{nn}^T + (\mathbf{I} - \mathbf{nn}^T)\cos(\theta) + \mathbf{n} \times \mathbf{I}\sin(\theta)). \tag{30}$$

Using vector manipulation identities, this convoluted expression is simplified as follows:

$$\begin{aligned} \mathbf{n} \times R = \mathbf{n} \times \mathbf{nn}^T + \mathbf{n} \times (\mathbf{I} - \mathbf{nn}^T)\cos(\theta) + \mathbf{n} \times (\mathbf{n} \times \mathbf{I})\sin(\theta) = 0 + (\mathbf{n} \times \mathbf{I} - 0)\cos(\theta) + (\mathbf{nn} \cdot \mathbf{I} - \mathbf{In} \cdot \mathbf{n})\sin(\theta) \\ = \mathbf{n} \times \mathbf{I}\cos(\theta) + (\mathbf{nn}^T - \mathbf{I})\sin(\theta) = -(\mathbf{I} - \mathbf{nn}^T)\sin(\theta) + \mathbf{n} \times \mathbf{I}\cos(\theta) \end{aligned} \tag{31}$$

which is identical to $dR/d\theta$, as obtained in Equation (29).

Recalling that the amount of rotation in the universal coordinate system is represented by $\theta\mathbf{n}$, the differential becomes $d\theta\mathbf{n}$, which yields the rotation about the x, y, and z axes. This is consistent with the differential operator derived from the matrix:

$$\begin{gathered} dR/d\theta = \mathbf{n} \times R = \mathbf{n} \times \mathbf{I}R, \\ dR = \mathbf{n} \times Rd\theta = \mathbf{n} \times \mathbf{I}Rd\theta = \mathbf{n} \times \mathbf{I}d\theta R = \mathbf{n}d\theta \times \mathbf{I}R = \Delta R, \end{gathered} \tag{32}$$

where Δ is the differential operator, defined below. Thus, the differential dR of R is a matrix multiplication of R by a matrix, called the differential operator and denoted by Δ.

$$\Delta = \mathbf{n}d\theta \times \mathbf{I} = \begin{bmatrix} 0 & -d\theta n_3 & d\theta n_2 \\ d\theta n_3 & 0 & -d\theta n_1 \\ -d\theta n_2 & d\theta n_1 & 0 \end{bmatrix}, \text{ or simply } = \begin{bmatrix} 0 & -d\theta_z & d\theta_y \\ d\theta_z & 0 & -d\theta_x \\ -d\theta_y & d\theta_x & 0 \end{bmatrix}. \tag{33}$$

Summarizing, if θ is the angle of rotation about $\mathbf{n}$ and $\theta\mathbf{n}$ represents the amount of rotation in the universal coordinate system, then the differential becomes $d\theta\mathbf{n}$, which yields the rotation about the x, y, and z axes; it is consistent with the differential operator derived from the matrix. Here Δ is the differential operator that represents the differential change from R_k to R_{k+1} with respect to the universal coordinate system, where R_k is the rotational submatrix of frame F_k. The rotational part is $\begin{bmatrix} 0 & -d\theta_z & d\theta_y \\ d\theta_z & 0 & -d\theta_x \\ -d\theta_y & d\theta_x & 0 \end{bmatrix}$ with the angular changes about the x, y, and z axes. In short, the rotational differential is $\begin{bmatrix} d\theta_x & d\theta_y & d\theta_z \end{bmatrix}^T$ and the translational differential is $\begin{bmatrix} dx & dy & dz \end{bmatrix}^T$. The composite differential is 6-DOF vector $\begin{bmatrix} dx & dy & dz & d\theta_x & d\theta_y & d\theta_z \end{bmatrix}^T$, representing a composite vector for the translational and rotational velocity about the x, y, and z axes, similar to the notation used in Jacobians for the velocity of robotic link velocity applications.

4. 6-DOF Motion Consideration

4.1. Motivation

It is not necessary to know the initial velocity in this case; only the differential change matters. At any time t, from the position of the plate, we can get the orientation from the frame $F(t)$. Also, during the motion, some features invisible to the camera may become visible, while some that are visible may become invisible. We need to identify at least three vertices, which will be flagged at all times for the sake of calculating the local coordinate system and the rotational motion, in addition to the translational motion.

As the debris moves, frame $F(t)$ at time t moves to the next frame $F(t+h)$ with time step h; the differential change in the frame, denoted by $dF = F(t+h) - F(t)$, can be used to derive the numerical instantaneous translational and rotational velocities. In this case, the debris is flying behind a building; the average velocity can be derived from visible positions (after the time lag of the hidden period) or continuously from the flags identified earlier. In such cases, we may opt to have one continuous trajectory, or multiple segments of trajectory.

4.2. The Relationship between Differential Transformations in Universal and Local Coordinate Systems

4.2.1. General Transformation

As a transformation is accomplished using two coordinate systems, the transformation can be expressed between them in terms of universal and local coordinate systems. In general, if T is the transformation of frame F with respect to the universal coordinate system, this is represented by U, while T' is its counterpart in the local coordinate system. They are related by the equation $T = FT'F^{-1}$. If a point or matrix X has coordinates with respect to frame F, then $TF(X) = F(T'(X))$. Equivalently, if Y has coordinates with respect to the universal frame, then $T(Y) = F\Big(T'\big(F^{-1}(Y)\big)\Big)$. Thus, T and T' accomplish the same task from different coordinate systems, and are related by the equations $T = FT'F^{-1}$ and $T' = F^{-1}TF$.

4.2.2. Rotational Differential Transform with Respect to Universal Coordinate System

We have shown that for rotation matrix R, the rotation differential operator with respect to the universal coordinate system is $\Delta = \mathbf{n}d\theta \times \mathbf{I}$. Let $\boldsymbol{\omega} = \mathbf{n}d\theta$ be called the angular differential, then $\Delta = \boldsymbol{\omega} \times \mathbf{I}$.

4.2.3. Differential Transform in Universal Coordinate System

The translational and rotational velocities are first computed in the universal coordinate system.

Definition 1. *For* $F = \begin{bmatrix} R & \mathbf{c} \\ \mathbf{0} & 1 \end{bmatrix}$ *and the universal differential transformation matrix T, the differential change in frames is defined as follows:*

$$dF = TF = \begin{bmatrix} dR & d\mathbf{c} \\ \mathbf{0} & 0 \end{bmatrix} = \begin{bmatrix} \Delta R & d\mathbf{c} \\ \mathbf{0} & 0 \end{bmatrix}, \tag{34}$$

where $\Delta = \boldsymbol{\omega} \times \mathbf{I}$, $\boldsymbol{\omega} = \mathbf{n}d\theta$, $d\mathbf{c} = \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix}$.

Now we define T as the differential operator with respect to the universal coordinate system. $dF = TF$ implies $T = dFF^{-1}$, then $T = \begin{bmatrix} \Delta & d\mathbf{c} - \Delta\mathbf{c} \\ \mathbf{0} & 0 \end{bmatrix}$.

4.2.4. Differential Transform in Local Coordinate System

For studying the different flight modes of a plate, it is desirable to have the computation of velocities also available in the local coordinate system, and is more conceptually reasonable. Before deriving the differential transformation in the local coordinate system, we will first show a few auxiliary results.

1. Show that the rotation differential operator $\Delta' = R^T \Delta R$ with respect to the local frame system, such that

$$\Delta' = \begin{bmatrix} \boldsymbol{\omega} \cdot \mathbf{u} \\ \boldsymbol{\omega} \cdot \mathbf{v} \\ \boldsymbol{\omega} \cdot \mathbf{w} \end{bmatrix} \times \mathbf{I}. \tag{35}$$

Proof.

$$\Delta' = R^T \Delta R = R^T \boldsymbol{\omega} \times \mathbf{I} R = R^T \boldsymbol{\omega} \times R = R^T [\boldsymbol{\omega} \times \mathbf{u}, \boldsymbol{\omega} \times \mathbf{v}, \boldsymbol{\omega} \times \mathbf{w}] = \left[R^T \boldsymbol{\omega} \times \mathbf{u}, R^T \boldsymbol{\omega} \times \mathbf{v}, R^T \boldsymbol{\omega} \times \mathbf{w} \right]$$

$$= \left[\begin{bmatrix} \boldsymbol{\omega} \times \mathbf{u} \cdot \mathbf{u} \\ \boldsymbol{\omega} \times \mathbf{u} \cdot \mathbf{v} \\ \boldsymbol{\omega} \times \mathbf{u} \cdot \mathbf{w} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\omega} \times \mathbf{v} \cdot \mathbf{u} \\ \boldsymbol{\omega} \times \mathbf{v} \cdot \mathbf{v} \\ \boldsymbol{\omega} \times \mathbf{v} \cdot \mathbf{w} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\omega} \times \mathbf{w} \cdot \mathbf{u} \\ \boldsymbol{\omega} \times \mathbf{w} \cdot \mathbf{v} \\ \boldsymbol{\omega} \times \mathbf{w} \cdot \mathbf{w} \end{bmatrix} \right] = \begin{bmatrix} 0 & -\boldsymbol{\omega} \cdot \mathbf{w} & \boldsymbol{\omega} \cdot \mathbf{v} \\ \boldsymbol{\omega} \cdot \mathbf{w} & 0 & -\boldsymbol{\omega} \cdot \mathbf{u} \\ -\boldsymbol{\omega} \cdot \mathbf{v} & \boldsymbol{\omega} \cdot \mathbf{u} & 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\omega} \cdot \mathbf{u} \\ \boldsymbol{\omega} \cdot \mathbf{v} \\ \boldsymbol{\omega} \cdot \mathbf{w} \end{bmatrix} \times \mathbf{I}.$$

□

2. Show that the linear differential $d\mathbf{c}' = R^T d\mathbf{c}$ in the local frame simplifies to

$$d\mathbf{c}' = \begin{bmatrix} d\mathbf{c} \cdot \mathbf{u} \\ d\mathbf{c} \cdot \mathbf{v} \\ d\mathbf{c} \cdot \mathbf{w} \end{bmatrix}. \tag{36}$$

Proof.

$$d\mathbf{c}' = R^T d\mathbf{c} = \begin{bmatrix} \mathbf{u}^T \\ \mathbf{v}^T \\ \mathbf{w}^T \end{bmatrix}, d\mathbf{c} = \begin{bmatrix} \mathbf{u}^T d\mathbf{c} \\ \mathbf{v}^T d\mathbf{c} \\ \mathbf{w}^T d\mathbf{c} \end{bmatrix} = \begin{bmatrix} \mathbf{u} \cdot d\mathbf{c} \\ \mathbf{v} \cdot d\mathbf{c} \\ \mathbf{w} \cdot d\mathbf{c} \end{bmatrix} = \begin{bmatrix} d\mathbf{c} \cdot \mathbf{u} \\ d\mathbf{c} \cdot \mathbf{v} \\ d\mathbf{c} \cdot \mathbf{w} \end{bmatrix}. \tag{37}$$

□

3. Differential operator with respect to local frame coordinate system.

Theorem 1. *The frame differential transformation matrix T' in the local coordinate system is*

$$T' = \begin{bmatrix} \Delta' & d\mathbf{c}' \\ \mathbf{0} & 0 \end{bmatrix}, \tag{38}$$

$$\textit{where } \Delta' = \begin{bmatrix} \boldsymbol{\omega}\cdot\mathbf{u} \\ \boldsymbol{\omega}\cdot\mathbf{v} \\ \boldsymbol{\omega}\cdot\mathbf{w} \end{bmatrix} \times \mathbf{I},\, d\mathbf{c}' = \begin{bmatrix} d\mathbf{c}\cdot\mathbf{u} \\ d\mathbf{c}\cdot\mathbf{v} \\ d\mathbf{c}\cdot\mathbf{w} \end{bmatrix}.$$

Proof. From $T' = F^{-1}TF$ (Section 4.2.1) and $TF = \begin{bmatrix} \Delta R & d\mathbf{c} \\ \mathbf{0} & 0 \end{bmatrix}$ (Equation (34)), we get

$$T' = \begin{bmatrix} R^T & -R^T\mathbf{c} \\ \mathbf{0} & 1 \end{bmatrix}\begin{bmatrix} \Delta R & d\mathbf{c} \\ \mathbf{0} & 0 \end{bmatrix} = \begin{bmatrix} R^T\Delta R & R^T d\mathbf{c} \\ \mathbf{0} & 0 \end{bmatrix} = \begin{bmatrix} \Delta' & d\mathbf{c}' \\ \mathbf{0} & 0 \end{bmatrix} \\ \Big(\text{recall that } \Delta' = R^T\Delta R \;\text{ and } d\mathbf{c}' = R^T d\mathbf{c}\Big). \tag{39}$$

Hence, the theorem is proved. □

Finally, the 6-DOF local translational and rotational velocities in the local frame are derived from the differential transformation as follows:

$$dF' = T'F = \begin{bmatrix} \Delta' & d\mathbf{c}' \\ \mathbf{0} & 0 \end{bmatrix}\begin{bmatrix} R & \mathbf{c} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \Delta' R & \Delta'\mathbf{c} + d\mathbf{c}' \\ \mathbf{0} & 0 \end{bmatrix}. \tag{40}$$

5. 6-DOF Motion Trajectory with Velocity

5.1. Velocity

We have determined the 6-DOF frame for position and orientation, as well as the differentials for translational and rotational motion of the debris. High-speed, high-resolution cameras are fast enough to provide data on very small intervals of time. Assuming that the time history of 6-DOF frames has been reconstructed according to Section 3, we now determine the numerical derivatives from this sequence of 3D temporal positions and orientations (i.e., displacements) of the debris using the differentials derived in Section 4. This gives the translational and rotational velocities from the numerical time histories of the displacements.

5.2. Experimenting with 6DOF Position and Orientation and 6DOF Motion Methods

For the experiments, the data structure of debris is represented as a thin $m \times n \times q$ plate, where $m \times n$ is a polygonal face shape and q is the thickness of the plate. The vertices of the plate are represented by a matrix array $deb = deb(x, y, z)$, where $x = x(1:m), y = y(1:n), z = z(1:q)$. For example, in Figure 2, with a $4 \times 4 \times 2$ simple rectangular plate, we may define $z(1) = 0$, $z(2) = 2$, $x(1:4) = [1, 50, 50, 1]$, $y(1:4) = [1, 1, 50, 50]$; thus, one flat face is $face1 = \{(x(k), y(k), z(1));\ k = 1, 2, 3, 4\}$, and the second face is $face2 = \{(x(k), y(k), z(2));\ k = 1, 2, 3, 4\}$.

5.3. Implementation Procedure for Debris Tracking with Both Displacement and Velocity Time Histories Determined

Conceptual algorithms and implementation go hand in hand. From the stereopairs of images captured by two cameras, we computed the 3D spatial positions of flagged points on the debris. Then, using three non-collinear points (e.g., the centroid and two vertices), we constructed a frame matrix $F = \begin{bmatrix} R & \mathbf{c} \\ \mathbf{0} & 1 \end{bmatrix}$ to represent the position and orientation of the debris. Here $R = \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{w} \end{bmatrix}$ represents the rotation matrix about unit vector $\mathbf{n}$ through the origin by angle θ, where $\theta = \cos^{-1}(\frac{u_1+v_2+w_3-1}{2})$ and $\mathbf{n} = \begin{bmatrix} v_3 - w_2 \\ w_1 - u_3 \\ u_2 - v_1 \end{bmatrix} / 2\sin(\theta)$. F changes its state over time as the debris moves. The motion trajectory is then captured by successive camera frames $F(t)$ to frame $F(t+h)$, where h is the time step, depending on the camera speed. For translational and rotational velocity, we calculated the differential of frame F as $dF = F(t+h) - F(t) = \begin{bmatrix} dR & d\mathbf{c} \\ \mathbf{0} & 1 \end{bmatrix}$, where $d\mathbf{c}$ is the differential of centroids of frame $F(t)$ and subsequent frame $F(t+h)$, and dR is the differential of orientation of frame $F(t)$ and subsequent frame $F(t+h)$. We have seen that $dR = \Delta R$ with $\Delta = \boldsymbol{\omega} \times \mathbf{I}$, $\boldsymbol{\omega} = \mathbf{n} d\theta$, $\mathrm{d}\theta = \cos^{-1}(\frac{u_1+v_2+w_3-1}{2})(t) - \cos^{-1}(\frac{u_1+v_2+w_3-1}{2})(t+h)$.

Now dF can be written in terms of frame differential operator T such that $dF = TF$, where $T = \begin{bmatrix} \Delta & d\mathbf{c} - \Delta\mathbf{c} \\ \mathbf{0} & 0 \end{bmatrix}$.

Now for the trajectory of N frames F_k, $k = 1, 2, \ldots, N$. Let the kth stereopair debris vertices be denoted by $Q_{k1i} = \left({}^1x_i, {}^1y_i, {}^1z_i\right)$, and $Q_{k2i} = \left({}^2x_i, {}^2y_i, {}^2z_i\right)$, $i = 1, 2, \ldots, 8$.

The 3D spatial location corresponding to the points Q_{k1i} and Q_{k2i} is denoted by P_{ki}, whose position vector is $\mathbf{p}_{ki}$.

Let $\mathbf{p}_{ki}$ be the position vectors of vertices P_{ki}, and $\mathbf{c}_k$ be the centroid of vertices P_{ki} for $i = 1, 2, \ldots, 8$.

Create frame F_k with $\mathbf{u}_k$, $\mathbf{v}_k$, $\mathbf{w}_k$, $\mathbf{c}_k$ as follows:

$\mathbf{c}_k = (\sum_{i=1,8} \mathbf{p}_{ki})/8$, $\mathbf{u}_k = \mathbf{p}_{k1} - \mathbf{c}_k$, normalize $\mathbf{u}_k = \frac{u_k}{|u_k|}$, $\mathbf{w}_k = \mathbf{u}_k \times (\mathbf{p}_{k2} - \mathbf{c}_k)$, normalize $\mathbf{w}_k = \frac{w_k}{|w_k|}$

$$\mathbf{v}_k = \mathbf{w}_k \times \mathbf{u}_k, \theta_k = \cos^{-1}(\frac{u_{k1} + v_{k2} + w_{k3} - 1}{2}), \mathbf{n}_k = \begin{bmatrix} v_{k3} - w_{k2} \\ w_{k1} - u_{k3} \\ u_{k2} - v_{k1} \end{bmatrix} / 2\sin(\theta_k)$$

$$d\theta_k = \cos^{-1}(\tfrac{u_{k1}+v_{k2}+w_{k3}-1}{2}) - \cos^{-1}(\tfrac{u_{k+1,1}+v_{k+1,2}+w_{k+1,3}-1}{2}), \Delta_k = \boldsymbol{\omega}_k \times \mathbf{I}, \boldsymbol{\omega}_k = \mathbf{n}_k d\theta_k. \tag{41}$$

This creates a sequence of consecutive frames F_k, $k = 1, 2, \ldots, N$ representing the position and orientation of debris at any point. To derive the translational velocity of frame F_k, it is sufficient to know the linear differential change from $\mathbf{c}_k$ to $\mathbf{c}_{k+1}$, the centroids of the consecutive coordinate frames. The rotational velocity is more complex, as seen earlier. In the universal coordinate system:

The differential translational change is $d\mathbf{c}_k = \mathbf{c}_{k+1} - \mathbf{c}_k$.

The differential rotational change is $dR_k = \Delta_k R_k$, where $\Delta_k = \boldsymbol{\omega}_k \times \mathbf{I}$.

The frame differential change is $\mathrm{d}F_k = T_k F_k = \begin{bmatrix} \Delta_k R_k & d\mathbf{c}_k \\ \mathbf{0} & 0 \end{bmatrix}$,

$$T_k = \begin{bmatrix} \Delta_k & d\mathbf{c}_k - \Delta_k \mathbf{c}_k \\ \mathbf{0} & 0 \end{bmatrix}. \tag{42}$$

With respect to the local coordinate system of frame F_k,

$$T_k{}' = \begin{bmatrix} \Delta'_k & d\mathbf{c}'_k \\ \mathbf{0} & 0 \end{bmatrix}. \tag{43}$$

The differential rotational change is $dR' = \Delta_k{}' R$, where

$$\Delta'_k = \begin{bmatrix} \boldsymbol{\omega}_k \cdot \mathbf{u}_k \\ \boldsymbol{\omega}_k \cdot \mathbf{v}_k \\ \boldsymbol{\omega}_k \cdot \mathbf{w}_k \end{bmatrix} \times \mathbf{I}. \tag{44}$$

The differential translational change is $d\mathbf{c}_k{}'$, where

$$d\mathbf{c}'_k = \begin{bmatrix} (\mathbf{c}_{k+1} - \mathbf{c}_k) \cdot \mathbf{u}_k \\ (\mathbf{c}_{k+1} - \mathbf{c}_k) \cdot \mathbf{v}_k \\ (\mathbf{c}_{k+1} - \mathbf{c}_k) \cdot \mathbf{w}_k \end{bmatrix}. \tag{45}$$

The frame differential change is $dF' = T'_k F_k = \begin{bmatrix} \Delta'_k R_k & d\mathbf{c}'_k + \Delta'_k \mathbf{c}'_k \\ \mathbf{0} & 0 \end{bmatrix}$.

With this we complete the derivation of the formulations for determining the 6-DOF position and orientation along with the 6-DOF translation and rotational velocity of the rigid body debris. The full process of the proposed method for measuring the trajectory of plate-type debris is illustrated in Figure 3.

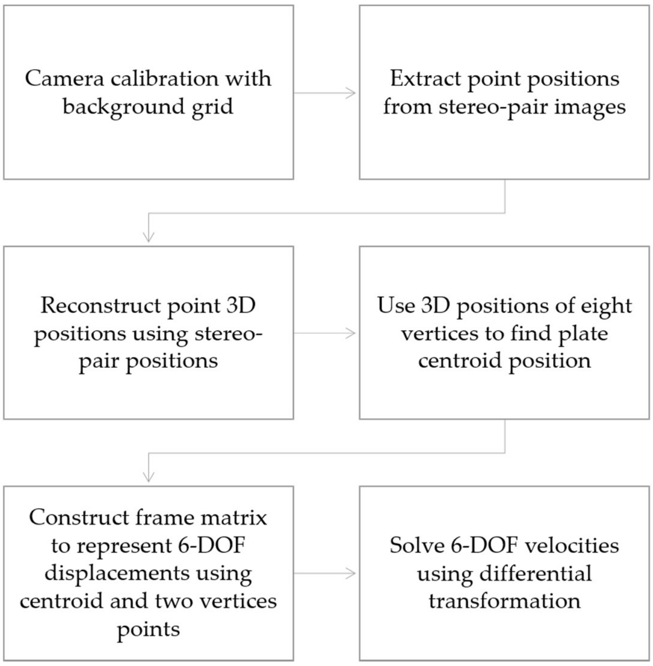
Camera calibration with background grid
Extract point positions from stereo-pair images
Reconstruct point 3D positions using stereo-pair positions
Use 3D positions of eight vertices to find plate centroid position
Construct frame matrix to represent 6-DOF displacements using centroid and two vertices points
Solve 6-DOF velocities using differential transformation

Figure 3. Flowchart of debris-tracking algorithm.

6. Conclusions

In this paper, we propose an algorithm to reconstruct the 6-DOF (three translational and three rotational) trajectory of flying debris using stereopairs of images from two high-speed cameras. Both the time histories of displacements and velocities can be determined by this algorithm. Specifically, this algorithm addresses the research gap in terms of a lack of algorithms/software for measuring three-rotational-DOF using stereophotogrammetry in the literature, and thus contributes to the new measurement science enabled by high-speed cameras and computer graphics theory. It is intended to be used for experimentally measuring the debris trajectory in a 3D wind field to support the investigation of the debris flight physics in an urban built environment and ultimately contributes to enhancing the reliability of building envelopes under windborne debris hazards. However, it may also be applied to reconstructing the 3D motion of people using surveillance cameras in buildings (e.g., to see how a robber is approaching a victim) or of vehicles in car accidents using traffic monitoring cameras. In this paper, we focused on the wind engineering application and developed this algorithm for two common types of debris, i.e., compact and thin-plate-like, in particular. For the former, the three-translational-DOF were reconstructed. Then the results were used to derive the three-rotational-DOF of the thin plate. In future studies, the algorithm can also be extended for rod-like debris, another type of debris commonly observed during wind risk. The current paper dealt with the theoretical development of the algorithm, while its implementation in conjunction with image processing techniques in wind tunnel experiments will be explored in our future studies.

Author Contributions: Conceptualization, C.L.S. and Y.G.; methodology, C.L.S. and Y.G.; writing—original draft preparation, C.L.S. and Y.G.; writing—review and editing, C.L.S. and Y.G.; visualization, C.L.S. and Y.G.

Funding: This research received no external funding.

Acknowledgments: The help from Yue Dong (a graduate student at Colorado State University) with preparing the manuscript was greatly appreciated.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Holmes, J.D. Windborne debris and damage risk models: A review. *Wind Struct.* **2010**, *13*, 95–108. [CrossRef]
2. Minor, J.E. Lessons learned from failures of the building envelope in windstorms. *J. Archit. Eng.* **2005**, *11*, 10–13. [CrossRef]
3. Lee, B.E.; Wills, J. Vulnerability of fully glazed high-rise buildings in tropical cyclones. *J. Archit. Eng.* **2002**, *8*, 42–48. [CrossRef]
4. Behr, R.A.; Minor, J.E. A survey of glazing system behavior in multi-story buildings during hurricane andrew. *Struct. Des. Tall Build.* **1994**, *3*, 143–161. [CrossRef]
5. Sparks, P.R.; Schiff, S.D.; Reinhold, T.A. Wind damage to envelopes of houses and consequent insurance losses. *J. Wind Eng. Ind. Aerodyn.* **1994**, *53*, 145–155. [CrossRef]
6. Dade County. *Building Code Evaluation Task Force Final Report*; Dade County: Miami, FL, USA, 1992.
7. Dade County. *Final Report of the Dade County Grand Jury, Dade County Grand Jury—Spring Term A.D. 1992*; Dade County: Miami, FL, USA, 1992.
8. Federal Emergency Management Agency. *Building Performance: Hurricane Andrew in Florida*; Federal Insurance Administration, FEMA: Washington, DC, USA, 1992.
9. Wind Engineering Research Council (WERC). *Preliminary Observations of WERC Post-Disaster Team*; Wind Engineering Research Council (WERC): College Station, TX, USA, 1992.
10. Oliver, C.; Hanson, C. Failure of residential building envelopes as a result of Hurricane Andrew in Dade County, Florida. In *Hurricanes of 1992: Lessons Learned and Implications for the Future*; Cook, R.A., Soltani, M., Eds.; ASCE: New York, NY, USA, 1994; pp. 496–508.
11. Beason, W.L.; Meyers Gerald, E.; James Ray, W. Hurricane related Window glass damage in Houston. *J. Struct. Eng.* **1984**, *110*, 2843–2857. [CrossRef]
12. Tachikawa, M. Trajectories of flat plates in uniform flow with application to wind-generated missiles. *J. Wind Eng. Ind. Aerodyn.* **1983**, *14*, 443–453. [CrossRef]
13. Lin, N.; Letchford, C.; Holmes, J. Investigation of plate-type windborne debris. Part I. Experiments in wind tunnel and full scale. *J. Wind Eng. Ind. Aerodyn.* **2006**, *94*, 51–76. [CrossRef]
14. Lin, N.; Holmes, J.D.; Letchford, C.W. Trajectories of wind-borne debris in horizontal winds and applications to impact testing. *J. Struct. Eng.* **2007**, *133*, 274–282. [CrossRef]
15. Jain, A. Wind borne debris impact generated damage to cladding of high-rise building. In Proceedings of the 12th Americas Conference on Wind Engineering (12ACWE), Seattle, WA, USA, 16–20 June 2013.
16. Visscher, B.T.; Kopp, G.A. Trajectories of roof sheathing panels under high winds. *J. Wind Eng. Ind. Aerodyn.* **2007**, *95*, 697–713. [CrossRef]
17. Kordi, B.; Kopp, G.A. Effects of initial conditions on the flight of windborne plate debris. *J. Wind Eng. Ind. Aerodyn.* **2011**, *99*, 601–614. [CrossRef]
18. Kordi, B.; Traczuk, G.; Kopp, G.A. Effects of wind direction on the flight trajectories of roof sheathing panels under high winds. *Wind Struct.* **2010**, *13*, 145–167. [CrossRef]
19. Crawford, K. Experimental and Analytical Trajectories of Simplified Debris Models in Tornado Winds. Master's Thesis, Iowa State University, Ames, IA, USA, 2012.
20. Hughes, J.F.; Dam, A.V.; McGuire, M.; Sklar, D.F.; Foley, J.D.; Feiner, S.K.; Akler, K. *Computer Graphics: Principle and Practice*, 3rd ed.; Addison Wesley Professional: Boston, MA, USA, 2013.
21. Niku, S.B. *Introduction to Robotics Analysis, Control, Applications*, 2nd ed.; Wiley: Hoboken, NJ, USA, 2011.