Structure from Motion

Shree K. Nayar

Monograph: FPCV-4-4

Module: Reconstruction II

Series: First Principles of Computer Vision

Computer Science, Columbia University

April, 2025

FPCV Channel

FPCV Website

Structure from Motion

Shree K. Nayar
Columbia University

Topic: Structure from Motion, Module: Reconstruction II
First Principles of Computer Vision

1

Uncontrolled (Casual) Video

[Pollefeys 2002] 2

This lecture is about an interesting vision problem known as structure from motion. Consider a video of the artifact shown on the right taken while simply walking around it. Such a video can be referred to as a casual, or uncontrolled, video because the motion of the camera during the capture is unknown. It turns out that from this video, we can not only reconstruct the 3D structure of the artifact but also determine how the camera moved in 3D space during the capture of the video. The technique that makes all this possible is called structure from motion.

Structure from motion is a method for computing from a video the 3D structure of a scene and the motion of the camera during the capture of the video. We will begin by formulating the structure from motion problem. The first step is to track features through the entire video. The image coordinates of these features are then arranged as a single matrix called the observation matrix. Our goal then is to recover scene structure and camera motion from the observation matrix.

Structure From Motion

Compute 3D scene structure and camera motion from a sequence of frames.

Topics:

(1) Structure from Motion Problem

(2) SFM Observation Matrix

(3) Rank of Observation Matrix

(4) Tomasi-Kanade Factorization

3

What makes the observation matrix interesting is that it has a very low rank. This allows us to construct a rank constraint, which can be used to decompose the observation matrix into two matrices: the scene structure matrix and the motion matrix. This algorithm was initially developed by Tomasi and Kanade and is called the factorization algorithm. We will discuss the algorithm in detail and conclude with some results produced by it.

Structure from Motion Problem

Shree K. Nayar
Columbia University

Topic: Structure from Motion, Module: Reconstruction II
First Principles of Computer Vision

4

Feature Detection and Tracking

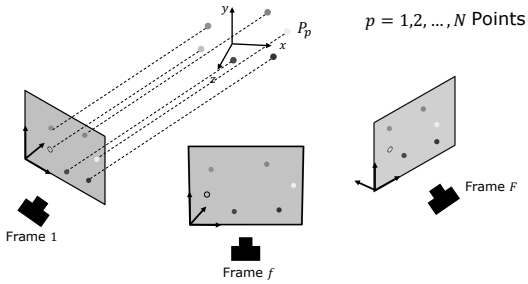
- Detect feature points: Corners, SIFT points, …
- Track feature points: Template Matching, Optical Flow…

5

Let us take a closer look at the structure from motion problem. Our input is a video of a scene, which is a sequence of frames. Shown on the right is the first frame of an input video. We apply feature detection to this frame and the detected features are overlaid as black dots. These features could be corners, SIFT features, or any type of interest points. We track these features through the entire video using any one of the methods we have discussed—template matching, optical flow, or comparing SIFT descriptors. Eventually, we end up with a set of features that are tracked through the entire video sequence. The image coordinates of these tracked features are the input to the structure from motion algorithm.

Shown here is a world coordinate frame and a set of 3D scene points denoted by P_p. These points represent the 3D structure we wish to recover. Multiple images (frames) of this structure are captured, each frame giving us 2D image projections of the 3D scene points. We assume that we have a total of N scene points and F frames. The 2D image coordinates of the scene points can then be denoted as $(u_{f,p}, v_{f,p})$, where f denotes the frame number and p denotes the point in the scene.

Orthographic Structure from Motion



Given sets of corresponding image points (2D): $(u_{f,p}, v_{f,p})$
Find scene points (3D) P_p, assuming orthographic camera.

[Tomasi 1992]

6

Given the set of image coordinates $(u_{f,p}, v_{f,p})$, we wish to find the 3D coordinates P_p of the scene points. This is the general formulation of the structure from motion problem. To make this problem tractable, Tomasi and Kanade used the simplifying assumption that the camera is orthographic.

In the case of an orthographic camera, we assume that the range of depths in the scene is small in comparison to the distance of the scene from the camera. Thus, the magnification of the camera is the same for all scene points and through the entire sequence of captured images. Under this camera model, in a given image, all scene points can be assumed to map to the image plane using parallel rays that are perpendicular to the image plane.

This simplifying assumption of an orthographic camera makes the structure from motion problem easier to formulate and solve. While this assumption was used in the early work of Tomasi and Kanade, it has been replaced with the more general perspective projection model in subsequent work.

We will now take all our tracked image coordinates, $(u_{f,p}, v_{f,p})$, and organize them into a single matrix called the observation matrix.

Observation Matrix

Shree K. Nayar

Columbia University

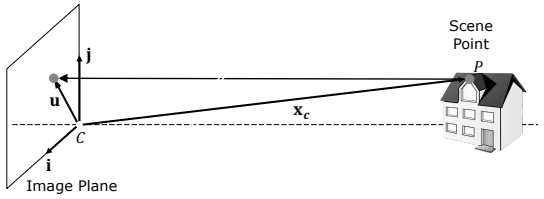
Topic: Structure from Motion, Module: Reconstruction II

First Principles of Computer Vision

7

Let us first look at how a 3D scene point is mapped to its 2D image coordinates in the case of orthographic projection. Shown here is a scene point P and a camera coordinate frame C. Unlike perspective projection, where the camera coordinate frame is placed at the pinhole of the camera, in this case, we will place it at one of the corners of the image. C is defined by two unit vectors, $\mathbf{i}$ and $\mathbf{j}$, which are aligned along the two edges of the image plane. We can now project the scene point P, which is represented in the camera coordinate frame by the vector $\mathbf{x}_c$, by using a ray parallel to the optical axis (dotted line) to obtain the image point $\mathbf{u} = (u, v)$, where u is the dot product between $\mathbf{x}_c$ and $\mathbf{i}$, and v is the dot product between $\mathbf{x}_c$ and $\mathbf{j}$. These dot products can be equivalently represented as $\mathbf{i}^T$ times $\mathbf{x}_c$ and $\mathbf{j}^T$ times $\mathbf{x}_c$, respectively [1].

From 3D to 2D: Orthographic Projection



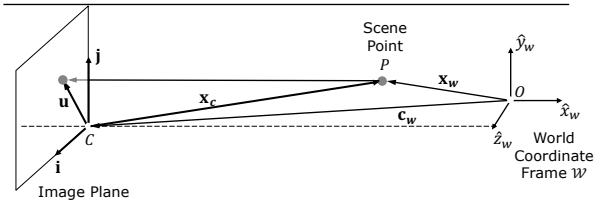
$$u = \mathbf{i} \cdot \mathbf{x}_c = \mathbf{i}^T \mathbf{x}_c$$
$$v = \mathbf{j} \cdot \mathbf{x}_c = \mathbf{j}^T \mathbf{x}_c \quad [1]$$

Perspective cameras exhibit orthographic projection when distance of scene from camera is large compared to depth variation within scene (magnification is nearly constant).

8

Ultimately, we wish to recover the 3D coordinates of each scene point P in the world coordinate frame $\mathcal{W}$. To this end, we want to relate our image coordinates (u, v) to the world coordinates $\mathbf{x}_w$ of P. Let the location of the camera in the world coordinate frame be $\mathbf{c}_w$. Then, in the world coordinate frame, the vector $\mathbf{x}_c$ is simply $(\mathbf{x}_w - \mathbf{c}_w)$. Substituting for $\mathbf{x}_c$ in the equations in slide 8, we get the two equations in 1. To simplify our notations, we will use P and C to denote the 3D coordinates of $\mathbf{x}_w$ and $\mathbf{c}_w$, respectively.

From 3D to 2D: Orthographic Projection



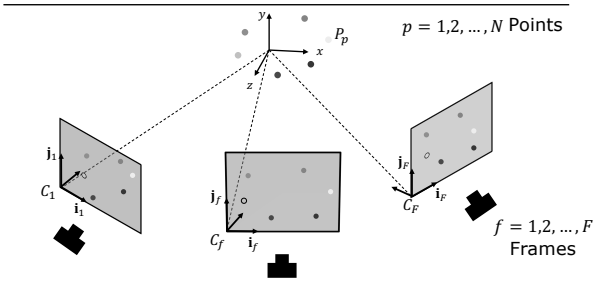
$$u = \mathbf{i}^T\mathbf{x}_c = \mathbf{i}^T(\mathbf{x}_w - \mathbf{c}_w) = \mathbf{i}^T(P - C)$$
$$v = \mathbf{j}^T\mathbf{x}_c = \mathbf{j}^T(\mathbf{x}_w - \mathbf{c}_w) = \mathbf{j}^T(P - C) \quad \boxed{1}$$

$$\boxed{\begin{aligned} u &= \mathbf{i}^T(P - C) \\ v &= \mathbf{j}^T(P - C) \end{aligned}}$$

9

The orthographic structure from motion problem can be stated as follows: given the 2D image points $(u_{f,p}, v_{f,p})$, where f denotes the frame number and p denotes the scene point, we wish to find the 3D coordinates P_p of each scene point p. Unfortunately, the camera positions C_f and their orientations $\mathbf{i}_f$ and $\mathbf{j}_f$ are also unknown.

Orthographic SFM



Given corresponding image points (2D) $(u_{f,p}, v_{f,p})$

Find scene points $\{P_p\}$.

Camera Positions $\{C_f\}$, camera orientations $\{(\mathbf{i}_f, \mathbf{j}_f)\}$ are unknown.

10

To account for the fact that we have a sequence of frames, we can rewrite the two equations in slide 9 as shown here (see 1). In these equations, everything on the right side of the equations—P_p, C_f, $\mathbf{i}_f$, and $\mathbf{j}_f$— is unknown. This is what makes the structure from motion problem challenging. In order to make it tractable, we need to reduce the number of unknowns. This brings us to the centering trick.

Orthographic SFM

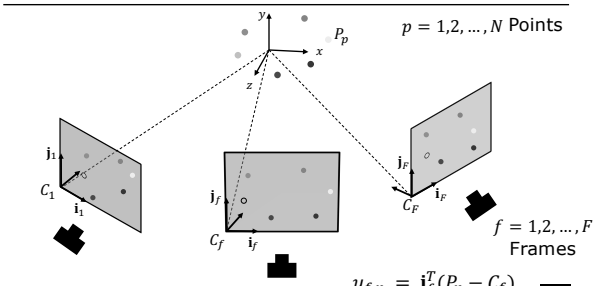


Image of point P_p in camera frame f:

$$\begin{aligned} u_{f,p} &= \mathbf{i}_f^T(P_p - C_f) \\ v_{f,p} &= \mathbf{j}_f^T(P_p - C_f) \end{aligned} \quad \boxed{1}$$

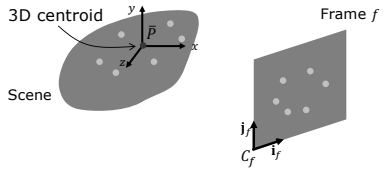
Known | Unknown

We can remove C_f from equations to simply SFM problem.

11

The centering trick allows us to remove an unknown, namely C_f, from the right-hand side of the equations in slide 11. Here is a set of 3D points that we project orthographically to the 2D points in the image f. We will assume that the origin of the world coordinate frame, which we are free to place anywhere we wish, lies at the centroid $\bar{P}$ of the 3D scene points that we are trying to estimate. $\bar{P}$ is simply the average of the 3D coordinates of all the N scene points P_p [1]. This implies that when we ultimately recover the 3D coordinates of the scene points P_p, they will be with respect to the centroid $\bar{P}$.

Centering Trick



Assume origin of world at centroid of scene points:

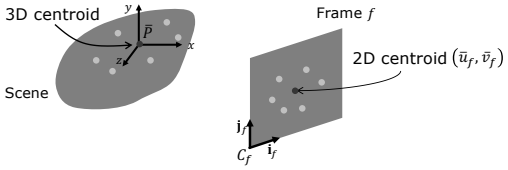
$$\frac{1}{N}\sum_{p=1}^{N} P_p = \bar{P} = \mathbf{0} \quad [1]$$

We will compute scene points w.r.t their centroid!

12

Let us now look at the centroid $(\bar{u}_f, \bar{v}_f)$ of the image coordinates of the scene points P_p. $\bar{u}_f$ is the average of all the $u_{f,p}$ coordinates in frame f. By substituting the $\mathbf{i}_f^T$ times P_p minus C_f for $u_{f,p}$ (from slide 11), we get expression [1]. Upon expanding [1], we get equation [2], where the first term includes the sum of all the scene points P_p. This term is equal to zero since we set the origin of the world coordinate frame to be the centroid of the scene points. The second term in [2] is an average computed over $\mathbf{i}_f^T$ times C_f, which is a constant. Thus, $\bar{u}_f$ is equal to the negative of $\mathbf{i}_f^T$ times C_f [3]. We get an analogous expression for $\bar{v}_f$. We see that now $\bar{u}_f$ and $\bar{v}_f$ are independent of the 3D scene points P_p.

Centering Trick



Centroid $(\bar{u}_f, \bar{v}_f)$ of the image points in frame f:

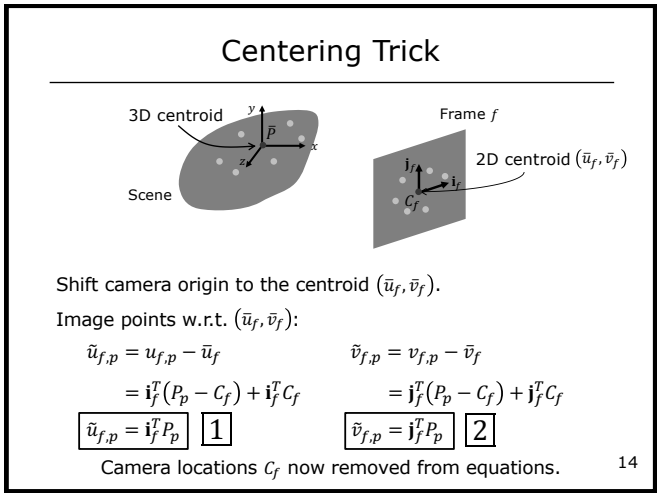
$$\bar{u}_f = \frac{1}{N}\sum_{p=1}^{N} u_{f,p} = \frac{1}{N}\sum_{p=1}^{N} \mathbf{i}_f^T(P_p - C_f) \quad [1] \qquad \bar{v}_f = \frac{1}{N}\sum_{p=1}^{N} v_{f,p} = \frac{1}{N}\sum_{p=1}^{N} \mathbf{j}_f^T(P_p - C_f)$$

$$\bar{u}_f = \frac{1}{N}\mathbf{i}_f^T\sum_{p=1}^{N} P_p - \frac{1}{N}\sum_{p=1}^{N} \mathbf{i}_f^T C_f \quad [2] \qquad \bar{v}_f = \frac{1}{N}\mathbf{j}_f^T\sum_{p=1}^{N} P_p - \frac{1}{N}\sum_{p=1}^{N} \mathbf{j}_f^T C_f$$

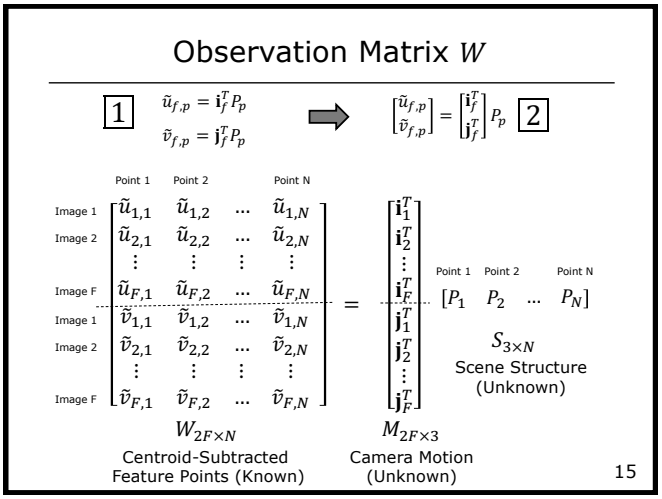
$$\bar{u}_f = -\mathbf{i}_f^T C_f \quad [3] \qquad \bar{v}_f = -\mathbf{j}_f^T C_f$$

13

Next, we shift the origin C_f of the camera to the centroid $(\bar{u}_f, \bar{v}_f)$, as shown here. We can now define all our image coordinates in this new coordinate frame, which we refer to as the centroid-subtracted coordinates $(\tilde{u}_{f,p}, \tilde{v}_{f,p})$. The expression for $\tilde{u}_{f,p}$ is simply to $\mathbf{i}_f^T$ times P_p, since the terms with C_f get cancelled out [1]. Similarly, the expression for $\tilde{v}_{f,p}$ simplifies to $\mathbf{j}_f^T$ times P_p [2]. Thus, the image coordinates of the scene points no longer have the camera center C_f in their expressions and only have the camera orientation, which is given by the two vectors $\mathbf{i}_f$ and $\mathbf{j}_f$.



We now have two expressions for each scene point in a particular camera frame [1]. These two expressions can be written in vector form as shown in [2]. We have N points in each of the F frames and all their expressions can be organized into a single equation in matrix form, as shown here. W on the left-hand side is a $2F \times N$ matrix called the observation matrix. Its first F rows consist of all the centroid-subtracted u values of the N points in each of the F frames. The next F rows consist of all the centroid-subtracted v values.



The right-hand side includes a matrix consisting the unit vectors **i** and **j** corresponding to the orientations of the camera in the F frames. This is called the camera motion matrix M and it is a $2F \times 3$ matrix. M is multiplied by the scene structure matrix S which is a $3 \times N$ matrix made up of the 3D coordinates of the N scene points. Therefore, the observation matrix W is the product of the camera motion matrix M and scene structure matrix S. Note that while W is known, the motion matrix M and structure matrix S are unknown.

So, the question is whether we can find the structure matrix S and the motion matrix M from the observation matrix W? It turns out that this is possible because W has the property that it is low in rank.

Observation Matrix W

$$\begin{matrix} \\ \text{Image 1} \\ \text{Image 2} \\ \vdots \\ \text{Image F} \\ \text{Image 1} \\ \text{Image 2} \\ \vdots \\ \text{Image F} \end{matrix} \begin{matrix} \begin{matrix} \text{Point 1} & \text{Point 2} & & \text{Point N} \end{matrix} \\ \begin{bmatrix} \tilde{u}_{1,1} & \tilde{u}_{1,2} & \dots & \tilde{u}_{1,N} \\ \tilde{u}_{2,1} & \tilde{u}_{2,2} & \dots & \tilde{u}_{2,N} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{u}_{F,1} & \tilde{u}_{F,2} & \dots & \tilde{u}_{F,N} \\ \tilde{v}_{1,1} & \tilde{v}_{1,2} & \dots & \tilde{v}_{1,N} \\ \tilde{v}_{2,1} & \tilde{v}_{2,2} & \dots & \tilde{v}_{2,N} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{v}_{F,1} & \tilde{v}_{F,2} & \dots & \tilde{v}_{F,N} \end{bmatrix} \\ W_{2F\times N} \end{matrix} = \begin{matrix} \begin{bmatrix} \mathbf{i}_1^T \\ \mathbf{i}_2^T \\ \vdots \\ \mathbf{i}_F^T \\ \mathbf{j}_1^T \\ \mathbf{j}_2^T \\ \vdots \\ \mathbf{j}_F^T \end{bmatrix} \\ M_{2F\times 3} \end{matrix} \begin{matrix} \begin{matrix} \text{Point 1} & \text{Point 2} & & \text{Point N} \end{matrix} \\ \begin{bmatrix} P_1 & P_2 & \dots & P_N \end{bmatrix} \\ S_{3\times N} \end{matrix}$$

Centroid-Subtracted Feature Points (Known) | Camera Motion (Unknown) | Scene Structure (Unknown)

Can we find M and S from W?

16

Before we discuss the rank of the observation matrix W, let us review the concept of the rank of a matrix.

Rank of Observation Matrix

Shree K. Nayar
Columbia University

Topic: Structure from Motion, Module: Reconstruction II
First Principles of Computer Vision

17

First, let us discuss the notion of linear independence. A set of vectors, $\mathbf{v}_1$ through $\mathbf{v}_n$, is said to be linearly independent if no vector in the set can be represented as a weighted sum of the other vectors in the set. Shown here are five vectors: $\mathbf{i}$, $\mathbf{j}$, $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$, in a two-dimensional space. Vectors $\mathbf{i}$ and $\mathbf{j}$ are linearly independent because they are orthogonal, and hence we cannot express $\mathbf{i}$ as a scaled version of $\mathbf{j}$. Vectors $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{v}_1$ are linearly dependent since we can represent any one of them as a linear combination of the other two. The same applies to the set of vectors $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{v}_3$ and the set $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$.

Linear Independence of Vectors

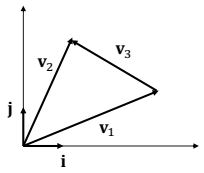
A set of vectors $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$ is said to be linearly independent if no vector can be represented as a weighted linear sum of the others.

$\{\mathbf{i}, \mathbf{j}\}$ is linearly independent.

$\{\mathbf{i}, \mathbf{j}, \mathbf{v}_1\}$ is linearly dependent.

$\{\mathbf{i}, \mathbf{j}, \mathbf{v}_3\}$ is linearly dependent.

$\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ is linearly dependent.



MATH PRIMER

18

The concept of linear dependence and independence can be used to determine the rank of a matrix. Consider a matrix A with m rows and n columns. We can view A as a concatenation of n column vectors, $\mathbf{c}_1$ through $\mathbf{c}_n$, or m row vectors, $\mathbf{r}_1$ through $\mathbf{r}_m$.

A matrix has two types of ranks: column rank and row rank. The column rank of a matrix is the number of linearly independent columns of the matrix, and the row rank of a matrix is the number of linearly independent rows of the matrix.

Rank of a Matrix

Column Rank: The number of linearly independent columns of the matrix.

Row Rank: The number of linearly independent rows of the matrix.

$$\underset{m}{\underset{n}{\begin{bmatrix} \\ A \\ \\ \end{bmatrix}}} = [\mathbf{c}_1 \quad \mathbf{c}_2 \quad \cdots \quad \mathbf{c}_n] = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \vdots \\ \mathbf{r}_m^T \end{bmatrix}$$

$$ColumnRank(A) \leq n \qquad RowRank(A) \leq m$$

$$\boxed{\begin{array}{c} ColumnRank(A) = RowRank(A) = Rank(A) \\ Rank(A) \leq \min(m, n) \end{array}}$$

MATH PRIMER 19

The column rank of A must be less than or equal to n because A has only n columns, and the row rank of A must be less than or equal to m because it has only m rows. For any matrix, the column rank is equal to the row rank, which is simply called the rank of the matrix. By this definition, the rank of A is less than or equal to the minimum of m and n.

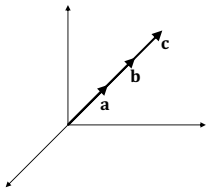
Let us now visualize the geometric meaning of the rank of a matrix by considering a 3×3 matrix A made up of the three column vectors $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$. The rank of A can be viewed as the dimensionality of the space spanned by its column or row vectors. If the column vectors $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$ are collinear (i.e., they lie on the same line), then $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$ can be expressed as a scaled version of a single vector. That is, only a single vector, or dimension, is needed to span the space defined by the three vectors. In this case, the rank of A is one.

Geometric Meaning of Matrix Rank

Rank is the dimensionality of the space spanned by column or row vectors of the matrix.

$$A = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix} = [\mathbf{a} \quad \mathbf{b} \quad \mathbf{c}]$$

$$Rank(A) = 1 \quad \Longleftrightarrow$$



MATH PRIMER 20

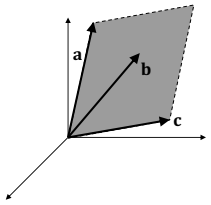
In this example, the three vectors **a**, **b** and **c** lie on a two-dimensional plane. We know that just two vectors on this plane are sufficient to represent all other vectors on the plane. Therefore, in this case, the rank of A is two.

Geometric Meaning of Matrix Rank

Rank is the dimensionality of the space spanned by column or row vectors of the matrix.

$$A = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix} = [\mathbf{a} \quad \mathbf{b} \quad \mathbf{c}]$$

$$Rank(A) = 2 \iff$$



MATH PRIMER 21

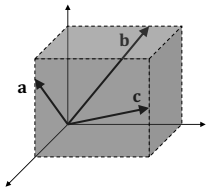
Here we have three arbitrary vectors, **a**, **b** and **c**, in a three-dimensional space. All three vectors are needed to express any arbitrary vector in this space. Therefore, the rank of A in this case is three. With the above examples, we now have a geometric intuition for the rank of a matrix.

Geometric Meaning of Matrix Rank

Rank is the dimensionality of the space spanned by column or row vectors of the matrix.

$$A = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix} = [\mathbf{a} \quad \mathbf{b} \quad \mathbf{c}]$$

$$Rank(A) = 3 \iff$$



MATH PRIMER 22

Here are the key properties of the rank of a matrix. First, the rank of A^T is equal to the rank of A. This is a consequence of the fact that the column rank is equal to the row rank. Second, the rank of the product of two matrices, A and B, is the minimum of the rank of matrix A and the rank of matrix B. If A is $m \times n$ and B is $n \times p$, the rank of the product of A and B must be less than or equal to the minimum of m, n and p. From the above two properties, we can show that the rank of AA^T is equal to the rank of A^TA, which is equal to the rank of A. Finally, if A is an $m \times m$ matrix that is invertible, then its rank is m, and we call it a full rank matrix.

Important Properties of Matrix Rank

- $Rank(A^T) = Rank(A)$
- $Rank(A_{m\times n}B_{n\times p}) \leq \min\left(Rank(A_{m\times n}), Rank(B_{n\times p})\right)$
  $\leq \min(m, n, p)$
- $Rank(AA^T) = Rank(A^TA) = Rank(A^T) = Rank(A)$
- $A_{m\times m}$ is invertible iff $Rank(A_{m\times m}) = m$ (Full Rank)

MATH PRIMER 23

Let us now return to our observation matrix W, which is composed of centroid-subtracted image coordinates of the scene points. W is a $2F \times N$ matrix, where F is the number of frames and N is the number of scene points. W is equal to the product of the motion matrix M and the structure matrix S. Our goal is to compute M and S from W.

…Back to Observation Matrix W

$$\begin{matrix} \\ \text{Image 1} \\ \text{Image 2} \\ \\ \text{Image F} \\ \text{Image 1} \\ \text{Image 2} \\ \\ \text{Image F} \end{matrix} \begin{matrix} \text{Point 1} \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix} \begin{bmatrix} \tilde{u}_{1,1} & \tilde{u}_{1,2} & \dots & \tilde{u}_{1,N} \\ \tilde{u}_{2,1} & \tilde{u}_{2,2} & \dots & \tilde{u}_{2,N} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{u}_{F,1} & \tilde{u}_{F,2} & \dots & \tilde{u}_{F,N} \\ \tilde{v}_{1,1} & \tilde{v}_{1,2} & \dots & \tilde{v}_{1,N} \\ \tilde{v}_{2,1} & \tilde{v}_{2,2} & \dots & \tilde{v}_{2,N} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{v}_{F,1} & \tilde{v}_{F,2} & \dots & \tilde{v}_{F,N} \end{bmatrix} \begin{matrix} \text{Point 2} & & \text{Point N} \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \end{matrix}$$

$W_{2F\times N}$
Centroid-Subtracted Feature Points (Known)

$$= \begin{bmatrix} \mathbf{i}_1^T \\ \mathbf{i}_2^T \\ \vdots \\ \mathbf{i}_F^T \\ \mathbf{j}_1^T \\ \mathbf{j}_2^T \\ \vdots \\ \mathbf{j}_F^T \end{bmatrix}$$

$M_{2F\times 3}$
Camera Motion (Unknown)

$$\begin{matrix} \text{Point 1} & \text{Point 2} & & \text{Point N} \end{matrix}$$
$$[P_1 \quad P_2 \quad \dots \quad P_N]$$

$S_{3\times N}$
Scene Structure (Unknown)

24

We know that the rank of M must be less than or equal to the minimum of $2F$ and 3. Similarly, the rank of S must be less than or equal to the minimum of 3 and N. Since W is the product of M and S, its rank must be less than or equal to the minimum of 3, N and $2F$. Since the number of scene points, N, and the number of camera frames, F, can both be expected to be large, we can conclude that the rank of W is less than or equal to 3. We will use this fact to develop our structure from motion algorithm.

Rank of Observation Matrix

$$\underset{2F\times N}{W} = \underset{2F\times 3}{M} \times \underset{3\times N}{S}$$

We know:

$$Rank(MS) \leq Rank(M) \qquad Rank(MS) \leq Rank(S)$$

$$\Rightarrow \quad Rank(MS) \leq \min(3, 2F) \qquad Rank(MS) \leq \min(3, N)$$

$$\Rightarrow \quad Rank(W) \leq Rank(MS) \leq \min(3, N, 2F)$$

Rank Theorem: $Rank(W) \leq 3$

[Tomasi 1992] 25

The structure from motion algorithm we present here was proposed by Tomasi and Kanade and is referred to as the factorization method.

Tomasi-Kanade Factorization

Shree K. Nayar
Columbia University

Topic: Structure from Motion, Module: Reconstruction II
First Principles of Computer Vision

26

We know that the rank of the observation matrix W must be less than or equal to 3. This rank constraint makes it possible for us to factorize the observation matrix W into its two components M and S. That is, from just the observed image coordinates we can determine both the motion of the camera as well as the structure of the scene.

Rank of Observation Matrix

$$\underset{2F\times N}{W} = \underset{2F\times 3}{M} \times \underset{3\times N}{S}$$

We know:

$Rank(MS) \leq Rank(M)$ $\quad$ $Rank(MS) \leq Rank(S)$

⇨ $Rank(MS) \leq \min(3, 2F)$ $\quad$ $Rank(MS) \leq \min(3, N)$

⇨ $Rank(W) = Rank(MS) \leq \min(3, N, 2F)$

Rank Theorem: $Rank(W) \leq 3$

We can "factorize" W into M and S!

[Tomasi 1992] 27

To factorize the observation matrix W, we will use a popular method in linear algebra called singular value decomposition (SVD). It should be noted that SVD can be applied to any $M \times N$ matrix A; it does not need to be our observation matrix W. The SVD of matrix A is the product of three matrices: U, Σ and V^T. Matrices U and V are orthonormal, which is a concept we have discussed before. The matrix Σ is a diagonal matrix and is important to us. It consists of all the singular values σ_i along its diagonal in decreasing order of value, with σ_1 being the largest and most important singular value, σ_2 being the next largest, and so on. All these singular values are non-negative. But what do these singular values really mean? From expression 1, it can be seen that the largest singular value σ_1 would end up being multiplied with the first column of matrix U and the first row of matrix V^T. Hence, this column and row make the largest contribution to the reconstruction of the matrix A. In other words, they are the most important column and row of U and V^T, respectively. This is one way to think about singular value decomposition.

Singular Value Decomposition (SVD)

For any matrix A there exists a factorization:

$$A_{M\times N} = U_{M\times M} \cdot \Sigma_{M\times N} \cdot {V^T}_{N\times N} \quad 1$$

where U and V^T are orthonormal and Σ is diagonal.

MATLAB: `[U,S,V] = svd(A)`

$$\Sigma_{M\times N} = \begin{pmatrix} \sigma_1 & 0 & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & 0 & 0 & \dots & 0 \\ 0 & 0 & \sigma_3 & 0 & \dots & 0 \\ 0 & 0 & 0 & \sigma_4 & \dots & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \dots & \sigma_N \\ 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \end{pmatrix}$$

$\sigma_1, \dots, \sigma_N$: Singular Values

If $Rank(A) = r$ then A has r non-zero singular values.

MATH PRIMER 28

Similarly, since σ_2 is the next most important singular value, the corresponding column and row of U and V^T, respectively, are the next most important. The columns of U are referred to as the left singular vectors of the matrix A and the columns of V are referred to as the right singular vectors of A. The most important aspect of SVD in the context of structure from motion is that if the rank of matrix A happens to be r, then only the first r singular values in Σ are non-zero.

Upon applying SVD to the observation matrix W, we get the three matrices, U, Σ and V^T, where U is a $2F \times 2F$ matrix, V^T is a $N \times N$ matrix, and Σ is a $2F \times N$ matrix.

Enforcing Rank Constraint

Using SVD: $W = U \Sigma V^T$

$$= \begin{bmatrix} & & \\ & U & \\ & & \end{bmatrix}_{2F \times 2F} \begin{bmatrix} \sigma_1 & 0 & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & 0 & 0 & \dots & 0 \\ 0 & 0 & \sigma_3 & 0 & \dots & 0 \\ 0 & 0 & 0 & \sigma_4 & \dots & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \dots & \sigma_N \\ 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \end{bmatrix}_{2F \times N} \begin{bmatrix} & & \\ & V^T & \\ & & \end{bmatrix}_{N \times N}$$

Where: $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \cdots \geq \sigma_N$ are the singular values of W

29

We know that the rank of W must be less than or equal to three. Therefore, W can have at most three non-zero singular values.

Enforcing Rank Constraint

Using SVD: $W = U \Sigma V^T$

$$= \begin{bmatrix} & & \\ & U & \\ & & \end{bmatrix}_{2F \times 2F} \begin{bmatrix} \sigma_1 & 0 & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & 0 & 0 & \dots & 0 \\ 0 & 0 & \sigma_3 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \end{bmatrix}_{2F \times N} \begin{bmatrix} & & \\ & V^T & \\ & & \end{bmatrix}_{N \times N}$$

Since $Rank(W) \leq 3$, $Rank(\Sigma) \leq 3$.

All except first 3 diagonal elements of Σ must be 0.

30

Let us focus on the shaded submatrices of U, Σ and V^T shown here. Since only the shaded 3×3 submatrix in Σ has non-zero values, the submatrix U_2 of matrix U makes no contribution to the observation matrix W. Similarly, submatrix V_2^T of matrix V^T makes no contribution to W. This is because all the values in U_2 and V_2^T are multiplied by zeros since all the corresponding singular values are zero. Only the submatrices U_1 and V_1^T end up contributing to W. Even though the observation matrix W is massive, and after applying SVD we get three large matrices U, Σ and V^T, only small parts of these large matrices have any bearing on W.

Enforcing Rank Constraint

Using SVD: $W = U \Sigma V^T$

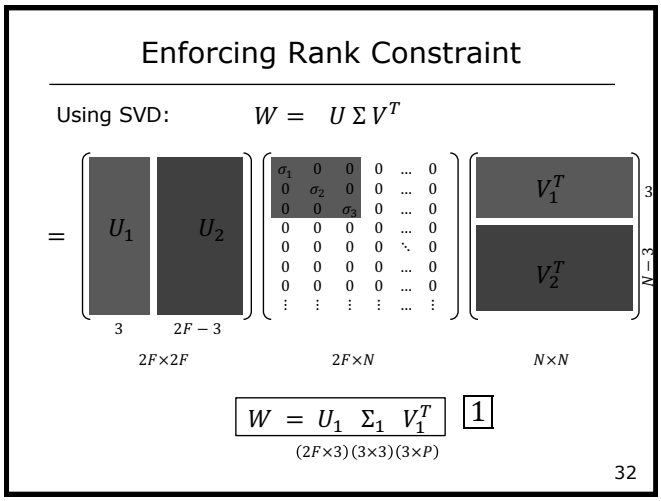
$$= \begin{bmatrix} U_1 & U_2 \\ 3 & 2F-3 \end{bmatrix}_{2F \times 2F} \begin{bmatrix} \sigma_1 & 0 & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & 0 & 0 & \dots & 0 \\ 0 & 0 & \sigma_3 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \end{bmatrix}_{2F \times N} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}_{N \times N} \begin{matrix} 3 \\ N-3 \end{matrix}$$

Since $Rank(W) \leq 3$, $Rank(\Sigma) \leq 3$.

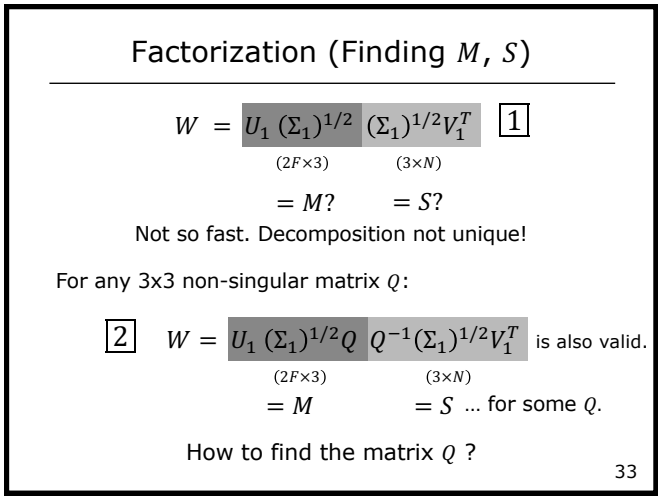
Submatrices U_2 and V_2^T do not contribute to W.

31

We therefore have a more economical representation of W, given by [1]. Note that this representation is not an approximation but in fact exact. In summary, the $2F \times N$ observation matrix W can be decomposed into three matrices: a $2F \times 3$ matrix U_1, a 3×3 matrix Σ_1, and a $3 \times N$ matrix V_1^T.

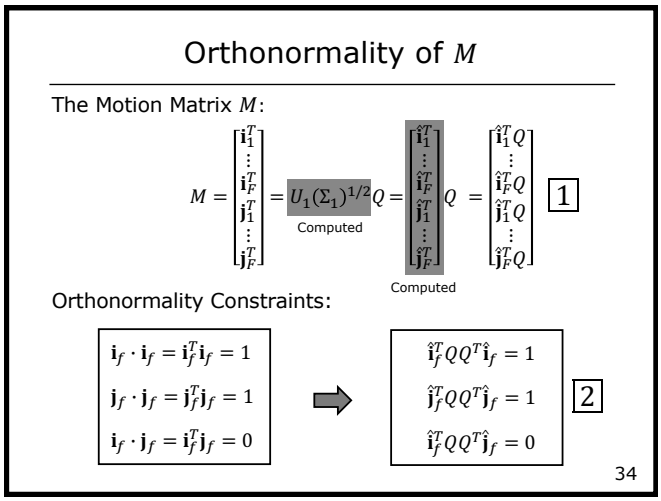


We have decomposed W into U_1, Σ_1 and V_1^T. But how do we factorize W into the motion matrix M and the structure matrix S? One way of factorizing W would be to split up Σ_1 into two equal components where each component is $\sqrt{\Sigma_1}$, as shown in [1]. This is a completely valid factorization into a $2F \times 3$ matrix and a $3 \times N$ matrix, but there is no reason why these factorized matrices should correspond to a valid motion matrix M and a valid structure matrix S.



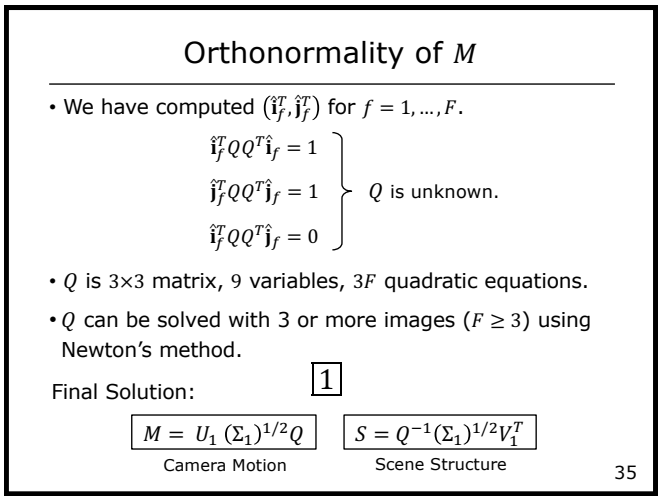
Let us now post-multiply the first factor $U_1(\Sigma_1)^{1/2}$ with some 3×3 matrix Q, and pre-multiply the second factor $(\Sigma_1)^{1/2}V_1^T$ with Q^{-1}, as shown in [2]. The matrix Q will have no impact on the observation matrix W but it will change both of the factors. Hence, we will pose the structure from motion problem as finding the 3×3 matrix Q that would give us a valid motion matrix M and a valid structure matrix S. In order to find this matrix Q, we need to invoke some additional information.

It turns out that there is an important constraint that we have not yet used, which is the orthonormality of the camera motion matrix M. Recall that M is made of the camera orientation vectors, **i**'s and **j**'s, of all the F camera frames. M can also be expressed as $U_1(\Sigma_1)^{1/2}Q$, where $U_1(\Sigma_1)^{1/2}$ (shaded matrix) has already been computed and the 3×3 matrix Q is unknown, as shown in 1.



Now, we know that the camera orientation vectors $\mathbf{i}_f$ and $\mathbf{j}_f$ are unit vectors and are orthogonal to each other. Thus, for any frame f, $\boldsymbol{i}_f$ times $\boldsymbol{i}_f$ is equal to one, $\boldsymbol{j}_f$ times $\boldsymbol{j}_f$ is equal to one, and $\boldsymbol{i}_f$ times $\boldsymbol{j}_f$ is equal to zero. These three orthonormality constraints can be rewritten as shown in 2, where $\hat{\mathbf{i}}_f$ and $\hat{\mathbf{j}}_f$ are known (see 1), and the only unknown is the matrix Q. Note that we get the three constraints given by 2 for each camera frame.

Since we have a total of F camera frames, we get $3F$ equations and 9 unknowns, namely, the elements of the matrix Q. As long as we have three or more frames ($F \geq 3$), which we almost always do, then we have enough equations to solve for the 9 unknowns. Note that the $3F$ equations we have are quadratic in the unknown elements of Q. Hence, a solution can be obtained using the Newton's method.



The structure from motion is solved once we have found Q because we can plug it into the expressions 1 for the motion matrix M and the structure matrix S.

Let us summarize the above structure from motion algorithm. First, we capture a video of a scene while moving around it. Next, we detect and track feature points through all the frames of the video. Then, we centroid-subtract the image coordinates of the feature points. We then construct the observation matrix W using the centroid-subtracted coordinates. We apply singular value decomposition to W and enforce the rank constraint to obtain an economical representation of *W*, i.e., $U_1\ \Sigma_1\ V_1^T$.

Summary: Orthographic SFM

1. Detect and track feature points.
2. Create the centroid subtracted matrix W of corresponding feature points.
3. Compute SVD of W and enforce rank constraint.

$$W = \underset{(2F\times 3)}{U}\ \underset{(3\times 3)}{\Sigma}\ \underset{(3\times P)}{V^T} = U_1\ \Sigma_1\ V_1^T$$

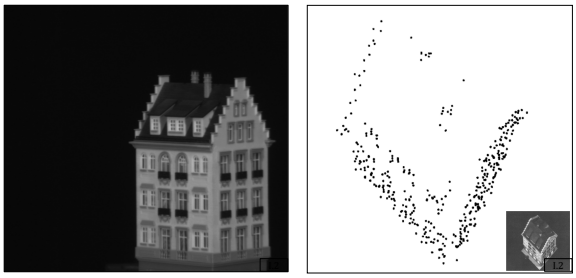
4. Set $M = U_1\ (\Sigma_1)^{1/2} Q$ and $S = Q^{-1}(\Sigma_1)^{1/2} V_1^T$.
5. Find Q by enforcing the orthonormality constraint.

[Tomasi 1992] 36

Next, we split Σ_1 into two equal factors and insert an unknown 3×3 matrix Q, so that the camera motion matrix M is $U_1(\Sigma_1)^{1/2}Q$ and scene structure matrix S is $Q^{-1}(\Sigma_1)^{1/2}V_1^T$. The only unknown here is Q, which is solved for by using the orthonormality of the camera orientation vectors. We then use Q to find the motion matrix M and the structure matrix S.

Shown here are some of the first factorization results obtained by Tomasi in the late 1980s. From an orthographic video of the house on the left, the 3D coordinates of points on the house were estimated and are shown on the right. The inset image on the right is a different perspective of the house that was rendered using its recovered 3D structure.

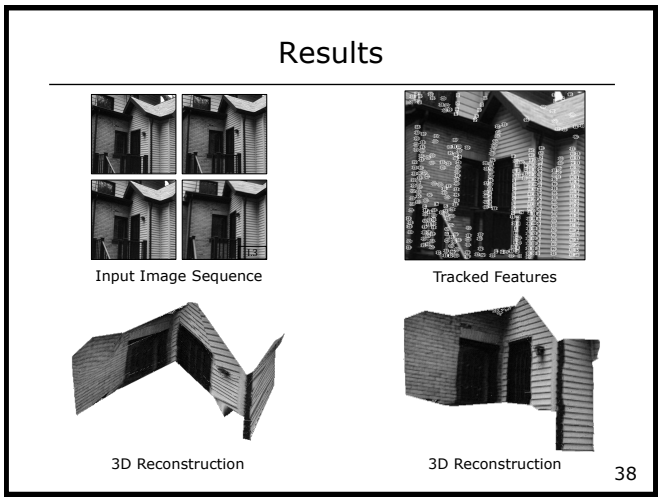
Results



Input Image Sequence Estimated 3D Points

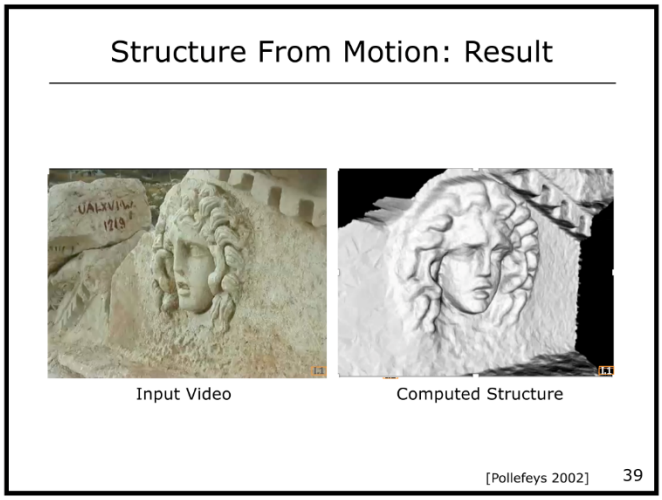
37

In this example, a sequence of images was taken of an outdoor scene. Some of the tracked features are shown on the top-right (white marks). The estimated structure was used to render the two new perspectives shown at the bottom.



These are essentially the first results of structure from motion using the factorization method. Since then, many extensions have been made to the algorithm to make it more widely applicable. For instance, the state-of-the art algorithms do not rely on the orthographic camera assumption and can work on perspective videos that even include zooming, i.e., variation in focal length. Note that we assumed that every feature is visible through the entire video. Recent algorithms can handle features that may appear and disappear during the capture of the video.

Shown here is a video demonstration of a more recent structure from motion algorithm developed by Marc Pollefeys. From a casual handheld video of the artifact seen here (see online lecture video), features were tracked, and then the 3D structure (depth map) shown on the right as well as the camera motion during the capture were computed. This depth map can be texture mapped using the captured video and then rendered from any perspective.



References and Credits

Shree K. Nayar

Columbia University

Topic: Structure from Motion, Module: Reconstruction II

First Principles of Computer Vision

40

References: Textbooks

Computer Vision: A Modern Approach
Forsyth, D and Ponce, J., Prentice Hall

Multiple View Geometry
Hartley, R. and Zisserman, A., Cambridge University Press

Computer Vision: Algorithms and Applications
Szeliski, R., Springer

An introduction to 3D Computer Vision
Cyganek, B., Siebert, J. P., Wiley Pub

41

References: Papers

[Tomasi 1992] Tomasi, C. and Kanade, T. "Shape and Motion from Image Streams under Orthography: A Factorization Method" IJCV, 1992.

[Pollefeys 2002] M. Pollefeys and L. Van Gool. Visual modeling: from images to images, The Journal of Visualization and Computer Animation, 13: 199-209, 2002.

42

Image Credits

I.1 Marc Pollefeys. Used with permission.

I.2 Carlo Tomasi. Used with permission.

I.3 Carlo Tomasi. Used with permission.

43

Acknowledgements: Thanks to Joel Salzman, Nikhil Nanda and Ayush Sharma for their help with transcription, editing and proofreading**.**

References

[Szeliski 2022] Computer Vision: Algorithms and Applications, Szeliski, R., Springer, 2022.

[Forsyth and Ponce 2003] Computer Vision: A Modern Approach, Forsyth, D. and Ponce, J., Prentice Hall, 2003.

[Horn 1986] Robot Vision, Horn, B. K. P., MIT Press, 1986.

[Hartley and Zisserman 2000] Multiple View Geometry, Hartley, R. and Zisserman, A., Cambridge University Press, 2000.

[Cyganek and Siebert 2009] An introduction to 3D Computer Vision, Cyganek, B. and Siebert, J. P., Wiley, 2009.

[Tomasi 1992] Shape and Motion from Image Streams under Orthography: A Factorization Method, Tomasi, C. and Kanade, T., IJCV, 1992.

[Pollefeys 2002] Visual modeling: from images to images, ePollefeys, M. and Van Gool, L., The Journal of Visualization and Computer Animation, 13: 199-209, 2002.

[Nayar 2022B] Image Formation, Nayar, S. K., Monograph FPCV-1-1, First Principles of Computer Vision, Columbia University, New York, February 2022.

[Nayar 2022E] Image Processing I, Nayar, S. K., Monograph FPCV-1-4, First Principles of Computer Vision, Columbia University, New York, March 2022.

[Nayar 2022F] Image Processing II, Nayar, S. K., Monograph FPCV-1-5, First Principles of Computer Vision, Columbia University, New York, March 2022.

[Nayar 2022I] SIFT Detector, Nayar, S. K., Monograph FPCV-2-3, First Principles of Computer Vision, Columbia University, New York, August 2022.

[Nayar 2025H] Camera Calibration, Nayar, S. K., Monograph FPCV-4-1, First Principles of Computer Vision, Columbia University, New York, April 2025.

[Nayar 2025I] Uncalibrated Stereo, Nayar, S. K., Monograph FPCV-4-2, First Principles of Computer Vision, Columbia University, New York, April 2025.

[Nayar 2025J] Optical Flow, Nayar, S. K., Monograph FPCV-4-3, First Principles of Computer Vision, Columbia University, New York, April 2025.