Chapter #

Visual Motion

Motion Field, Optical Flow and Structure from Motion

Zhigang Zhu
Department of Computer Science, The City College of New York, New York, NY 10031

1. INTRODUCTION

This chapter will discuss another important topic in 3D reconstruction: obtaining structure from camera or object motion. We put it in the general area of visual motion, which includes both *structure from motion* and motion recognition. But we are going to focus on the structure from motion problem. The outline of this chapter is as follows:

Section 1. (this section). Problems and Applications
- The importance of visual motion
- Problem Statement

Section 2. The Motion Field of Rigid Motion
- Basics – Notations and Equations
- Three Important Special Cases: Translation, Rotation and Moving Plane
- Motion Parallax

Section 3. Optical Flow
- Optical flow equation and the aperture problem
- Estimating optical flow
- 3D motion & structure from optical flow

Section 4. Feature-based Approach
- Two-frame algorithm – feature matching

- Multi-frame algorithm – SLAM and SfM
- Structure from motion – Factorization method

Section 5. Advanced Topics: Motion-Based Video Computing
- Change detection
- Video mosaicing
- Layered Representation

1.1 The Importance of Visual Motion

Structure from Motion (SfM) using the so-called apparent visual motion is a strong visual clue for 3D reconstruction. It is more than a multi-camera stereo system. As an example, a human vision system can do remarkably well in the recognition of structure only by motion. In fact, biological visual systems can use visual motion to infer properties of the 3D world with little a priori knowledge of it. Figure 1 shows a close-up image of a 15×20-pixel digital frame from a blurred image sequence. If we watch the video sequence, we can recognize the object from motion (a person sitting down) even if we cannot distinguish it in any images of the sequence.

Figure 1.A close-up of an image of resolution 15x20 pixels

From: James W. Davis at MIT Media Lab

In fact visual motion studies typically deal with video sequences, which include the following research and application topics:

- Video Coding and Compression: MPEG 1, 2, 4, 7…

- Video Mosaicing and Layered Representation for IBR
- Surveillance, Human Tracking and Traffic Monitoring
- HCI using Human Gesture
- Image-based Rendering
- And many more…

1.2 Problem Statement

The problem of structure from motion can be divided into the following two sub-problems:

1. **Correspondence**: Which elements in one frame correspond to which elements in the next frame?
2. **Reconstruction**: Given a number of correspondences and possibly, the knowledge of the camera's intrinsic parameters, how to recover the 3-D motion and structure of the observed world

For both of these two sub-problems, there are two or more approaches. For the correspondence problem, differential methods create dense measures (optical flow), whereas matching methods generate sparse measures. The reconstruction problem here is more difficult than in stereo, since we have to estimate motion (i.e., 3D transformation between frames) as well as the structure of 3D objects. Meanwhile, small "baseline" lengths ease the correspondence problem but they also cause large errors in 3D reconstruction.

What are the main differences between motion and stereo?

First, regarding the correspondence problem, disparities between consecutive frames are much smaller due to dense temporal sampling in Structure from Motion (SfM) than in stereo vision.

Second, the visual motion evaluated during reconstruction may, however, be caused by multiple motions (instead of a single 3D rigid transformation in the case of stereo vision).

Furthermore, we may have a third or even fourth sub-problem, namely:

Motion segmentation: what are the regions in the image plane corresponding to different moving objects?

Motion understanding: lip reading, gesture, expression, event understanding, and etc..

The motion segmentation problem is a chicken and egg problem: Which should be solved first - matching or segmentation? We need to do segmentation for extracting matching elements, but we may need to perform matching for scene and motion segmentation.

2. THE MOTION FIELD OF RIGID OBJECTS

In order to fully understand the structure from motion problem, we will need to understand the details of the motion field of rigid objects. This will serve as the basis for structure from motion. A motion in 3D can be characterized by a rotation matrix **R** and a translation vector **T**, and can be caused by the motion of a camera viewing a static scene, or a stationary camera viewing a single object in motion. However, we can always use one rigid, relative motion between the camera and the scene (or object). The *image motion field* is defined as the 2D vector field of velocities of the image points induced by the relative motion.

The input of visual motion is an image sequence that may include several frames, captured at time t=0, 1, 2, … For the basic principle of visual motion, we will only consider two consecutive frames (i.e., a reference frame and its consecutive frame). In this case, the image motion field can be viewed as a disparity map of two frames captured at two consecutive camera locations (assuming we have a moving camera).

2.1 Basic Equations of Motion Field

To formally define the motion field, we need to have the following notations: Let $\mathbf{P} = (X,Y,Z)^T$ represent a 3-D point in the camera reference frame, and $\mathbf{p} = (x,y,f)^T$ the projection of the scene point in the pinhole camera, then we have,

$$\mathbf{p} = \frac{f}{Z}\mathbf{P} \tag{1}$$

The relative motion of **P** in the camera coordinate system is defined as (Figure 2):

$$\mathbf{V} = -\mathbf{T} - \boldsymbol{\omega} \times \mathbf{P} \tag{2}$$

where $\mathbf{T}= (T_x,T_y,T_z)^T$ is the translation component of the motion, and $\boldsymbol{\omega}=(\omega_x, \omega_y, \omega_z)^T$ is the angular velocity, which is further explained below.

Now we would like to answer the following two questions:

1) How can we connect the equation with the stereo geometry using **R** and **T**?
2) How can the image velocity **v** **be** represented in terms of the 3D point **P**?

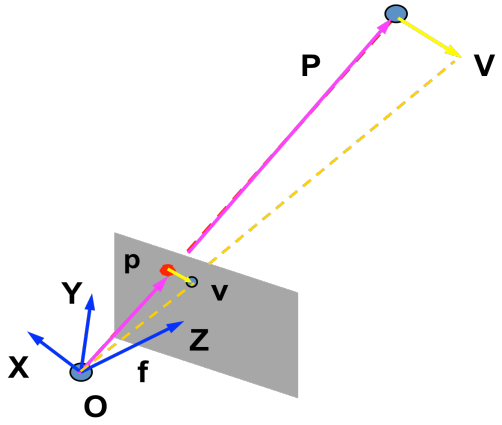

Figure 2. Motion field of a rigid body

Answering the first question will help us to better understand 3D motion. Angular velocity is defined by a rotation axis $\boldsymbol{\omega} / |\boldsymbol{\omega}|$ (a unit vector) and a rotation angle $|\boldsymbol{\omega}|$. So, the cross product $\boldsymbol{\omega} \times \mathbf{P}$ describes the rotational movement of the point **P**. With this we have:

$$\mathbf{P} - \mathbf{P}' = \mathbf{V} = -\mathbf{T} - \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \mathbf{P}$$

where P and P' represent the points before and after the motion. After some re-arrangement, we have the following form that looks familiar to us:

$$\mathbf{P}' = \mathbf{RP} + \mathbf{T} \tag{3}$$

where the rotation matrix is

$$\mathbf{R} = \begin{bmatrix} 1 & -\omega_z & \omega_y \\ \omega_z & 1 & -\omega_x \\ -\omega_y & \omega_x & 1 \end{bmatrix} \tag{4a}$$

Think about where the diagonal ones (1's) came from. Recall that a rotation matrix that is generated by performing three rotations around the X, Y and Z axis consecutively can be written as:

$$\mathbf{R} = \begin{bmatrix} \cos\beta\cos\gamma & -\cos\beta\sin\gamma & \sin\beta \\ \sin\alpha\sin\beta\cos\gamma + \cos\alpha\sin\gamma & -\sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & -\sin\alpha\cos\beta \\ -\cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma & \cos\alpha\sin\beta\sin\gamma + \sin\alpha\cos\gamma & \cos\alpha\cos\gamma \end{bmatrix} \quad (4b)$$

Equation (4b) will be almost the same as equation (4a) when the three angles α, β and γ are all very small angles.

The answer to the second question will lead to the motion field equation of rigid body motion. Taking the time derivative of both sides of the projection equation (1), we have

$$\mathbf{v} = \frac{f}{Z^2}(Z\mathbf{V} - V_z\mathbf{P})$$

Inserting the 3D motion equation (2) into the above equation, we obtain the motion field equation:

$$\begin{pmatrix} v_x \\ v_y \end{pmatrix} = \frac{1}{f}\begin{pmatrix} xy & -(x^2+f^2) & fy \\ y^2+f^2 & -xy & -fx \end{pmatrix}\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} + \frac{1}{Z}\begin{pmatrix} -f & 0 & x \\ 0 & -f & y \end{pmatrix}\begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix} \quad (5)$$

Rotation part: no depth information

Translation part: depth Z

Note that the motion field is the sum of two components: the translational part that includes the depth information, and the rotational part that doesn't have any depth information. Here we assume the intrinsic parameters are known.

Table 1 summarizes a comparison of motion field and stereo disparity.

Table 1. Motion field vs. stereo disparity

	Stereo	**Motion**
Terms	Disparity	Motion field
Concepts	Displacement – (dx, dy)	Differential concept – velocity (v_x, v_y), i.e. time derivative (dx/dt, dy/dt)
Constraints	Epipolar geometry for searching corresponding points	Consecutive frame close each other to guarantee good discrete approximation of derivative

2.2 Special Cases

Before discussing the general case of motion parallax, it would be interesting to see some very useful special cases: pure translation, pure rotation and motion of a plane.

2.2.1 Special Case 1: Pure Translation

Under pure translation (i.e., ω =0), the motion field can be simplified as,

$$\begin{pmatrix} v_x \\ v_y \end{pmatrix} = \frac{1}{Z}\begin{pmatrix} -f & 0 & x \\ 0 & -f & y \end{pmatrix}\begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix} \tag{6}$$

which can be further divided into the following two cases: radial motion field and parallel motion field.

1. ***Radial Motion Field*** ($Tz \neq 0$)

We define the *vanishing point* $p_0 = (x_0, y_0)^T$ as :

$$\begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = \frac{f}{T_z}\begin{pmatrix} T_x \\ T_y \end{pmatrix} \tag{7}$$

which can be used to compute the 3D motion direction. Then we can write the motion field equation as

$$\begin{pmatrix} v_x \\ v_y \end{pmatrix} = \frac{T_z}{Z}\begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix} \tag{8}$$

From equation (8), we can easily see that the vanishing point p_0 represents the focus of expansion (FOE) if $Tz < 0$, since all the motion vectors point away from p_0; if $Tz > 0$, it is called the focus of contraction (FOC) since all the motion vectors move towards p_0 (Figure 2a – TO DO) . The depth of a point can be estimated as,

$$Z = \frac{|T_z|}{|\mathbf{v}|}\sqrt{(x - x_0)^2 + (y - y_0)^2} \tag{9}$$

Here we can clearly see that the depth Z is inversely proportional to the magnitude of motion vector **v**, and proportional to both the translation magnitude in the Z direction and the distance from p to p_0.

2. *Parallel Motion Field* (Tz = 0)

The motion field of a translational motion with Tz = 0 can be written as

$$\begin{pmatrix} v_x \\ v_y \end{pmatrix} = -\frac{f}{Z}\begin{pmatrix} T_x \\ T_y \end{pmatrix} \tag{10}$$

which shows that the motion field is a parallel field in the direction of the motion (Tx,Ty) (Figure 2b – TO DO). The depth of a 3D point can be estimated as

$$Z = \frac{f}{|\mathbf{v}|}\sqrt{{T_x}^2 + {T_y}^2} \tag{11}$$

Again, the depth is inversely proportional to magnitude of motion vector **v**, and proportional to the 3D motion magnitude.

2.2.2 Special Case 2: Pure Rotation

Under pure rotation (**T** = 0), the motion field equation can be written as

$$\begin{pmatrix} v_x \\ v_y \end{pmatrix} = \frac{1}{f}\begin{pmatrix} xy & -(x^2+f^2) & fy \\ y^2+f^2 & -xy & -fx \end{pmatrix}\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \tag{12}$$

Clearly, the motion field of a pure rotation does not carry any 3D information; Z is not included in the equation. The motion vector equation is a quadratic polynomial function of image coordinates $(x,y,f)^T$. Given more than two points with known velocities, we will have more than four linear equations of the three angles, and therefore they can be fully estimated.

Note that in practice, the motion field equation is an approximation when the motion is very small. For pure rotation, we can actually write an accurate image transformation between two frames, and the rotational motion can be large. A 3D rotation transformation can be written as:

$$\mathbf{P'} = \mathbf{RP}$$

where P and P' are the 3D point representations before and after the rotation. Since we can write the projections of the 3D point before and after motion as,

$$\mathbf{p} = \frac{f}{Z}\mathbf{P} \quad \text{and} \quad \mathbf{p}' = \frac{f'}{Z'}\mathbf{P}'$$

we have

$$\mathbf{p}' \cong \mathbf{R}\mathbf{p} \tag{12}$$

where the equality is a projective equality. Equation (12) can be implemented using pure image transformation, therefore two images before and after a pure rotation can be precisely registered. This can be used for image mosaicing from a rotating camera, for example, to generate 360 degree panoramas.

2.2.3 Special Case 3: Moving Plane

Planes are common in the man-made world. So it is interesting to see what relation can be obtained for the motion field of a plane when the camera undertakes an arbitrary motion, or a moving plane is viewed by a stationary camera. A plane equation can be written as

$$\mathbf{n}^T\mathbf{P} = d$$

where n = (nx, ny, nz) ^{T} is the normal of the plane, and P = (X, Y,Z)^T is a 3D point on the plane. Using the camera projection equation, we have

$$\frac{1}{Z} = \frac{(n_x x + n_y y + n_z f)}{fd} \tag{13}$$

Inserting equation (13) into the motion field equation (5) and therefore eliminating Z, which is different from point to point, we will get a quadratic polynomial equation in the image, which only has 8 independent parameters. (Question 2. Write it out!). However, we again note that the motion field is only an approximation given small motion; if the motion between two images is large, we can use a precise image transformation. For an arbitrary motion, we have

$$\mathbf{p}' \cong \mathbf{A}\mathbf{p} \tag{14}$$

where **A** is the homography (3x3 matrix) for all points [Question 3. Derive homography]. Note that equation (14) has the same form as equation (12); therefore equation (12) can also be viewed as a special homography, which is a rotational matrix. Equation (14) is very useful for generating image mosaicing for a planar scene; examples include generating wide field-of-view mosaics from an aerial image sequence, or a video of a classroom blackboard.

2.2.4 Special Cases: Summary

Let us summarize the three special cases:

- Pure Translation
  - Vanishing point and FOE (focus of expansion)
  - Only translation contributes to depth estimation
- Pure Rotation
  - Does not carry 3D information
  - Motion field: a quadratic polynomial in image, or
  - Transform: Homography (3x3 matrix R) for all points
  - Image mosaicing from a rotating camera
- Moving Plane
  - Motion field is a quadratic polynomial in image, or
  - Transform: Homography (3x3 matrix A) for all points
  - Image mosaicing for a planar scene

2.3 Motion Parallax

To understand motion parallax, which reflects the different apparent motion vectors due to different depths of the 3D points, we make the following two observations:

[**Observation 1**] The relative motion field of two instantaneously coincident points does not depend on the rotational component of the motion, and points towards (or away from) the vanishing point of the translation direction.

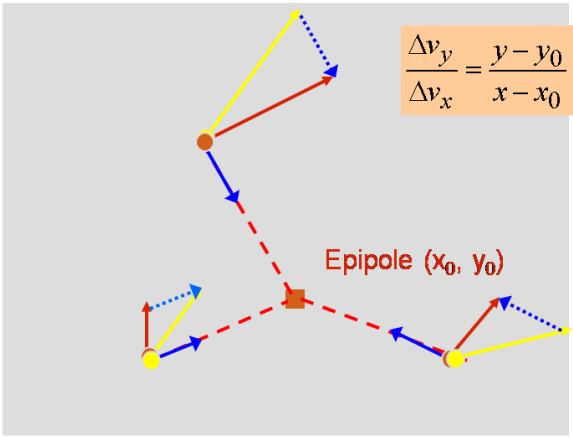

Figure 3. Motion parallax

Observation 1 is derived from image motion equation (5) where the pixel location (x,y) is the same for two points at the time t, thus instantaneously coincident points. Figure 3 shows how we can derive the motion parallax. At instant t, three pairs of points happen to be coincident. The difference of the motion vectors of each pair cancels the rotational components and the relative motion field point towards or away from the vanishing point of the translational direction.

[**Observation 2**] The motion field of two frames after rotation compensation includes only the translation component

$$\frac{v_y^T}{v_x^T} = \frac{y - y_0}{x - x_0} \tag{15}$$

The motion field also has the following two properties:
(1) It points towards (or away from) the vanishing point p0 (the *instantaneous epipole*)
(2) The length of each motion vector is inversely proportional to the depth, and directly proportional to the distance from the point p to the vanishing point p0 of the translation direction (if Tz <> 0)

$$|\mathbf{v}| = \frac{T_z}{Z}\sqrt{(x - x_0)^2 + (y - y_0)^2} \tag{16}$$

Rotation compensation can be done by image warping after finding three pairs of coincident points [Question 4]

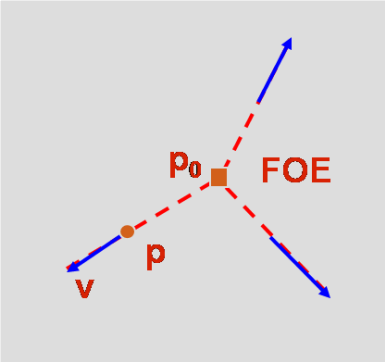

Figure 4. Instantaneous epipole

2.4 Summary

Here is a brief summary of the basic visual motion concepts.

- Image motion field of rigid objects
  - Time derivative of both sides of the projection equation
- Three important special cases
  - Pure translation – FOE
  - Pure rotation – no 3D information, but lead to mosaicing
  - Moving plane – homography with arbitrary motion
- Motion parallax
  - Only depends on translational component of motion

3. OPTICAL FLOW

In this section, we will discuss the basic techniques for estimating the motion field using *optical flow*. Three important aspects will be discussed: (1) the notation of optical flow; (2) the estimation of optical flow; and (3) the use of optical flow.

3.1 Notation of Optical Flow

An image sequence can be represented as a 3D cube I(x,y,t). A point in the 3D cube at location (x,y,t) with intensity I(x,y,t) will move by δx, δy and δt between the two image frames due to the relative motion between the camera and the scene. Under most circumstances, the apparent brightness of moving objects remains constant. Then, we have the following *brightness constancy equation*:

$$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) \tag{16a}$$

Assuming the movement to be small, the left hand side of equation (16a) can be expanded to a Taylor series at I(x,y,t) as:

$$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t + \text{H.O.T.} \tag{16b}$$

Comparing equations (16a) and (16b), and ignoring the higher order terms (H.O.T.), we arrive at the famous optical flow equation:

$$E_x u + E_y v + E_t = 0 \tag{16}$$

where Ex, Ey and Et are the partial derivatives (i.e., ∂I/∂x, ∂I/∂y, ∂I/∂t) of I(x,y,t) in x, y, and t directions, respectively, and (u,v) =(δx/δt, δy/δt) is the optical flow vector at the current point (x,y,t).

The optical flow equation builds a relation between the apparent motion (u,v), and the spatial and temporal derivatives of the image brightness function; the derivatives can be estimated directly from the image function I(x,y,t). However, it includes two variables in a single equation for each point. This leads to the *aperture problem* as shown in Figure 5: only the component of the motion field in the direction of the spatial image gradient can be determined. An example of the aperture problem is the barber pole illusion. The stripes within the pole appear to move upwards, but the actual motion is a horizontal movement. The component in the direction perpendicular to the spatial gradient is not constrained by the optical flow equation, unless a corner can be seen in the small aperture window that is used to calculate the spatial-temporal gradients [Question 5]. In the following subsection, we will discuss a few basic techniques to solve this problem in more principled ways.

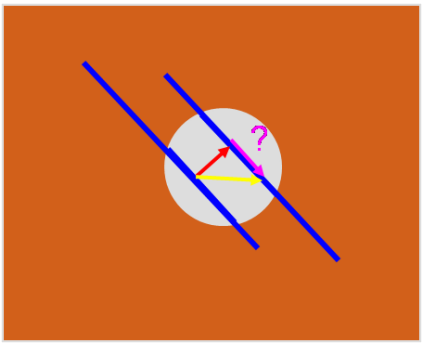

Figure 5. Aperture problem

3.2 Estimating Optical Flow

We will discuss three techniques for estimating optical flow: the constant flow method, the weighted least square method and the affine flow method. All the three methods are designed to give solutions to the aperture problem and to generate a dense optical flow field. The complexity of the methods is from the simple to sophisticated and accurate.

1. ***Constant Flow Method***
   - **Assumption:** The motion field is well approximated by a constant vector within any small region of the image plane (corresponding to a frontal planar patch).

- **Solution:** Least square of two variables (u,v) from n×n equations that are provided by an n×n spatial window in I(x,y) at time t. Thus a linear equation system can be formed as $\mathbf{Av} = \mathbf{b}$, where $\mathbf{A}$ is the $n^2 \times 2$ coefficient matrix, $\mathbf{v}=(u,v)$, and $\mathbf{b}$ is an n^2 dimension vector.
- **Condition**: $\mathbf{A}^T\mathbf{A}$ is NOT singular (null or parallel gradients)

2. ***Weighted Least Square Method***
   - **Assumption**: The motion field is approximated by a constant vector within any small region, and the error made by the approximation increases with the distance from the center where optical flow is to be computed
   - **Solution**: Weighted least square of two variables (u,v) from n×n equations that are provided by an n×n spatial window in I(x,y) at time t

3. ***Affine Flow Method***
   - **Assumption**: the motion field is well approximated by a affine parametric model $\mathbf{u}^T = \mathbf{Ap}^T + \mathbf{b}$ (a plane patch with arbitrary orientation)
   - **Solution**: Least square of 6 variables (**A,b**) from n×n equations that are provided by an n×n spatial window in I(x,y) at time t

Figure 6. Estimating optical flow: three methods

3.3 Using Optical Flow

Now we give a sketch of the algorithm to infer 3D motion and structure from optical flow.

[Input] We assume that the intrinsic camera parameters are known, and a dense motion field (optical flow) of single rigid motion has been already estimated.

[Algorithm]The algorithm is a good compromise between ease of implementation and quality of results.

Stage 1: Translation direction

The instantaneous epipole (x_0, y_0) can be obtained through approximating the motion parallax. Then, we can obtain the translational vector up to a scale s (following equation (7)):

$$(T_x, T_y, T_z) = s\,(x_0, y_0, f) \tag{16}$$

The key is to find more than two pairs of instantaneously coincident image points. In reality, an approximation is made to estimate differences of motion vectors for almost coincident image points, then the epipole can be found by the intersection of the translational motion vectors (equation (15), Figures 3 and 4).

Stage 2: Rotation flow and depth

For a point (x,y) in the reference image, since we know its original flow vector (v_x, v_y), and the direction of 3D translational component as in equation (16), we can derive one linear equation (without including depth) of the three angles from equation (5). This can be achieved by re-arranging equation (5) as:

$$\begin{pmatrix} v_x \\ v_y \end{pmatrix} - \frac{1}{f}\begin{pmatrix} xy & -(x^2+f^2) & fy \\ y^2+f^2 & -xy & -fx \end{pmatrix}\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = \frac{1}{Z}\begin{pmatrix} -f & 0 & x \\ 0 & -f & y \end{pmatrix}\begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix} \tag{17}$$

Equation (17) includes two equations; by dividing the first one by the second one, we arrive at a linear equation of the three angles $(\omega_x, \omega_y, \omega_z)$. Then a least approximation can be calculated by using at least three points with known motion vectors.

Finally, the depth Z of each point can be found up to the same scale s, by using equation (17), given that we know the angular velocity $(\omega_x, \omega_y, \omega_z)$, and the translation vector $(T_x, T_y, T_z) = s\,(x_0, y_0, f)$ up to the scale s.

[Output]: As a summary, here is the list of outputs using the algorithm:

- Direction of translation $(f\,T_x/T_z,\ f\,T_y/T_z,\ f) = (x_0, y_0, f)$, thus the translation vector up to a scale s: $(T_x, T_y, T_z) = s\,(x_0, y_0, f)$
- Angular velocity $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$
- Depth Z (up to a scale s) given T/s and $\boldsymbol{\omega}$
- 3-D coordinates of scene points (up to a common unknown scale s)

4. FEATURE-BASED APPROACH (FROM HAO TANG'S SURVEY)

4.1 Two frame method - Feature matching approach

The key in a feature matching method is to detect features. A number of feature detection methods can be found in Section 4.2.2 in the Stereo Vision Chapter. After features are extracted from two consecutive frames, a match can be performed between them. Due to the small motion between the two consecutive frames, corresponding features should be very similar to each other. After the feature correspondences are built, we can treat them as a sparse motion field. Then, the optical flow method we described in Section 3.3 can be applied to estimate 3D structure and motion. Alternative more general approaches have been developed for solving the structure from motion problem even under large motion.

First we assume that the intrinsic parameters of a camera or cameras are known. After correspondences based on feature points between two or more different views are found, relative camera poses can be estimated. The problem can be stated as the following:

Problem statement: recovering relative camera poses from the correspondences of a set of 2D points in two consecutive 2D frames of the image sequence.

Nonlinear solution

The basic idea of a direct solution is as follows. Two observations of a 3D point in the two images give us four measurements, i.e., a pair of 2D points (x1,x2), (y1,y2), but we have three unknowns, i.e., its 3D coordinates <X, Y, Z>. This is in addition to the 6 unknown extrinsic parameters (3 for orientation and 3 for translation) common for all the points. Thus, if n points are observed in the two views, we have $6 + 3n$ unknowns, and $4n$ measurements (nonlinear equations). If we only have to recover the camera translation up to a scale factor, we only have $5 + 3n$ unknowns. Therefore, in theory five corresponding points are enough to solve the nonlinear system. Kruppa (1913) proves that the system may have up to eleven solutions using five points. Recently, Nistér (Nistér 2004) proposed an efficient solution. It uses the epipolar constraint, and constraints of the fundamental matrix and the essential matrix to construct a ten-degree polynomial and then obtain the camera pose by solving it. Because the solution is somewhat ad hoc, H. Li and R. Hartley (Li and Hartley 2005) give a simple solution using a hidden variable technique.

Linear solution

A non-iterative algorithm (i.e., 8-point algorithm) to solve the problem of relative camera placement was first given by (Longuet-Higgins 1981). Assuming the intrinsic parameters of the camera are known, the extrinsic parameters (R and T) can be solved given 8 pairs of point correspondences in two views. A unique solution may be obtained by solving the linear equation system. Later, the algorithm was generalized for uncalibrated cameras (Hartley and Zisserman 2000), i.e., cameras with unknown intrinsic parameters.

4.2 Multiple frame method – SLAM and SfM

Pose estimation is a key step in visual motion. In this subsection, typical solutions that obtain both camera pose and structure of sparse features are reviewed. These solutions are classified into two groups: *Simultaneous Localization and Mapping (SLAM)*, and *Structure form Motion (SFM)*.

SLAM was first proposed in robot field around mid-1980s (Smith, and Cheeseman 1986). This topic addresses an important problem in robotics: to build up a map in an unknown environment meanwhile keeping track of a robot's current position. It is typically treated as the problem of estimating spatial uncertainties of both scene structure and the robot's location, and it is usually modeled in a probabilistic framework. The Kalman filtering is one of the most popular mathematic tools used in SLAM systems. It is a linear Gaussian filter within the probabilistic framework to produce optimal estimates of the states of a dynamic system. Starting from the 90's, the framework has been extensively applied in the robotic field to solve SLAM problems (Leonard and Durrant-whyte 1991). In robot navigation, the estimation of a mobile robot's locations at different time stamps is assumed to be a dynamic system. At any time stamp, the Kalman filter predicts a robot location based on the estimated location in the previous time stamp under the assumption that the predication has linear relation with the robot actual location and that the noise complies with Gaussian. Then, a sensor (i.e., a laser range finder) gives a measurement of robot location, which has a linear relation with the robot's actual location with a Gaussian noise added. Then, the final optimal estimate of robot location is obtained by combining both measurement and predication.

Basics of the Kalman Filtering: (TO DO)

Different traditional range sensors (e.g. laser rangefinder and sonar sensor) are used in SLAM. Recently, however, video camera has been applied because it can provide richer information than the traditional sensors, although the algorithms with a camera are not yet as robust as people expect them to be.

On the other hand, classical SFM methods solve the "mapping and localization" problem by using iterative global optimization methods (bundle adjustments) that minimizes the overall re-projection errors of sparse features among an image sequence (Hartley and Zisserman 2000). Recently these two classes of methods are coming together; many solutions have been provided by combining the above two groups of methods (e.g., Mouragnon 2006, Klein and Murray 2007).

Recent advances in vision algorithms and hardware enable the design and implementation of real time visual navigation systems. However, algorithms using pure vision methods still face the well-known problems (i.e. drift and break), though systems using stereo camera can produce relatively reliable results. One general solution relies on an optimization method, either locally or globally, such as in bundle adjustment techniques [XXX]. However, solving the problem in real-time for a large environment is not recommended due to the expensive computation of global optimization. The PTAM partially solves the problem, but it's still limited to a small work space. Therefore, incorporating probabilistic framework gives us an alternative solution to build a more reliable system. In other words, the Kalman filter is beneficial when processing time is limited, otherwise bundle adjustments can give optimized solutions. This is shown in the work of Strasdat, et al. (Strasdat, et al. 2010). Further, hardware improvement or hardware speedup may enable real time visual odometry systems using global optimization to produce reliable and accurate results in the near future.

4.3 Using a sparse motion field - Factorization method (More here)

- 3D motion and structure by feature tracking over frames
- Factorization method
  - Orthographic projection model
  - Feature tracking over multiple frames
  - SVD

Figure 7. Factorization

5. MOTION-BASED VIDEO COMPUTING

5.1 Change Detection (From Tao Wang's Survey)

There are mainly two conventional approaches to change detection with a stationary camera: temporal difference and background subtraction. Note that in surveillance applications, cameras are usually stationary.

The first approach (temporal difference) consists of the subtraction of two consecutive frames followed by thresholding. The second approach (background subtraction) includes the subtraction of a reference background model and current image followed by a labeling process. Those images are usually smoothed in the preprocessing and the noise in the difference image is reduced by morphological operations.

The advantages of using temporal difference are: 1) it is adaptive to changes in dynamic environment; and 2) no assumption is made about the scene. However, only the motion at edges is detectable for a homogeneous object. On the other hand, background subtraction has better performance in extracting whole objects, but it is sensitive to dynamic changes in the environment.

Therefore, the performance of the object detection using background subtraction is significantly affected by the background modeling step. The terms foreground and background are not scientifically defined. A moving object is usually considered as a foreground but when it remains still for a long period of time, it is considered a part of the background. Also, it is possible to have a moving object to be considered a part of the background if it is not of interest for the target application. Therefore the meaning of background may vary across different tasks. In short, a relatively static model in the scene is considered as the background which is partially occluded by entering objects that is considered as the foreground. For video sequence, the background needs to be continuously updated during a period of time in order to make the foreground extraction more robust. However, a good updating of the background model is very difficult to achieve due to the factors of illumination variance, movement of the background objects such as shaking branches, their shadows, etc. A good overview of the most frequently cited background modeling algorithms is given in (McIvor, 2000). A comparison between various background modeling algorithms is given in (Toyama et al., 1999), as well as a discussion on the general principles of background maintenance systems.

The background modeling can be achieved using a Gaussian model that models the intensity of each pixel with a single Gaussian distribution (Wren et al., 1997) or with a mixture of Gaussians (Stauffer and Grimson, 2000; Tian et al., 2008a). A single Gaussian models the color of each pixel of a stationary background with a single 3D (Y, U, and V color space) Gaussian. Initially several consecutive frames are trained for the mean and covariance of the background model. The likelihood of pixel color is computed for every pixel in the input frame. The pixels that deviate from the background model are labeled as the foreground pixels. However, a single Gaussian is not a good model to multiple color changes due to the repetitive object motion, shadows or reflectance in outdoor surveillance (Gao et al., 2000). Stauffer and Grimson (2000) use a mixture of Gaussians to check a pixel in the current frame against the background model by comparing it with every Gaussian in the model until a matching Gaussian is found. If a match is found, the mean and variance of the matched Gaussian is updated; otherwise a new Gaussian with the mean equal to the current pixel color and some initial variance is introduced into the mixture. Each pixel is classified based on whether the matched distribution represents the background process. However, it cannot be used to adapt to quick lighting changes and handle shadows. Tian et al. (2008a) model the mixture of Gaussians adaptively and integrate texture information of the area that is caused by the lighting changes similar to the background into the foreground mask computation. To remove shadows, they normalize the intensities calculated at each pixel of the foreground region between the current frame and background image. Finally, the abandoned objects are detected from the static foreground using a region growing method.

In outdoor surveillance, most color bands are sensitive to illumination variations. In scenarios where the illumination effect is inevitable, optical flow based method can be used to detect independent moving objects even in the presence of camera motion. Although it is commonly used as a feature for contour tracking (Cremers et al., 2003), it is useful in motion segmentation where a motion vector is assigned to every pixel of the image by comparison of successive frames. Thus, object correspondences can be built to extract the foreground objects from the background. More detailed discussion of using optical flow can be found in Barron (1994) and Shin (2005).

In the last few years, there has been increasing interest in moving target detection (particularly human detection) using cameras on moving platforms. This is more challenging than object detection from stationary cameras, but these methods could be used for surveillance applications when cameras are mounted on an aerial or a ground vehicle, and will extend the capacity of traditional video surveillance. Typically, there are two approaches in object

(human) detection with a moving camera. The first approach uses a brute force multi-scale shift window to generate many candidates for possible humans, and rely on the following classifiers to pick up the correct human regions (Wojek et al., 2009). The second approach uses a motion-based background alignment, for example, by assuming the moving people are on a relatively flat ground plane. This is valid for high-altitude aerial surveillance or low-altitude aerial surveillance with a relative flat background (e.g., Yu and Medioni, 2009), or a moving vehicle on a flat road (e.g., Enzweiler et al., 2008). But for aerial or ground surveillance of urban areas, a 3D background alignment (Tang and Zhu, 2008) approach and a multi-frame affine background modeling approach (Sheikh, et al, 2009) have been proposed.

5.2 Video mosaicing (From Zhu's ECV Item)

Image mosaicing is the process of generating a composite image (mosaic) from a video sequence, or in general from a set of overlapping images of a scene or an object, usually resulting in a mosaic image with a larger field of view than any of the original images.

5.2.1 Background

When collecting video about a scene or object, each individual image in the video may be limited compared to the desired final product, including limitations in the field of view, dynamic range, or image resolution. This is the case not only with personal video capture [1,9, 10], but also with image-based rendering [12, 14,15], aerial videography [7, 11, 18-20], and document digitization [5]. Generating mosaics with larger fields of view [5, 6, 9, 10, 14, 20], higher dynamic ranges [4], and/or higher image resolutions [8] facilitates video viewing, video understanding, video transmission and archiving. When the major objective of video mosaicing is to generate a complete (e.g. 360 degrees) view of an object (or a scene) by aligning and blending a set of overlapping images, the resulting image is also called a video panorama [10, 14, 15].

5.2.2 Theory and Application

Video mosaicing takes as input a video sequence and generates one or more mosaiced images with either a larger field of view, a higher dynamic range, a higher image resolution, or a combination of them. This entry will mainly discuss the principles in generating large field of view mosaics

(panoramas), but the principles can also be applied to mosaics for other objectives (high dynamic range imaging and super-resolution imaging). Here, *video* mosaicing implies that the images in the sequence are taken by a video camera, usually at 30 frames per second, but images taken by a digital camera in such a way that there is a large amount of spatial overlap between two consecutive frames can also be viewed as a video sequence.

There are three key components in a typical video mosaicing algorithm: (1) motion modeling, (2) image alignment, and (3) image composition.

Depending on the type of camera motion and the structure of the objects or scenes, the ***motion model*** can be a 2D rigid motion model (rotation, translation, scaling), an affine model, a perspective model (homography), or a full 3D motion model. Many popular video mosaicing methods [16], e.g., in [4, 15], assume a pure rotation model of the camera in which the camera rotates around its center of projection (i.e., the optical center, sometimes called nodal point). In this case, the motion between two consecutive frames can be modeled by a homography, which is a 3x3 matrix. Then, depending on the fields of view (FOVs) of the mosaic, the projection model of the mosaic can be either a perspective projection (FOV is less than 180 degrees), a cylindrical projection (FOV is 360 degrees in one direction), or a spherical projection (full 360 degrees FOV in both direction). Figure 8 illustrates the relations between the original images and the three types of mapping surfaces each image can be projected onto: planar, cylindrical and spherical surfaces.

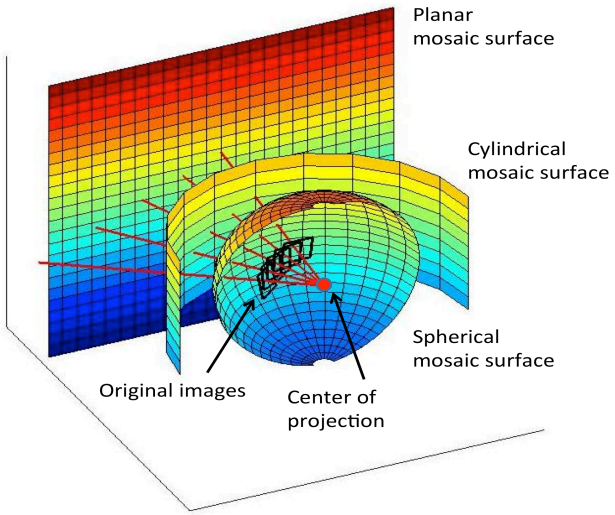

Figure 8. Mapping a set of overlapping images into a mosaic: planar, cylindrical or spherical

However, the applications of video mosaics from a pure rotation camera are limited to mostly consumer applications such as personal photography, entertainment and online maps. For more specialized applications such as surveillance, remote sensing, robot navigation, and land planning, to name a few, the motion of the camera cannot be limited to a pure rotation. Translational motion usually cannot be avoided, causing the *motion parallax* problem to arise. There are three kinds of treatments for the motion parallax problem.

First, when the translational components are relatively small, the motion models can be approximated by a pure rotation. In this case, the generated mosaics lack geometric accuracy but with some treatments for the small motion parallax and moving targets, such as de-ghosting [15] the mosaics usually look very good.

Second, if the scene can be regarded as planar, for example, because the distance between the camera and the scene is much larger than the depth range of the scene, the perspective motion model (homography) or in some applications, a 2D rigid motion model or an affine model can be used [6,11,19]. In these cases, the problems are much simpler due to the 2D scene assumption.

Finally, a 3D camera motion model is applied when the translational components of the camera motion are large and the scene is truly 3D. In this case motion parallax cannot be ignored or eliminated. Examples include a camera mounted on an airplane or a ground vehicle translating a large distance [7, 9, 12, 20], or a camera's optical center moving on a circular path [10, 14]. Here, multi-perspective projection models are used to generate the mosaics, enabling stereo mosaics or stereo panoramas to be created that preserve the 3D information in the scene, allowing the structure to be reconstructed and viewed in 3D. In this case, the accuracy of geometric modeling and image alignment is crucial for achieving the accuracy of 3D reconstruction and viewing.

Image alignment (or ***image registration***) is the process of finding the alignment parameters (e.g., the homography in the rotational case) between two consecutive images. Image alignment is a critical step in mosaic generation, for both seamless mosaicing and for accurate geometric representation. There are two approaches to image registration: direct methods and feature-based methods.

In a direct method, a correlation approach is used to find the motion parameters. Here, the images are divided into small blocks and each block in the first image is searched for over a predefined spatial range in the second image. The best match is determined by finding the maximal correlation value. Other approaches such as using optical flow or using an iterative

optimization framework also belong to the direct methods, in which no explicit feature points are extracted.

In a feature-based method, a feature detection operator such as the Harris corner or SIFT (Scale Invariant Feature Transform) detector is used first, then the detected features are matched over the two frames to build up matches [16]. Either way, a parameter model is fitted using all the matches, usually using a robust parameter estimation method to eliminate erroneous feature matches. For more accurate or consistent results, a global optimization can be applied to more than two frames. For example, global alignment may be applied to all the frames in a full 360-degree circle in order to avoid gaps between the first and the last frame [15].

Image composition is the step of combining aligned images together to form the viewable mosaic. There are three important issues in this step: (1) compositing surface determination, (2) coordinate transformation and image sampling, and (3) pixel selection and blending. Mosaicing with the rotational camera model is a good starting point to discuss these issues (Figure 8); mosaic compositing under other motion models are discussed afterwards.

If the video sequence only has a few images, then one of the images can be selected as the reference image, and all the other images are warped and aligned with this reference image. In this case, the reference image with a perspective projection is the compositing surface, and therefore the final mosaic is a larger perspective image, which is an extension of the field of view of the reference image. However, this approach only works when the view angles of the images span less than 90 degrees. If the camera rotates more than 90 degrees, a cylindrical or a spherical surface should be selected as the compositing surface. A cylindrical surface is a good representation when a full 360 panoramic mosaic is to be generated, in one direction. And a spherical surface is suitable if 360x360 degree mosaics are to be created.

After a compositing surface is selected, the next issue is coordinate transformation and sampling. This is also called image warping. Given the motion parameters obtained in the image registration step, the mapping between each frame to the final compositing surface can be calculated: for any pixel in an original image frame, its pixel location in the composition surface can be calculated. For generating dense pixels, an interpolation schema is needed, such as nearest neighbor, bilinear, or cubic interpolation methods. Usually a backward mapping relation is utilized such that in the mapping area on the compositing surface, each pixel obtains a value from the original image frame, line by line, and column by column. Therefore, for each integer pixel location in the mosaic, a decimal pixel location can be found in the original image; then an interpolation method is used in the original image to generate the value of the pixel in the mosaic.

The third important issue in image composition is pixel selection and blending. Naturally in generating mosaics, there are overlaps among consecutive frames, resulting in two key questions: First, *Where do we place the seam (i.e., the stitching line)?* (the pixel selection problem.) Second, *How do we select the values of pixels in the overlapping areas?* (the pixel blending problem.) For the second problem, the simplest methods are to average all the pixels in the same location in the overlapping area, or to use their median value. The former might create a so called ghost effect due to moving objects, small motion parallax or illumination changes, while the latter approach may generate a slightly better view effect. More sophisticated blending methods include Laplacian pyramid blending [3] and gradient domain blending [1]. The pixel selection problem is important when moving objects or motion parallax exist in the scene. In these cases, to avoid a person being cut in half or appearing twice in the mosaic, or to avoid cutting a 3D object that exhibits obvious motion parallax and hence could produce obvious misalignment in the mosaic, an optimal seam line can be selected at pixel locations where there are minimum misalignments between two frames [4].

Other considerations in image composition are high dynamic range imaging [4] and improved image resolution mosaicing [8]. For the former, a composite mosaic represents larger dynamic ranges than individual frames using varying shutter speeds and exposures, while the latter uses the camera motion to generate higher spatial resolution in the mosaiced image than that of the original images.

So far the discussions on image composition have focused primarily on 2-D mosaics, assuming either the camera motion is (almost) a pure rotation, or the scene is flat or very far from the camera, in order to avoid or reduce the motion parallax problem. When motion parallax cannot be avoided, 3-D mosaics have to be considered. Methods have been proposed to generate mosaics, for example, for curved documents based on 3-D reconstruction [5], when the camera motion has translational components. Needless to say, with 3-D reconstruction, a composite image with a new perspective view, or a new projection representation (such as orthogonal projection), can be synthesized from the original images. However, a drawback of this approach is that a full 3-D reconstruction is needed, which is both computationally expensive and prone to noise. A more practical yet still fundamental approach without 3-D reconstruction is to generate multi-perspective mosaics from a video sequence, such as mosaics on an adaptive manifold [11], creating stitched images of scenes with parallax [7], and creating multiple-center-of-projection images [12]. When the dominant motion of the

camera is translation, the projection model of the mosaic can be a parallel-perspective projection, in that the projection in the direction of the motion is parallel, whereas the projection perpendicular to the motion remains perspective. This kind of mosaic is also called pushbroom mosaic [17] since the projection model of the mosaic in principle is the same as pushbroom imaging in remote sensing. A more interesting case is that by selecting different parts of individual frames, a pair of stereo mosaics can be generated that exhibit motion parallax, while each of them represent a particular viewing angle of parallel projection [20]. To generate stereo mosaics, the motion model is 3D and therefore a bundle adjustment for 3D camera orientation is needed. The projection model is parallel-perspective, and therefore the composition surface is a plane that holds the parallel-perspective image. To generate a true parallel-perspective view in each mosaic for accurate 3D reconstruction, pixel selection is carried out for that particular viewing angle and a coordinate transformation is performed based on matches between at least two original images for each pixel. A similar principle can be applied to concentric mosaics with circular projection [10, 14].

In some applications such as surveillance and mapping, geo-referencing mosaicing is also an important topic. This is usually done when geo-location metadata is available, for example, from GPS and IMU measurements [18, 19] taken with the video/images. Geo-referenced mosaics assign a geo-location to each pixel either by directly using the metadata from the video frames used to generate the mosaic, or when metadata is not available, the video frames are aligned to a geo-referenced reference image such as a satellite image.

Video mosaicing techniques are also used for dynamic scenes, for example, to generate dynamic pushbroom mosaics for moving target detection [17], and to create animated panoramic video textures in which different portions of a panoramic scene are animated with independently moving video loops [2, 13].

Figure 9 shows a 360-degree panoramic mosaic represented on a cylindrical surface, which is generated from a video sequence taken by a video camera that roughly rotates around its optical center. Figures 10 and 11 show two stereo mosaics that can be viewed with a pair of 3D glasses, red for the right eye and the cyan for the left eye. High resolution mosaics can be viewed by clicking the images in the figures in the online edition. The concentric stereo mosaic in Figure 10 is generated from a video sequence taken by a hand-held video camera that undertakes an off-center rotation

with 360 degrees of field of view coverage. Figure 11 is a pair of pushbroom stereo mosaics created from a video sequence taken by a camera looking down from an airplane flying over the Amazon rain forest.

Figure 9. A 360-degree panoramic mosaic generated on a cylindrical surface

http://www-cs.engr.ccny.cuny.edu/~zhu/ThlibCylinder.JPG

Figure 10. A pair of concentric mosaics of the City College of New York campus

http://www-cs.engr.ccny.cuny.edu/~zhu/CSCI6716/CCNYCampus.jpg

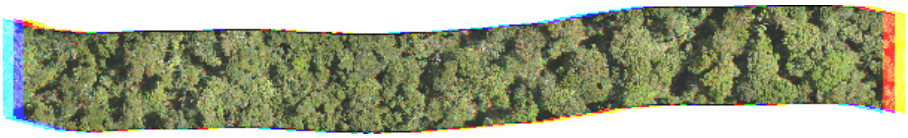

Figure 11. A pair of pushbroom mosaics of the Amazon rain forest

http://www-cs.engr.ccny.cuny.edu/~zhu/57z10StereoColor.jpg

5.3 Layered representation (From Hao Tang's Survey)

Even though a precise 3D model reconstruction is often desirable, it is not a trivial task and may not be required in some applications, such as image-based rendering and video coding. Therefore, layered-based methods, under the names of motion segmentation, layered segmentation, or layered representations have been proposed to represent a 3D scene from multiple images or a video sequence captured by a camera without explicitly computing the real 3D model. Common to these methods is that pixels sharing a common motion model are grouped into a single motion layer by analyzing the 3D geometric relations among different layers in the scene. Both ego-motion induced by the camera and independent motion induced by moving targets can be represented.

In order to improve the robustness, the layered segmentation problem is usually modeled as a global optimization problem. Representative approaches to solving the problem include Bayesian frameworks (Torr et al. 2001), rank constraints (Ke and Kanade 2001, 2004) and graph cuts (Xiao and Shah 2004).

The layered segmentation task usually consists of three sub-tasks:

(1) the initial estimation of the number of layers;

(2) the parameter (motion model) estimation of each layer; and

(3) the assignment of a layer label to each pixel.

The algorithms of the layer segmentation are usually performed iteratively.

6. CONCLUDING REMARKS

After learning motion, you should be able to:

- Explain the fundamental problems of motion analysis
- Understand the relation of motion and stereo
- Estimate optical flow from an image sequence
- Extract and track image features over time
- Estimate 3D motion and structure from sparse motion field
- Extract depth from a video under translational motion
- Know some important applications of motion, such as change detection, image mosaicing and motion-based segmentation

7. QUESTIONS AND PROJECTS

7.1 Questions

1. Could you obtain 3D information of a scene by viewing the scene by a camera rotating around its optical center? **Show why or why not.** What about moving the camera along its optical axis?

2. Write out the equation of a moving plane.

3. Derive the homography of a moving plane.

4. Show that the rotation compensation can be done by image warping after finding three (3) pairs of coincident points

5. Show that the aperture problem can be solved if a corner is visible through the aperture.

7.2 Projects

[TO DO]

REFERENCES

Digital Image Processing
Computer Graphics
Photogrammetry

E. Trucco and A. Verri. Introductory Techniques for 3 - D COMPUTER VISION, 1st ed. Prentice Hall, 1998.

Recommended Readings (Zhu's ECV Item)

[1]. Agarwala, A., Dontcheva, M., Agrawala, M., Drucker, S., Colburn, A., Curless, B., Salesin, D. & Cohen, M. (2004). Interactive digital photomontage. ACM Transactions on Graphics, 23(3), 292–300.

[2]. Agarwala, A., Zheng, C., Pal, C., Agrawala, M., Cohen, M., Curless, B., Salesin, D., Szeliski, R. (2005). Panoramic video textures. ACM Transactions on Graphics, 24(3),821–827.

[3]. Burt, P. J. and Adelson, E. H. (1983). A multiresolution spline with applications to image mosaics. ACM Transactions on Graphics, 2(4), 217–236.

[4]. Eden, A., Uyttendaele, M., and Szeliski, R. (2006). Seamless image stitching of scenes with large motions and exposure differences. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2006), pages 2498–2505

[5]. Iketani, A., Sato, T., Ikeda, S., Kanbara, M., Nakajima, N., Yokoya, N. (2006). Video Mosaicing for Curved Documents Based on Structure from Motion. In ICPR (4): 391-396

[6]. Irani, M., Anandan, P., Hsu, S.C. (1995). Mosaic Based Representations of Video Sequences and Their Applications. In ICCV: 605-611

[7]. Kumar, R., Anandan, P., Irani, M., Bergen, J., and Hanna, K. (1995). Representation of scenes from collections of images. In IEEE Workshop on Representations of Visual Scenes, pages 10–17

[8]. Marzotto, R., Fusiello, A., Murino, V. (2004). High Resolution Video Mosaicing with Global Alignment. In CVPR (1): 692-698

[9]. Rousso, B., Peleg, S., Finci, I., Rav-Acha, A. (1998). Universal Mosaicing using Pipe Projection. In ICCV: 945-952

[10]. Peleg, S., Ben-Ezra, M. (1999). Stereo Panorama with a Single Camera. In CVPR: 1395-1401

[11]. Peleg, S., Rousso, B., Rav-Acha, A., Zomet, A. (2000). Mosaicing on Adaptive Manifolds. IEEE Trans. Pattern Anal. Mach. Intell.: 1144-1154

[12]. Rademacher, P. and Bishop, G. (1998). Multiple-center-of-projection images. In Computer Graphics Proceedings, Annual Conference Series, pages 199–206

[13]. Rav-Acha, A., Pritch, Y., Lischinski, D., and Peleg, S. (2005). Dynamosaics: Video mosaics with non-chronological time. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005), pages 58–65

[14]. Shum, H.-Y. and Szeliski, R. (1999). Stereo reconstruction from multiperspective panoramas. In Seventh International Conference on Computer Vision (ICCV'99), 14–21

[15]. Shum, H., Szeliski, R. (2000). Systems and Experiment Paper: Construction of Panoramic Image Mosaics with Global and Local Alignment. International Journal of Computer Vision 101-130

[16]. Szeliski, R. (2006). Image Alignment and Stitching: A Tutorial. Foundations and Trends in Computer Graphics and Vision, 2(1): 1-104

[17]. Tang, H., Zhu, Z., Wolberg, G. (2006). Dynamic 3D Urban Scene Modeling Using Multiple Pushbroom Mosaics. In 3DPVT: 456-463

[18]. Taylor, C.N., Andersen, E.D. (2008). An automatic system for creating geo-referenced mosaics from MAV video. In IROS: 1248-1253

[19]. Zhu, Z., Riseman, E.M., Hanson, A.R., Schultz, H.J. (2005). An efficient method for geo-referenced video mosaicing for environmental monitoring. Mach. Vis. Appl.: 203-216

[20]. Zhu, Z., Hanson, A.R., Riseman, E.M. (2004). Generalized Parallel-Perspective Stereo Mosaics from Airborne Video. IEEE Trans. Pattern Anal. Mach. Intell.: 226-237