Chapter 5: Reconstruction from Two Views

Konrad Koniarski

2016/07/19

1 The Reconstruction Problem

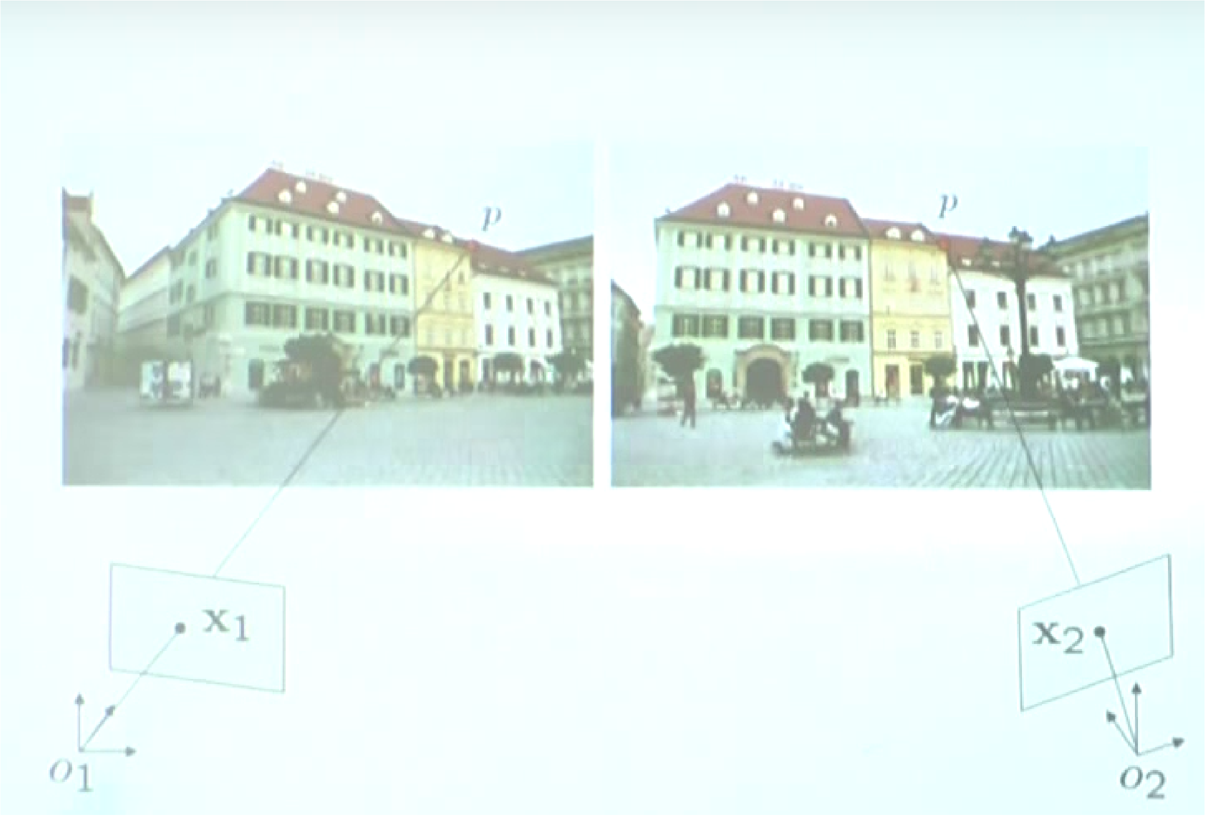
1.1 Problem Formulation

In the last section, we discussed how to idenfify point correspondences between two consecutive frames. In this section, we will tackle the next problem, namely that of **reconstructing the 3D geometry of cameras and points**. To this end, we will make the following assumptions:

- We assume that we are given a **set of corresponding points** in two frames taken with the same camera from different vantage points.
- We assume that the **scene is static**, i.e. none of the observed 3D points moved during the camera motion.
- We also assume that the **intrinsic camera (calibration) parameters are known**.

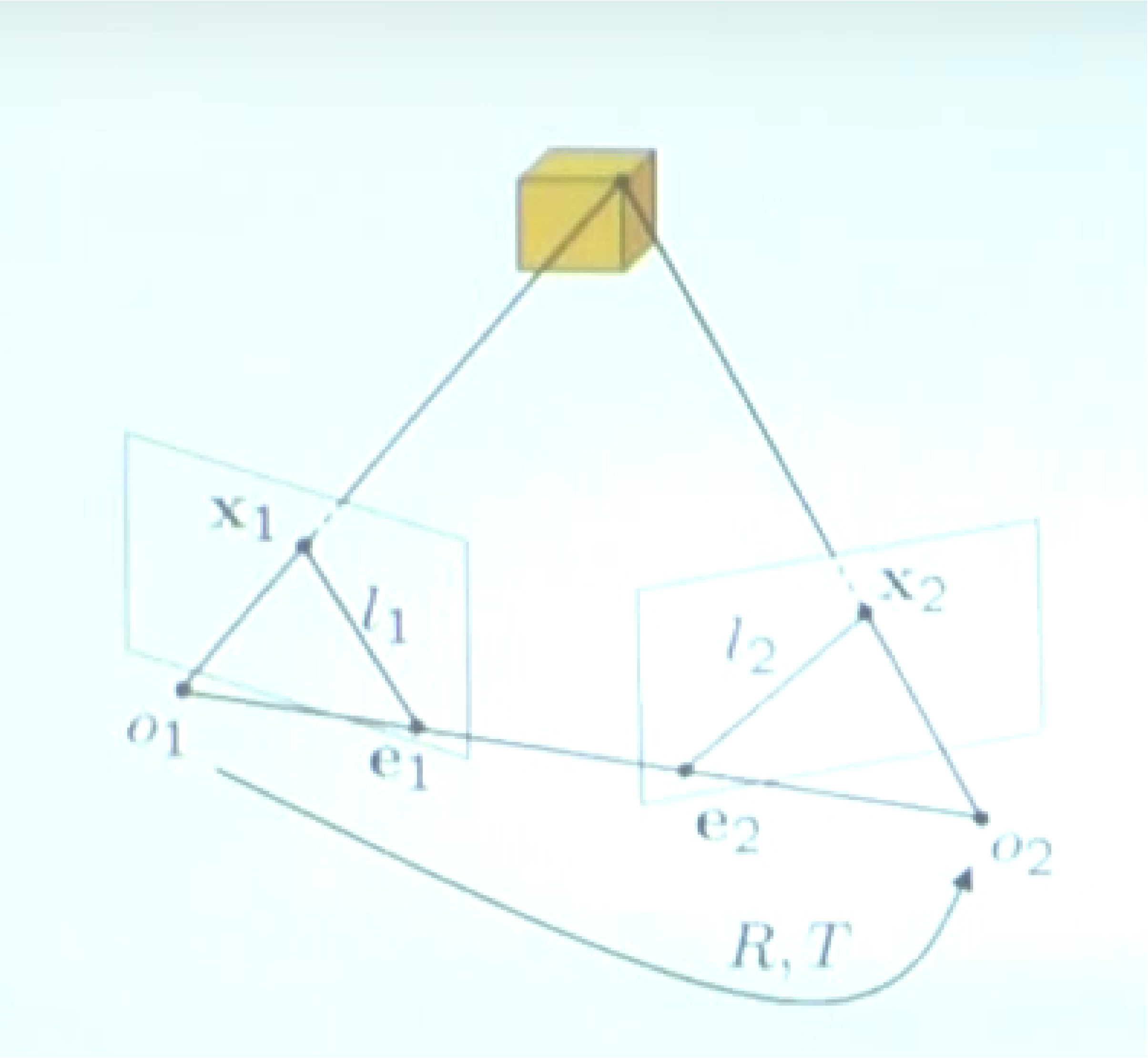
We will first estimage the **camera motion** from the set of corresponding points. Once we know the relative location and orientation of the camera, we can reconstruct the 3D location of all corresponding points by **triangulation**.

1.2 Problem Formulation Example



Goal: Estimage camera motion and 3D scene structure from two views.

1.3 Epipolar Geometry: Some Notation



The projections of a point X onto the two images are denoted by x_1 and x_2. The **optical centers** of each camera are denoted by o_1 and o_2. The intersections of the line (o_1, o_2) with each image planes are called the **epipoles** e_1 and e_2. The intersections between the **epipolar plane** (o_1, o_2, X) and the image planes are called **epipolar lines** I_1 and I_2. There is one epipolar plane for each 3D point X.

1.4 The Reconstruction Problem

In general 3D reconstruction is a challeging problem. If we are given two views with 100 feature points in each of them, then we have 200 point coordinates in 2D. The goal is to estimage

- 6 parameters modeling the camera motion R, T and
- 100 × 3 coordinates for the 3D points X_j

This could be done by minimizing the **projection error**:

$$E(R, T, X_1, ..., X_{100}) = \sum_j ||x_1^j - \pi(X_j)||^2 + ||x_2^j - \pi(R, T, X_j)||^2 \tag{1}$$

This amounts to a **difficult optimization problem** called **bundle adjustment**. It turns out that there is a more elegant solution which allows to entirely get rid of the 3D coordinates.

2 The Epipolar Constraint

2.1 The Epipolar Constraint

We know that x_1 (in homogeneous coordinates) is the projection of a 3D point X. Given known camera parameters ($K = 1$) and no rotation or translation of the first camera, we merely have a projection with unknown depth λ_1. From the first to the second frame we additionally have a camera rotation R and translation T followed by a projection. This gives the equations:

$$\lambda_1 x_1 = X, \lambda_2 x_2 = RX + T \tag{2}$$

Insetting the first equation into the second, we get:

$$\lambda_2 x_2 = R(\lambda_1 x_1) + T \tag{3}$$

Now we remove the translation by multiplying with $\hat{T}(\hat{T}v \equiv T \times v)$:

$$\lambda_2 \hat{T} x_2 = \lambda_1 \hat{T} R x_1 \tag{4}$$

And projection onto x_2 gives the **epipolar constraint**:

$$\boxed{x_2^T \hat{T} R x_1 = 0} \tag{5}$$

2.2 The Epipolar Constraint

The epipolar constraint

$$\boxed{x_2^T \hat{T} R x_1 = 0} \tag{6}$$

provides a relation between the 2D point coordinates of a 3D point in each of the two images and the camera transformation parameters. The original 3D point coordinates have been removed. The matrix

$$e = \hat{T} R \in \mathbb{R}^{3 \times 3} \tag{7}$$

is called the esential matrix. The epipolar constraint is alsow known as essential constraint or bilinear constraint. Geometrically, this constraint states that the three vectors $o_1\vec{X}$, $o_2\vec{o}_1$ and $o_2\vec{X}$ form a plane, i.e the triple product of these vectors (measuring the volume of the parallelepiped) is zero: In coordinates of the second frame Rx_1 gives the direction of the vector $o_1\vec{X}$; T gives the direction of $o_2\vec{o}_1$, and $\vec{x}_2$ is proportional to the vector $x_2\vec{X}$ such that

$$volume = x_2^T (T \times R x_1) = 0 \tag{8}$$

2.3 Properties of Essential Matrix E

The space of all essential matrices is called the **essential space**:

$$\varepsilon \equiv \{\hat{T}R | R \in SO(3), T \in \mathbb{R}^3\} \subset \mathbb{R}^{3\times 3} \tag{9}$$

Theorm (Huang & Faugeras, 1989) Characterization of the essential matrix: A none zero matrix $E \in \mathbb{R}^{3\times 3}$ is an essential matrix if and only if E has a singular value decomposition (SVD) $E = U\Sigma V^T$ with

$$\Sigma = diag(\sigma, \sigma, 0) \tag{10}$$

for some $\sigma > 0$ and U, $V \in SO(3)$.

Theorem (Pose recovery from the essential matrix): There exist exactly two relative poses (R, T) with $R \in SO(3)$ and $T \in \mathbb{R}^3$ corresponding to an essential matrix $E \in \varepsilon$. For $E = U\Sigma V^T$ we have:

$$(\hat{T}_1, R_1) = (UR_z(+\frac{\pi}{2}))\Sigma U^T, UR_z^T(+\frac{\pi}{2}V^T) \tag{11}$$

$$(\hat{T}_2, R_2) = (UR_z(-\frac{\pi}{2}))\Sigma U^T, UR_z^T(-\frac{\pi}{2}V^T) \tag{12}$$

In genral, only one of these gives meaningful (positive) depth

3 Eight Point Algorithm

3.1 A Basic Reconstruction Algorithm

We have seen that the 2D-coordinates of each 3D point are coupled to the camera parameters R and T through and epipolar constraint. In the following , we will derive a 3D reconstruction algorithm which proceeds as follow:

- **Recover the essential matrix E**: from the epipolar constraints associated with a set of point pairs.
- **Extract the relative translation and rotation** from the essential matrix E.

In general, the matrix E recovered from a set of epipolar constraints will not be an essential matrix. One can resolve this problem it two ways:

- Recover some matrix $E \in \mathbb{R} \times \mathbb{R}$ from the epipolar constraints and then project it onto the essential space.
- Optimize the epipolar constraints in the essential space.

While the second approach is a principle more accurate it involves a nonlinear constrained optimization. So we will pursue the first approach which is simpler and faster.

3.2 The Eight-Point Linear Algorithm

First we rewrite the epipolar constraint as a scalar product in the elements of the matrix E and the coordinates of the points x_1 and x_2. Let

$$E^s = (e_{11}, e_{21}, e_{31}, e_{12}, e_{22}, e_{32}, e_{13}, e_{23}, e_{33}) \in \mathbb{R}^9 \tag{13}$$

be the vector of elements of E and

$$a \equiv x_1 \otimes x_2 \tag{14}$$

the Kronecker product of the vectors $x_j \equiv (x_j, y_j, z_j)$ defined as

$$a = (x_1x_2, x_1y_2, x_1z_2, y_1x_2, y_1y_2, y_1z_2, z_1x_2, z_1y_2, z_1z_2)^T \in \mathbb{R}^9 \tag{15}$$

Then the epipolar constraint can be written as

$$x_2^T E x_1 = a^T E^s = 0 \tag{16}$$

For n point pairs, we can combine this into the linear system:

$$\boxed{\chi E^s = 0, \text{ with } \chi = (a_1, a_2, ..., a_n)^T} \tag{17}$$

3.3 The Eight-Point Linear Algorithm

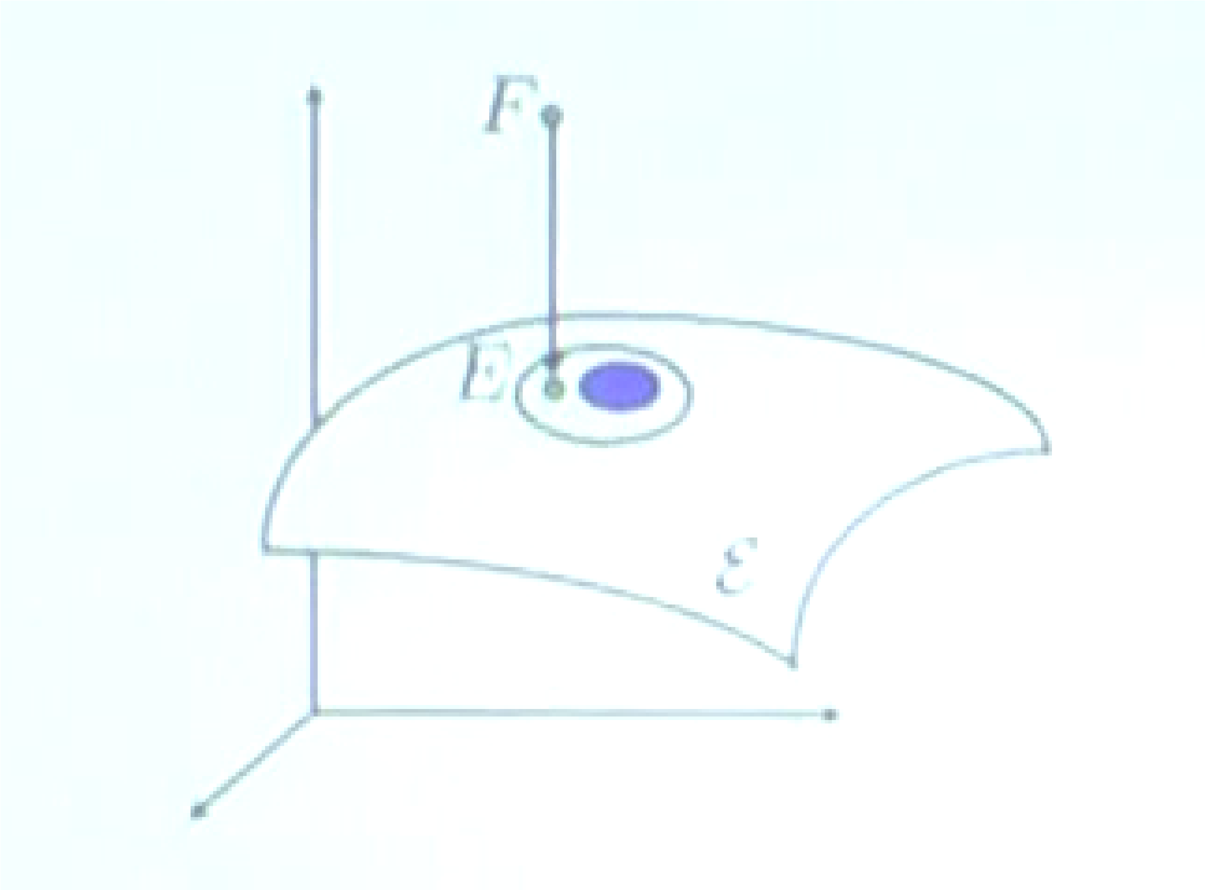
According to

$$\chi E^s = 0, \text{ with } \chi = (a_1, a_2, ..., a_n)^T \tag{18}$$

we see that the vector of coefficients of the essential matrix E defines the null space of the matrix χ. In order for the above system to have a unique solution (up to a scaling factor and ruling out the trivial solution $E = 0$). the rank of the matrix χ needs to be exactly 8. Therefor we need at least 8 point pairs. In certain degenerate cases, the solution for the essential matrix is not unique even if we have 8 or more point pairs. One such example is the case that all points lie on a line or on a plane. Clearly, we will not be able to revcover the sigh of E. Since with each E, there are two possible assignments of rotation R and translation T, we therefore end up with four possible solutions for rotation and translation.

3.4 Projection onto Essential Space

The numerically estimated coefficients E^s will in general not correspond to an essential matrix. One can resolve this problem by projecting it back to the essential space.



Theorem (Projection onto essential space): Let $F \in \mathbb{R}^{3\times3}$ be an arbitrary matrix with SVD.

$F = Udiag\{\lambda_1, \lambda_2, \lambda_3\}V^T, \lambda_1 \leq \lambda_2 \leq \lambda_3$. Then the essential matrix E which minimizes the Frobenius norm $||F - E||_f^2$ is given by

$$E = Udiag\{\sigma, \sigma, 0\}V^T, \text{ with } \sigma = \frac{\lambda_1 + \lambda_2}{2} \tag{19}$$

3.5 Eight Point Algorithm (Longuet-Higgins '81)

Given a set of $n = 8$ or more point pairs x_1^i, x_2^i :

- **Compute an approximation of the essential matrix**: Construct the matrix $\chi = (a^1, a^2, ...a^n)^T$. Where $a^j = x_1^j \otimes x_2^j$. Find the vector $E^s \in \mathbb{R}^9$ which minimizes $||\chi E^s||$ as the ninth column of V_χ in the $SVD_\chi = U\chi\Sigma_\chi V_\chi^T$. Unstack E^s into 3×3-matrix E.
- **Project onto essential space**. Compute the $SVDE = Udiag\{\sigma_1, \sigma_2, \sigma_3\}V^T$. Since in the reconstruction, E is only defined up to a scalar, we project E onto the normalized essential space by replacing the singular values σ_1, σ_2, σ_3 with 1,1,0.
- **Recover the displacement from the essential matrix**. The four possible solutions for rotation and translation are:

$$R = UR_Z^T(\pm\frac{\pi}{2})V^T, \hat{T} = UR_Z^T(\pm\frac{\pi}{2})\Sigma U^T \tag{20}$$

with a rotation by $\pm\frac{\pi}{2}$ around z:

$$R_Z^T(\pm\frac{\pi}{2}) = \begin{pmatrix} 0 & \pm 1 & 0 \\ \mp 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{21}$$

3.6 Do We Need Eight Points?

The above reasoning showed that we need at least eight points in order for the matrix χ to have rank 8 and therefore guarantee a unique solution for E. Yet, one can take into account the special structure of E. **The space of essential matrices is actually a five dimensional space**, i.e. E only has 5 (and not 9) degree of freedom. A simple way to **take into account the algebraic properties of E** is to make use of the fact that $detE = 0$. If now we have only 7 point pairs, the null space of χ will have (at least) Dimension 2, spanned by two vectors E_1 and E_2. Then we can solve for E by determining α such that:

$$detE = det(E_1 + \alpha E_2) = 0 \tag{22}$$

Along similar lines, **Kruppa proved in 1913 that one needs only five point pairs to recover** (R, T). In the case of degenerate motion (for example planar of circular motion), one can resolve the problem with even fewer point pairs.

3.7 Limitation and Further Extensions

Among the four possible solutions for R and T, there is generally **only one meaningful one** (which assigns positive depth to all points).

The algorithm fails if the translations is exactly 0, since then $E = 0$ and noting can be recovered. Due to noise this typically does not happen.

In the case of infinitesimal view point change, one can adapt the eight point algorithm to the **continuous motion case**, where the epipolar constraint is replaced by the **continuous epipolar constraint**. Rather than recovering (R, T) one recovers the linear and angular velocity of the camera.

In the case of independently moving objects, one can generalize the epipolar eonstraint. For two motions for example we have:

$$(x_2^T E_1 x_1)(x_2^T E_2 x_1) = 0 \tag{23}$$

with two essential matrices E_1 and E_2. Given a sufficiently large number of point pairs, one can solve the respective equtions for multiple essential matrices using polynomial factorization.

4 Structure Reconstruction

4.1 Structure Reconstruction

The linear eight-point algorithm allows us to estmate the camera transformation parameters R and T from a set of corresponding point pairs. Yet, the essential matrix E and hence the translation T are **only defined up to an arbitrary scale** $||E|| = ||T|| = \xi \in \mathbb{R}^+$. After recovering R and T, we therefore have for point X^i:

$$\lambda_2^j x_2^j = \lambda_1^j R x_1^j + \gamma T, j = 1, ..., n \tag{24}$$

with unknown scale parameters λ_i^j. We can eliminate one of these scales by applying $\hat{x_2^j}$:

$$\lambda_1^j \hat{x_2^j} R x_1^j + \gamma \hat{x_2^j} T = 0, j = 1, ..., n \tag{25}$$

This cooresponds to n linear system of the form

$$() \begin{pmatrix} \lambda_1^j \\ \xi \end{pmatrix} = 0, j = 1, ..., n \tag{26}$$

4.2 Structure Reconstruction

Combining the parameters $\vec{\lambda} = (\lambda_1^1, \lambda_1^2, ..., \lambda_1^n, \gamma)^T \in \mathbb{R}^{n+1}$, we get the linear equation system

$$\boxed{M\vec{\lambda}} = 0 \tag{27}$$

with

$$M \equiv \begin{pmatrix} \hat{x_2^1} R x_1^1 & 0 & \vdots & 0 & 0 & \hat{x_2^1} T \\ 0 & \hat{x_2^2} R x_1^2 & \vdots & 0 & 0 & \hat{x_2^2} T \\ \dots & \dots & \ddots & \dots & \dots & \dots \\ 0 & 0 & \vdots & \hat{x_2^{n-1}} R x_1^{n-1} & 0 & \hat{x_2^{n-1}} T \\ 0 & 0 & \vdots & 0 & \hat{x_2^n} R x_1^n & \hat{x_2^n} T \end{pmatrix} \tag{28}$$

The linear least squares estimation for $\vec{\lambda}$ is given by the eigenvector corresponding to the smallest eigenvalues of $M^T M$. It is **only defined up to a global scale**. It reflect the **ambiguity** hat the camera has moved twice the distance, the scene is twice larger and twice as far away.

5 Bundle Adjustment

5.1 Optimality in Noisy Real World Conditions

The eight-point algorithm discussed before has several nice properties. In particular, we found **closed-form solutions** to estimate the camera parameters and the 3D structure, based on **singular value decomposition**. However, if we have noisy data $\widetilde{x}_1$, $\widetilde{x}_2$ (correspondences not exact or even incorrect), then we have:

- **no guarantee that R and T are as close as possible to the true solution.**
- **no guarantee that we will get a consistent reconstruction.**

  [IMAGE 9/17 28:12]

5.2 Nonlinear Optimization Methods

In order to take noise and statistical fluctuation into account, one can revert to a **Bayesian formulation** and determine the most likely camera transformation R, T and 'true' 2D coordinates x fiven the measured coordinates $\widetilde{x}$, by performing a **maximum aposteriori estimage**:

$$arg \max_{x,R,T} \mathcal{P}(x, R, T|\widetilde{x}) = arg \max_{x,R,T} \mathcal{P}(\widetilde{x}|x, R, T)\mathcal{P}(x, R, T) \tag{29}$$

This approach will however involve modeling probability densities $\mathcal{P}$ on the fairly complicated space $SO(3) \times \mathbb{S}^2$ of rotation and translation parameters, as $R \in SO(3)$ and $T \in \mathbb{S}^2$ (3D translation with unit length).

Alternatively, one can perform a **constrained optimization** by minimizing a cost function (similarity to measurements):

$$\phi(x, R, T) = \sum_{j=1}^{n} \sum_{i=1}^{2} ||\widetilde{x}_i^j - x_i^j||^2 \tag{30}$$

subject to (consistent geometry):

$$x_2^{jT} \hat{T} R x_1^j = 0, x_1^{jT} e_3 = 1, x_2^{jT} e_3 = 1, j = 1, 2, ..., n. \tag{31}$$

5.3 Bundle Adjustment

Interestingly, the unknown depth parameters λ_i **do not actually appear in the above cost functions**.

The depth parameters appear directly in the **unconstrained optimization problem**:

$$\sum_{j=1}^{n} ||\widetilde{x}_1^j - \pi_1(X^j)||^2 + ||\widetilde{x}_2^j - \pi_2(X^j)||^2 \tag{32}$$

where π_j denote the projections onto the two images. Expressed in coordinates of the first camera frame, this is equal to the cost function:

$$\phi(x_1, R, T, \lambda) = \sum_{j=1}^{n} ||\widetilde{x}_1^j - x_1^j||^2 + ||\widetilde{x}_2^j - \pi(R\lambda_1^j x_1 + T)||^2 \tag{33}$$

This optimization procedure is known as **bundle adjestment**. The constrained optimization and the unconstrained bundle adjustment can be seen as **different parametrization of the same optimization objective**.

5.4 Degenerate Configurations

The eight-point algorithm only provides unique solutions (up to a scalar factor) if all 3D points are in a "general position". this is no longer the case for certain **degenerate configurations**, for which all points lie on certain 2D surfaces which are called **critical surfaces**.

Typically these critical surfaces are described by a quadratic equations in the three point coordinates, such that they are referred to as **quadratic surfaces**.

While most critical configurations do not actually arise in practice, a specific degenerate configuration which does arise often is the case that **all points lie on a 2D plane** (such as floors, table, walls, ...).

For the structure-from-motion problem in the context of points on a plane, one can exploit additional constraints which leads to the so-called **four-point algorithm**.

6 Four-Point Algorithm

6.1 Planar Homegraphies

Let us assume that all points lie on a plane. If $X_1 \in \mathbb{R}^3$ denotes the point coordinates in the first frame, and these lie on a plane with normal $N \in \mathbb{S}^2$, then we have:

$$N^T X_1 = d \Leftrightarrow \frac{1}{d} N^T X_1 = 1 \tag{34}$$

In frame two, we therefore have the coordinates:

$$X_2 = RX_1 + T = RX_1 + T\frac{1}{d}N^T X_1 = (R + \frac{1}{d}TN^T)X_1 \equiv HX_1 \tag{35}$$

where

$$H = R + \frac{1}{d}TN^T \in \mathbb{R}^{3\times 3} \tag{36}$$

is called a **homography matrix**. Inserting the 2D coordinates, we get:

$$\lambda_2 x_2 = H\lambda_1 x_1 \Leftrightarrow \boxed{x_2 \sim Hx_1} \tag{37}$$

where $\sim$ means equality up to scaling. This expression is called a **planar homography**. H depends on camera and plane parameters.

6.2 From Point Pairs to Homography

For a pait of corresponding 2D points we therefore have

$$\lambda_2 x_2 = H\lambda_1 x_1 \tag{38}$$

By multiplying with $\widehat{x_2}$ we can eliminate λ_2 and obtain:

$$\widehat{x_2} H x_1 = 0 \tag{39}$$

This equation is called the **planar epipolar constraint** or **planar homography constraint**. Again, we can cast this equation into the form

$$a^T H^s = 0 \tag{40}$$

where we have stacked the elements of H into a vector

$$H^s = (H_{11}, H_{21}, ..., H_{33}) \in \mathbb{R}^9 \tag{41}$$

and introduced the matrix

$$a \equiv x_1 \otimes \widehat{x_2} \in \mathbb{R}^{9\times 3} \tag{42}$$

6.3 The Four Point Algorithm

Let us now assume we have $n \geq 4$ pairs of corresponding 2D points $\{x_1^j, x_2^j\}$, $j = 1, ..., n$ in the two images. Each point pair induces a matrix a^j, we integrate these into a larger matrix

$$\chi \equiv (a^1, ..., a^n)^T \in \mathbb{R}^{3n \times 9}, \tag{43}$$

and obtain the system

$$\chi H^s = 0 \tag{44}$$

As in the case of the essential matrix, **the homography matrix can be estimated up to a scale factor**.

This gives rise to the **four point algorithm**:

- For the point pairs, compute the matrix χ.
- Compute a solution H^s for the above equation by singular value decomposition of χ.
- Extract the motion parameters from the homography matrix $H = R + \frac{1}{d} T N^T$

6.4 General Comments

Clearly, the derivation of the **four-point algorithm** is in close analogy to that of the **eight-point algorithm**.

Rather then estimating the **essential matrix** E one estimates the **homography matrix** H to derive R and T. In the four-point algorithm, the **homography matrix is decomposed into** R**,** N **and** T/d. In other words, one can reconstruct the normal of the plane, but the translation is only obtained in units of the offset d of the plane and the origin.

The **3D structure of the points** can then be computer in the same manner as before.

Since one uses the strong constraint that all points lie in a plane, the **four-point algorithm only requires four correspondences**.

There exist **numerous relations** between the essential matrix $E = \hat{T}R$ and corresponding homography matrix $H = R + Tu^T$ with some $u \in \mathbb{R}^3$, in particular:

$$E = \hat{T}H, H^T E + E^T H = 0 \tag{45}$$

7 The Uncalibrated Case

7.1 The Case of an Uncalibrated Camera

The reconstruction algorithms introduced above all assume that the camera is calibrated (K = 1). The general transformation from a 3D point to the iamge is given by:

$$\lambda x' = K \Pi_0 g X = (KR, KT) X \tag{46}$$

with the **intrinsic parameter matrix** on **calibration matrix**:

$$K = \begin{pmatrix} fs_x & s_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{pmatrix} \in \mathbb{R}^{3\times3} \tag{47}$$

The calibration matrix maps metric coordinates into image (pixel) coordinates, using the focal length f, the optical center o_x, o_y, the pixel size s_x, x_y and a skew factor s_θ. If these parameters are known then one can simply **transform the pixel coordinates x' to normalized coordinates** $x = K^{-1}x'$ to obtain the representation used in the previous sections. This amounts to centering the coordinates with respect to the optical center etc.

7.2 The Fundamental Matrix

If the camera parameters K cannot be stimated in calibration procedure beforehand, then one has to deal with reconstruction from uncalibrated views.

By transforming all images coordinates x' with the inverse calibration matrix K^{-1} into metric coordinates x, we obtain the epipolar constraint for uncalibrated cameras:

$$x_2^T \hat{T} R x_1 = 0 \Leftrightarrow x_2^{'T} K^{-T} \hat{T} R K^{-1} x_1' = 0 \tag{48}$$

which can be written as

$$\boxed{x_2^{'T} F x_1' = 0} \tag{49}$$

with the fundamental matrix defined as:

$$\boxed{F \equiv K^{-T} \hat{T} R K^{-1} = K^{-T} E K^{-1}} \tag{50}$$

Since the invertible matrix K does not affect the rank of this matrix, we know that F has an SVD F= $U\Sigma V^T$ with $\Sigma = diag(\sigma_1, \sigma_2, 0)$. In fact, **any matrix of rank 2 can be a fundamental matrix**.

7.3 Limitations

While it is straight-forward to extend the eight-point algorithm, such that one can extract a **fundamental matrix**, from a set of corresponding image points, it is less straight forward how to proceed from there.

Firstly, one cannot impose a strong constraint on the specific structure of the fundamental matrix (apart from the fast that the last singular value is zero).

Secondly, for a given fundamental matrix F, there does not exist a finite number of decompositions into extransic parameters R, T and intrinsic parameters K (even apart from the global scale factor).

As a consequence, one can only determine so-called **projective reconstructions**, i.e. reconstructions of geometry and camera position which are defined up to a so-called projective transformation.

As a solution, one typically choses a **canonical reconstruction** from the family of possible reconstructions.