AUTO-CALIBRATION.
FACTORORIZATION.
STRUCTURE FROM MOTION.
Thank you for the slides.
They come mostly from the following sources.

Marc Pollefeys U. of North Carolina
Martial Hebert CMU
Silvio Savarese U. of Michigan
Dan Huttenlocher Cornell U.
Motivation

• Avoid explicit calibration procedure
  • Complex procedure
  • Need for calibration object
  • Need to maintain calibration
Factorization

• Factorise observations in structure of the scene and motion/calibration of the camera

• Use all points in all images at the same time

✓ Affine factorisation non-iterative least square
✓ Projective factorisation iterative least square
Affine camera

The affine projection equations are

\[
\begin{bmatrix}
  x_{ij} \\
  y_{ij} \\
  1
\end{bmatrix} =
\begin{bmatrix}
  P_i^x \\
  P_i^y \\
  0001
\end{bmatrix}
\begin{bmatrix}
  X_j \\
  Y_j \\
  Z_j \\
  1
\end{bmatrix}
\]

\[
\begin{bmatrix}
  x_{ij} \\
  y_{ij} \\
  1
\end{bmatrix} =
\begin{bmatrix}
  P_i^x \\
  P_i^y \\
  0001
\end{bmatrix}
\begin{bmatrix}
  X_j \\
  Y_j \\
  Z_j \\
  1
\end{bmatrix}
\]

Affine Factorization Algorithm


inhomogeneous world and image coordinates

Translation

\[
x_{ij} = 1/n \sum_{i=1}^{n} x_{ij}
\]

\[
y_{ij} = x_{ij} - t_i
\]

\[
\tilde{x}_{ij} = P_i^x
\]

\[
\tilde{y}_{ij} = P_i^y
\]

\[
\begin{bmatrix}
  x_{ij} - P_i^x \\
  y_{ij} - P_i^y \\
\end{bmatrix} =
\begin{bmatrix}
  \tilde{x}_{ij} \\
  \tilde{y}_{ij}
\end{bmatrix}
\begin{bmatrix}
  \tilde{P}_i^x \\
  \tilde{P}_i^y \\
\end{bmatrix}
\begin{bmatrix}
  X_j \\
  Y_j \\
  Z_j
\end{bmatrix}
\]

centroid of 3D points maps to the centroid of projections
Orthographic factorization
(Tomasi Kanade’92)

The orthographic projection equations are

\[
\mathbf{m}_{ij} = \mathbf{P}_i \mathbf{M}_j, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n
\]

where

\[
\mathbf{m}_{ij} = \begin{bmatrix} \tilde{x}_{ij} \\ \tilde{y}_{ij} \end{bmatrix}, \quad \mathbf{P}_i = \begin{bmatrix} \bar{P}_{ix} \\ \bar{P}_{iy} \end{bmatrix}, \quad \mathbf{M}_j = \begin{bmatrix} X_j \\ Y_j \\ Z_j \end{bmatrix}
\]

All equations can be collected for all \(i\) and \(j\)

\[
\begin{bmatrix}
\mathbf{m}_{11} & \mathbf{m}_{12} & \cdots & \mathbf{m}_{1n} \\
\mathbf{m}_{21} & \mathbf{m}_{22} & \cdots & \mathbf{m}_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\mathbf{m}_{m1} & \mathbf{m}_{m2} & \cdots & \mathbf{m}_{mn}
\end{bmatrix}
= \begin{bmatrix}
\mathbf{P}_1 \\
\mathbf{P}_2 \\
\vdots \\
\mathbf{P}_m
\end{bmatrix}
= \mathbf{P}
= \begin{bmatrix}
\mathbf{M}_1 \\
\mathbf{M}_2 \\
\vdots \\
\mathbf{M}_n
\end{bmatrix}
\]

Note that \(\mathbf{P}\) and \(\mathbf{M}\) are resp. 2\(m\times3\) and 3\(x\)\(n\) matrices and therefore the rank of \(\mathbf{m}\) is at most 3.
Orthographic factorization

Factorize $\mathbf{m}$ through singular value decomposition

$$\mathbf{m} = \mathbf{U} \Sigma \mathbf{V}^T$$

An affine reconstruction is obtained as follows

$$\mathbf{\tilde{P}} = \mathbf{U}, \mathbf{\tilde{M}} = \Sigma \mathbf{V}^T$$

or $$\mathbf{P}^* = \mathbf{U} \Sigma \mathbf{M}^* = \mathbf{V}^T$$

Closest rank-3 approximation yields MLE!

$$\min \left\| \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1n} \\ m_{21} & m_{22} & \cdots & m_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{m1} & m_{m2} & \cdots & m_{mn} \end{bmatrix} - \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_m \end{bmatrix} \begin{bmatrix} \mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n \end{bmatrix} \right\|$$
A metric reconstruction is obtained

\[ P_- = P \sim A^{-1} \quad M_- = A M \sim \quad A \text{ a } 3 \times 3 \text{ matrix} \]

The matrix \( A \) can be obtained by self-calibration or knowing the image of absolute conic and doing Cholesky factorization

\[ A A^T = (P_{13}^T \omega_{1 \ldots m} P_{13})^{-1} \quad \text{or for } i = 1, \ldots, m \]

\[ i^T_i A A^T i_i = 1 \]
\[ j^T_i A A^T j_i = 1 \]
\[ i^T_i A A^T j_i = 0 \]

First RANSAC (or similar) to eliminate outliers.
Not all the inlier points are present all the frames!

If three points are present in four images and the fourth one is only present in three images, it can be recovered. Point are recovered one by one. Start from the beginning from time to time.
Factorization Results
Perspective factorization

The camera equations for a fixed image $i$ can be written in matrix form as

$$\lambda_{ij} m_{ij} = P_i M_j, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n$$

for a fixed image $i$ can be written in matrix form as

$$m_i \Lambda_i = P_i M$$

where

$$m_i = [m_{i1}, m_{i2}, \ldots, m_{in}] \quad \text{and} \quad M = [M_1, M_2, \ldots, M_n]$$

and

$$\Lambda_i = \text{diag}(\lambda_{i1}, \lambda_{i2}, \ldots, \lambda_{in})$$
Perspective factorization

All equations can be collected for all $i$ as

$$m = PM$$

where

$$m = \begin{bmatrix} m_1 \Lambda_1 \\ m_2 \Lambda_2 \\ \vdots \\ m_m \Lambda_m \end{bmatrix}, \quad P = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_m \end{bmatrix}$$

In these formulas $m$ are known, but $\Lambda_i, P$ and $M$ are unknown.

Observe that $PM$ is a product of a $3m \times 4$ matrix and a $4 \times n$ matrix, i.e. it is a rank 4 matrix.
Iterative perspective factorization

When $\Lambda_i$ are unknown the following algorithm can be used:

1. Set $\lambda_{ij} = 1$ (affine approximation).

2. Factorize $PM$ and obtain an estimate of $P$ and $M$. If $\sigma_5$ is sufficiently small then STOP.

3. Use $m$, $P$ and $M$ to estimate $\Lambda_i$ from the camera equations (linearly) $m_i \Lambda_i = P_i M$


In general the algorithm minimizes the proximity measure $P(\Lambda, P, M) = \sigma_5$

Note that structure and motion recovered up to an arbitrary projective transformation.

(normalize the data)
Normalizing depths.
\[ \alpha_i P_i \] and \[ \beta_j X_j \] does not change \[ \alpha_i \beta_j \lambda_{ij} x_{ij} \].
i-th column and/or j-th row can bring \[ \lambda_{ij} \] close to one.

Normalizing the image coordinates.
To centroid values 0 and average distance \( \sqrt{2} \).

The \[ \lambda_{ij} \sim 1 \] is valid at first iteration if the ratio of true depths in 3D is approximately constant through the sequence!

Bundle adjustment.
The final step! Needs good initialization and has many parameter involved… Levenberg-Marquardt matrix factored… impossible without sparse method.
Further Factorization work

Factorization with uncertainty

(Irani & Anandan, IJCV’02)

Factorization for dynamic scenes

(Costeira and Kanade ‘94)

(Bregler et al. 2000, Brand 2001)


affine, points with different covariances

affine, multibody

non-rigid shapes
Extensions

- Paraperspective
  [Poelman & Kanade, PAMI 97]

- Sequential Factorization
  [Morita & Kanade, PAMI 97]

- Factorization under perspective
  [Christy & Horaud, PAMI 96]
  [Sturm & Triggs, ECCV 96]

- Factorization with Uncertainty
  [Anandan & Irani, IJCV 2002]
Hierarchical structure and motion recovery

- Compute 2-view \((\text{fundamental matrices, } F)\)
- Compute 3-view \((\text{trifocal tensors, } T)\)
- Stitch 3-view reconstructions
- Merge and refine reconstruction
Determining *close* views

- If viewpoints are *close* then most image changes can be modelled through a *planar homography*

- *Qualitative distance measure* is obtained by looking at the *residual error* on the *best possible planar homography*

  \[
  \text{Distance} = \min \text{median} \ D\left( H_m, m' \right)
  \]
Computation of initial structure and motion according to Hartley and Zisserman

“this area is still to some extend a black-art”

All features not visible in all images
\implies No direct method
\implies Build partial reconstructions and assemble
  (more views is more stable, but less corresp.)

1) Sequential structure and motion recovery
2) Hierarchical structure and motion recovery
Structure from Motion: Limitations

- Very difficult to reliably estimate *metric* structure and motion unless:
  - Large $(x$ or $y)$ rotation, or
  - Large field of view and depth variation
- Camera calibration important for Euclidean reconstructions
- Need good feature tracker
Constraints?

• **Scene constraints**
  - Parallellism, vanishing points, horizon, ...
  - Distances, positions, angles, ...
    Unknown scene → no constraints

• **Camera extrinsics constraints**
  - Pose, orientation, ...
    Unknown camera motion → no constraints

• **Camera intrinsics constraints**
  - Focal length, principal point, aspect ratio & skew
    Perspective camera model too general
    → some constraints
Self-calibration

Upgrade from *projective* structure to *metric* structure using *constraints on intrinsic* camera parameters

- Constant intrinsics
  (Faugeras et al. ECCV’92, Hartley’93, Triggs’97, Pollefeys et al. PAMI’98, ...)

- Some known intrinsics, others varying
  (Heyden&Astrom CVPR’97, Pollefeys et al. ICCV’98,...)

- Constraints on intrinsics and restricted motion
  (e.g. pure translation, pure rotation, planar motion)
  (Moons et al.’94, Hartley ’94, Armstrong ECCV’96, ...)

...
Stratification of geometry

- Projective: 15 DOF
  - plane at infinity
  - parallelism

- Affine: 12 DOF
  - absolute conic
  - angles, rel.dist.

- Metric: 7 DOF

More general → More structure
A counting argument

- To go from projective (15DOF) to metric (7DOF) at least 8 constraints are needed
- Minimal sequence length should satisfy

\[
\frac{n \times (\# known)}{\text{HERE cameras}} + \frac{(n - 1) \times (\# fixed)}{\text{internal parameters known in each view}} \geq 8
\]

- Independent of algorithm
- Assumes general motion (i.e. not critical)

if unknown = 5 the equations cannot be solved

other way
\[
\omega_i^{-1} = P_i \cdot Q^* \cdot \text{inf} \cdot P_i^T
\]

dual image of the absolute conic

absolute dual quadric: symmetric, rank deficient 10 - 1 - 1 = 8 dof
Projective SFM Theorem:
Reconstruction is possible as long as
\[2mn \geq 11m + 3n - 15\]

Projective SFM Theorem
Given \(m\) images and \(n\) features
Each point is represented by its homogeneous coordinates \(P_j = [X \ Y \ Z \ 1]^T\)
Each feature is represented by its homogeneous coordinates in the image plane \(p_{ij} = [u_{ij} \ v_{ij} \ 1]^T\)
Each image is represented by its 3x4 projection matrix \(M_i = [A_i \ b_i]\)
(A_i is a 3x3 matrix, \(b_i\) is a 3 vector)
For each feature, we have: \(p_{ij} \sim M_i P_j\) (~ is the same as the triple-bar homogeneous equality: left hand side proportional to right hand side).

Key results:
1. The unknowns \(M_i\) and \(P_j\) can be recovered only up to a 4x4 projective transformation \(Q\). That is, for any 4x4 \(Q\), \((M_i Q Q^{-1} P_j)\) yields the same image projections as \((M_i P_j)\).
2. The unknowns \(M_i\) and \(P_j\) can be recovered if \(2mn \geq 11m + 3n - 15\)
3. In particular, for \(m=2\) cameras, \textit{at least 7 points} are needed.
Metric Upgrade:

The projective reconstruction gives us a set of 3x4 projection matrices $M_i$ for each camera $i=1,\ldots,m$. The next problem is to convert this projective reconstruction to a metric reconstruction. Specifically, we want to find a 4x4 matrix $Q$ such that:

$$M_iQ \equiv K_i[R_i|t_i]$$

$R_i$ and $t_i$ are the rotation/translation between the coordinate system of camera $i$ and an arbitrary coordinate system.

$K_i$ is the matrix of intrinsic parameters of camera $i$, which is defined as:

$$K = \begin{bmatrix} \alpha_u & s & u_o \\ 0 & \alpha_v & v_o \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \alpha & -\alpha\cot\theta & u_o \\ 0 & \beta & v_o \\ 0 & \sin\theta & 1 \end{bmatrix}$$

$\alpha_x$ and $\alpha_y$ are the scales in the $x$ and $y$ directions, $x_o$ and $y_o$ are the coordinates of the center, and $s$ is the skew of the camera ($s = 0$ if the axes are orthogonal.).
Fundamental Transformation:

Our fundamental equation is:

$$M_i Q \equiv K_i [R_i, t_i]$$

Denoting the matrix formed by taking the first 3 columns of $Q$ by $Q_3$, such that $Q = [Q_3, q_4]$, we have: $M_i Q_3 \equiv K_i R_i$.

Taking the first 3 columns and observing that $R_i$ is a rotation matrix:

$$M_i Q_3 R_i^T M_i^T \equiv K_i K_i^T$$

This is the key observation: By writing that the first three columns of the product of $M$ by $Q$ is a rotation, we are able to eliminate the rotation from the unknowns. All that is left are the matrices of internal parameters for each of the images. This process is sometimes called auto-calibration since it amounts to calibrating the internal parameters of the cameras directly from images.

It is important to understand the number of degrees of freedom in $Q_3$. The total number of entries in $Q_3$ is 4x3=12. The matrix is defined up to scale since all the equalities are homogeneous. Moreover, the matrix is defined up to a rotation since for any arbitrary rotation $R$: $Q_3 R R^T Q_3^T = Q_3 Q_3^T$ so that if $Q_3$ is a solution, so is $Q_3 R$. These additional degrees of freedom simply reflect the fact that one can choose the orientation and scale of the global coordinate system arbitrarily. Therefore, $Q_3$ is characterized by $12 - 1 - 3 = 8$ unknowns.

...also like the d.o.f. in absolute dual quadric...
Basic trick:

\[ M_i Q \equiv K_i \begin{bmatrix} R_i & t_i \end{bmatrix} \]

\[ \downarrow \]

\[ M_i Q_3 \equiv K_i R_i \]

Use the fact that \( R \) is a rotation

\[ \downarrow \]

\[ M_i Q_3 Q_3^T M_i^T \equiv K_i R_i R_i^T K_i^T = K_i K_i^T \]

For convenience, we denote the matrix the matrix \( Q_3 Q_3^T \) by \( L \) (a 4x4 matrix) and \( M_i LM_i^T \) by \( \omega_i \). The set of equations to solve is:

\[ M_i LM_i^T \equiv K_i K_i^T \quad i=1,\ldots,m \]

Each image generates 5 independent equations (the left hand side is a 3x3 symmetric matrix, but the equality is up to scale). The total number of unknowns is 8 (\( Q_3 \)) + 5\( m \) (\( K_i K_i^T \)). Therefore, the number of equations (5\( m \)) is *always* lower than the number of unknowns (8 + 5\( m \)) and we can never solve this system of equations without some constraints on the cameras. The key question is what constraints can be used. A couple of constraints are investigated below, followed with a general result.

**Case 1: Identical Intrinsic Parameters:**

Let us suppose now that we do not know the intrinsic parameters of the cameras, but that we do know that they are all identical, that is, \( K_i = K \) for all cameras \( i \). For all the cameras, we have:

\[ M_i LM_i^T \equiv \omega_i \quad i=1,\ldots,m \]  
(with the equality up to scale)

Where \( \omega_i \) is computed from the first image:

\[ M_i LM_i^T \equiv M_i LM_i^T \quad i=2,\ldots,m \]

This gives us 5\((m-1)\) independent equations for 8 unknowns (in \( Q_3 \)). Therefore, we can solve the reconstruction problem in this case if:

\[ 5(m-1) \geq 8 \quad \Rightarrow \quad m \geq 3 \]
\[ M_{11} = K [R_{11} \ T_{11}] \]

\[ M_{m} = K [R_{m} \ t_m] \]

\[ M_{1} = K [R_i \ t_i] \]

\[ K = \begin{bmatrix}
\alpha_u & s & u_o \\
0 & \alpha_v & v_o \\
0 & 0 & 1
\end{bmatrix} \]

Note also that we solved for 8 unknowns for \( Q_3 \), even though it is a 4x3 matrix \( \rightarrow 12 \) elements. The difference is due to the fact that the reconstruction is defined up to a global rotation, in other words, \( Q_3 R \) for any rotation \( R \) is also a solution.

We can verify the parameter count:

\[ 8 + 3 + 4 + 1 = 16 \]

This is equal to a 4x4 matrix.

\[ Q_3 Q_3^T \]

Arbitrary rotation

\[ q_4 \]

Arbitrary scale

It is important to note that by manipulating the equations so that the unknown becomes \( L \), we have effectively eliminated the rotation and translation and reduced the problem to the recovery of the intrinsic parameters. This step is often termed \textit{self-calibration}. 

13
Case 2: Principal point at origin

In that case, \( u_o = v_o = 0 \), which implies that \( \omega_{13} = \omega_{23} = 0 \). Therefore, going back to the original equation, we can write that:

\[
\begin{bmatrix}
\alpha_u^2 + s^2 + u_o^2 & s\alpha_v + u_o v_o & u_o \\
\alpha_v^2 + v_o^2 & v_o \\
\end{bmatrix}
\]

Those are two equations in \( L \) that are independent of \( K_i \). We have 2\( m \) such equations for \( m \) views for 8 unknowns in \( L \) (meaning, 8 unknowns in \( Q_3 \)). These equations are independent of \( K \). Therefore:

If the principal point is at the origin, a metric reconstruction can be obtained from a minimum of 4 views.
### Assumption Conditions

<table>
<thead>
<tr>
<th>Assumption</th>
<th>Fixed</th>
<th>Known</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant $K$</td>
<td>5</td>
<td>0</td>
<td>$\omega_{ij}/\omega_{33} = \omega_{ij}/\omega_{33}$</td>
</tr>
<tr>
<td>Principal point known at the origin</td>
<td>0</td>
<td>2</td>
<td>$\omega_{13}=\omega_{23}=0$</td>
</tr>
<tr>
<td>Aspect ratio and skew constant</td>
<td>2</td>
<td>0</td>
<td>more complicated</td>
</tr>
<tr>
<td>Zero Skew</td>
<td>0</td>
<td>1</td>
<td>$\omega_{12}=\omega_{33}=\omega_{13}=\omega_{23}$</td>
</tr>
<tr>
<td>P.P. known and Zero skew</td>
<td>0</td>
<td>3</td>
<td>$\omega_{12}=0$ $\omega_{13}=\omega_{23}=0$</td>
</tr>
</tbody>
</table>

**Case 3: Zero-Skew**

If the skew is zero but all the other parameters are allowed to vary, then we have the constraint:

$$\omega_{12}\omega_{33} = \omega_{13}\omega_{23}$$

from 2HZTable 19.1 it gives $s=0$.

This provides $m$ constraints. Therefore, we must have $m>=8$, thus 8 images are necessary.

### General counting argument:

The total number of parameters to be estimated from the set of equations above is 8: $L = Q_3Q_5^T$ is a 4x4 symmetric matrix that is defined by 10 entries, 9 of which are independent because of the scale factor. There is one more constraint that $\det(L) = 0$ (it is of rank 3). Thus the number of independent parameters is 10-1-1=8.

- If we know $k$ internal parameters, then we have $km$ constraints
- If we know that $f$ internal parameters are fixed (but unknown) we have $f(m-1)$ constraints
- Therefore, we can recover a metric reconstruction iff:

$$mk + (m-1)f >= 8$$
Recovering Q:

One problem in the previous result is that, while we used the matrix L for convenience, the actual unknown is $Q_3$ with $L = Q_3 Q_3^T$, which is a non-linear relation. This can be inconvenient because we have to solve a set of non-linear equations. In fact, it is possible in many cases to solve the problem linearly. This is done by using the same trick as before with $F$ and $E$, and with the factorization method: Let’s pretend first that we solve the equations in $L$, which has 9 degrees of freedom ($4 \times 4$ symmetric matrix = 10, but it is up to scale), and then decompose $L$ into $L = Q_3 Q_3^T$.

For example, in case 2 (Principal point at origin), $u_o = v_o = 0$, which implies that $\omega_{13} = \omega_{23} = 0$. Therefore:

$$(M_i L M_i^T)_{13} = 0 \text{ and } (M_i L M_i^T)_{23} = 0$$

Those are two equations that are linear in $L$ and can be solved very easily!!

Once we have $L$, we need to decompose it back into $L = Q_3 Q_3^T$. There is a practical difficulty here: Since $Q_3$ is a $4 \times 3$ matrix, for such a decomposition to exist, $L$ must be of rank at most 3 which is not enforced in the linear solution.

We can find the closest matrix $L_3$ of rank three as follows:

$L$ is a symmetric matrix so $L = U D U^T$, where $D$ is a $4 \times 4$ diagonal matrix of eigenvalues and $U$ is a $4 \times 4$ rotation matrix. The matrix $L_3$ of rank 3 that is closest to $L$ can be formed by eliminating the smallest eigenvalue of $L$, that is, $L_3 = U_3 D_3 U_3^T$, where $U_3$ is the $4 \times 3$ matrix obtained by removing the last column of $Q$ and $D_3$ is the $3 \times 3$ upper right block of $D$. With this decomposition, the matrix $Q_3 = U_3 D_3^{\frac{1}{2}}$ is a solution since $L_3 = Q_3 Q_3^T$.

$K$ can be recovered by Cholesky decomposition $\omega = K K^T$. 

dual image of the absolute conic
• Given $M_i$
• Solve for $L$ such that:
  • $M_iLM_i^T \sim M_iLM_i^T$ ($i = 2,..,m$)
• Diagonalize $L$: $L = UDUT$
• Approximate by rank-3 matrix: $L_3 = U_3D_3U_3^T$
• Compute $Q_3$: $Q_3 = U_3D_3^{1/2}$
• Compute $q_4$ by setting the origin of the first camera to 0: $M_1q_4 = 0$
• Return $Q = [Q_3 q_4]$

Given $Q_3$, the last column of $Q$ is computed by setting the origin at the origin of the first camera, that is: $M_1q_4 = 0$. Note that any scaled version of $q_4$ is a solution. This is a consequence of the fact that it is not possible to recover the absolute scale of the translation between the cameras. [similarity invariant!]

Remember that we solved for 8 unknowns for $Q_3$, even though it is a 4x3 matrix $\rightarrow$ 12 elements. The difference is due to the fact that the reconstruction is defined up to a global rotation, in other words, $Q_3R$ for any rotation $R$ is also a solution.

We can verify the parameter count:

$$8 + 3 + 4 + 1 = 16$$

It is important to note that by manipulating the equations so that the unknown becomes $L$, we have effectively eliminated the rotation and translation and reduced the problem to the recovery of the intrinsic parameters. This step is often termed auto-calibration.
Non-Linear Approach: Bundle Adjustment (projective case)

The discussion so far has assumed a linear reconstruction that (implicitly) minimizes the error $|λ_{ij}p_{ij} - M_i p_j|^2$. This error is not the true geometric error, i.e., the error between a feature position and the projection of a reconstructed point.

Suppose that we have estimated the projection matrices $M_i (i=1..m)$ using the previous (linear) techniques. Suppose also that we a set of corresponding points in the images $p_{ij} (j=1..n)$ corresponding to $n$ points in the scene. For each $j=1..n$, we assume that we have an initial reconstruction of the corresponding scene point $p_j$.

Ideally, the data point $p_{ij} = [u_{ij}, v_{ij}, 1]^T$ should be identical to the projection of the reconstructed point $M_i p_j$. Starting with an initial estimate of the projection matrices and the scene points, we want to minimize the geometric distance between $p_{ij}$ and $M_i p_j$:

$$E = \sum_{i,j} \left( u_{ij} - \frac{m_{i1}^T P_j}{m_{i2}^T P_j} \right)^2 + \left( v_{ij} - \frac{m_{i2}^T P_j}{m_{i3}^T P_j} \right)^2$$

Where $m_{i1}^T, m_{i2}^T, m_{i3}^T$ are the rows of $M_i$. Summing over all the points and all the images, we have to find the minimum of:

Over all the $M_i$ and $P_j$ (a total of $11m + 3n$ variables). This technique is called bundle adjustment, widely used in photogrammetry.
Minimization algorithm: Let $X$ be the vector formed by concatenating all the $N = 11m + 3n$ unknowns of the problem, the matrices $M_i$ and the points $P_j$. The error function can be written as:

$$E(X) = \|f(X)\|^2$$

with

$$f(X) = \left[ \varepsilon_{11} \quad \varepsilon_{12} \quad \ldots \quad \varepsilon_{ij} \quad \ldots \quad \varepsilon_{mn} \right]^T$$

with $\varepsilon_{ij} = \left[ u_g - \frac{m_{ij}^T P_j}{m_{ij}^T P_j} \quad v_g - \frac{m_{ij}^T P_j}{m_{ij}^T P_j} \right]^T$

A basic implementation of bundle adjustment uses an iterative Gauss-Newton algorithm for solving the non-linear least-squares: If $X_k$ is the value at iteration $k$, a first order approximation gives us:

$$f(X_k + \Delta X) = f(X_k) + J_k \Delta X$$

Where $J_k$ is the Jacobian ($2mn$ rows by $N$ columns matrix of derivatives) of $f$. We want to find $\Delta X$ that minimizes the right-hand side of this first order equation. The solution is found using the standard least-squares solution:

$$\Delta X = (J_k^T J_k)^{-1} J_k^T f(X_k)$$

$\Delta X$ is added to the current estimate $X_k$ to yield the next estimate $X_{k+1}$.

Convergence issues: The algorithm assumes that the starting point $X_o$ is close to the minimum. The algorithm may converge to a local minimum or even diverge otherwise. Typically, $X_o$ is obtained from one of the previous linear techniques.

---

see also Levenberg-Marquardt in HZ or Section 4 in Lucas-Kanade 20 paper (lecture 5)
Final Adjustment (metric case):

Once we have the intrinsic parameters \( K_i \) through self-calibration, we recover the perspective projection matrices by transforming the ones obtained from the projective reconstruction as: \( M_i Q \).

Similarly, the reconstructed points are transformed by the inverse transformation: \( Q^{-1} P_j \).

This gives an initial estimate of the \( P_j \)'s. In general, reconstruction systems include a last step in which all the matrices and points are adjusted simultaneously using bundle adjustment. This is the same bundle adjustment objective function as before, except that the minimization is done explicitly with respect to the components \( K_i, R_i, \) and \( t_i \) instead of with respect to the \( M_i \)'s since we need to enforce a metric reconstruction.

\[
\text{Min}_{P_j, K_i, R_i, t_i} \sum_{i,j} \left( u_{ij} - \frac{m_{1i}^{T} P_j}{m_{ji}^{T} P_j} \right)^2 + \left( v_{ij} - \frac{m_{2i}^{T} P_j}{m_{3i}^{T} P_j} \right)^2
\]

It is also important to know the uncertainty on the resulting reconstruction: Points far away are more uncertain, for example. If we assume that all the errors are Gaussian, the matrix \( J_i^{T}J \) is the covariance matrix representing the error distribution of all the unknown parameters. The principal direction of \( J_i^{T}J \) are the main directions of uncertainty, the eigenvalues are the uncertainties in those directions. The matrix can be interpreted as characterizing the curvature of the error surface in different directions.
**Computational issues:** Bundle adjustment may involve hundreds of variables. As a result, the computation of $J_k^TJ_k$ and its inversion may be expensive and numerically unstable. An important fact to note is that the derivatives of $\epsilon_{ij}$ with respect to $M_i$ for $l \neq i$ is zero. Similarly, the derivative of $\epsilon_{ij}$ with respect to $P_l$ is zero for any $l$ other than $l=j$. Therefore, $J_k^TJ_k$ is very sparse and this property can be exploited to speed up the iterations.
- **Gauge issues:** The reconstruction is defined up to a similarity transformation. That is, one can transform the entire set of $P_j$ by an arbitrary transformation and transform the projection matrices accordingly to get a completely equivalent reconstruction. This causes a numerical problem because the minimum of the surface $E = E(X)$ is not a point but it is a valley of equivalent minima corresponding to different transformations. It is therefore important to either fix the transformation, or to modify the function so that the minimization is invariant to transformations. Technically, this the gauge issue (choosing a particular transformation is choosing a gauge.) A simple way to do this, for example, is to fix the projection matrix of the first camera and to express everything with respect to the first camera. Another possibility is to fix a set of scene points (a basis). It is important to note that, for numerical reasons, the result may be different depending on the selecting gauge.

- **Outliers:** This has assumed so far that all the correspondences are correct. In fact, if some correspondences are incorrect (outliers), the entire function $E$ is corrupted. To avoid this, a different function is used in place of the squared distance between features and scene projections. This is the object of *robust estimation*, to be discussed later.
Uncertainty:
The uncertainty on each point $P_j$ and the cameras orientation $R$ and position $t$ can be recovered by projecting on the appropriate subspaces of parameters (technically, taking the marginal of the distributions).

The distribution of uncertainty depends strongly on the gauge constraint: Obviously, the uncertainty is 0 around the camera that is chosen as reference. This is the same situation in mobile robotics in which one gets a different distribution of uncertainty if the map is expressed with respect to the starting position of the robot, or with respect to the current position.
**COMPLETE SYSTEM** images + features

**PROJECTIVE**

- **Epipolar geometry:** Fundamental matrix estimation (min. 2 images + 7 correspondences)
  \[ \text{Min} \sum_i (p_i^T F p_i')^2 + \text{rank-2 SVD reduction} \]
- **Linear eight-point + RANSAC**
- **Non-linear refinement**
  \[ \text{Min} \sum_i d^2(p_i, Fp_i') + d^2(p_i', F'p_i') \]

**Self-calibration (intrinsic parameter matrix K):**
\[ M_i Q_i Q_i^T \equiv K_i K_i' \]

**MÉTRIC**

- **Projective reconstruction:**
  \[ F \rightarrow A = [b] F \ F^T b = 0 \rightarrow M = [A \ b] \]

- **Metric reconstruction:**
  \[ M_i Q = K_i [R_i, t_i] \ P_j \leftarrow Q^{-1} P_j \]

**REFINEMENT**

- **Bundle adjustment:**
  \[ \text{Min} \sum_{i,j} \left( u_{ij} - \frac{m_i^T p_j}{m_i^T p_j} \right)^2 + \left( v_{ij} - \frac{m_i^T p_j}{m_i^T p_j} \right)^2 \]

---

dual image for absolute conic, omega*, may not be positive-definite even if you correct maybe not consistent with metric reconstruction
Structure from motion problem

From the $m \times n$ correspondences $x_{ij}$, estimate:

- $m$ projection matrices $M_i$
- $n$ 3D points $X_j$
The Structure-from-Motion Problem

Given $m$ images of $n$ fixed points $X_j$ we can write

$$x_{ij} = M_i X_j \quad \text{for } i = 1, \ldots, m \quad \text{and} \quad j = 1, \ldots, n.$$  

**Problem:** estimate the $m$ $3 \times 4$ matrices $M_i$ and the $n$ positions $X_j$ from the $m \times n$ correspondences $x_{ij}$.

- With no calibration info, cameras and points can only be recovered up to a $4 \times 4$ projective
- Given two cameras, how many points are needed?
- How many equations and how many unknown?

$2m \times n$ equations in $11m + 3n - 15$ unknowns

So 7 points! [$2 \times 2 \times 7 = 28; \ 11 \times 2 + 3 \times 7 - 15 = 28$]
Bundle adjustment

Non-linear method for refining structure and motion

Minimizing re-projection error

\[ E(M, X) = \sum_{i=1}^{m} \sum_{j=1}^{n} D(x_{ij}, M_i X_j)^2 \]
Bundle adjustment

Non-linear method for refining structure and motion
Minimizing re-projection error

$$E(M, X) = \sum_{i=1}^{m} \sum_{j=1}^{n} D(x_{ij}, M_i X_j)^2$$

**Advantages**
- Handle large number of views
- Handle missing data

**Limitations**
- Large minimization problem (parameters grow with number of views)
- Requires good initial condition

Used as the final step of SFM
Removing the ambiguities: the Stratified reconstruction

• up grade reconstruction from perspective to affine
  [by measuring the plane at infinity]

• up grade reconstruction from affine to metric
  [by measuring the absolute conic]

Recovering the metric reconstruction from the perspective one is called self-calibration
Self-calibration

Process of determining intrinsic camera parameters directly from un-calibrated images

Suppose we have a projective reconstruction \( \{M_i, X_j\} \)

**GOAL:** find a rectifying homography \( H \) such that

\[
\{M_i H, H^{-1} X_j\}
\]

is a metric reconstruction

\[
\overline{M}_i = M_i H \quad i = 1 \cdots m \\
\overline{M}_i = K_i[R_i \ T_i]
\]

If world ref. system = camera 1 ref. system:

\[
\overline{M}_1 = K_1[I \ 0]
\]

If the perspective camera is canonical:

\[
\overline{M}_1 = [I \ 0]
\]
Self-calibration basic equation

\[ \overline{M}_i = M_i \ H \quad i = 2 \cdots m \]

\[ M_i = [A_i \ a_i] \quad = \text{perspective reconstruction of the camera} \]

\[ \overline{M}_i = K_i[R_i \ T_i] \]

\[ H = \begin{bmatrix} K_1 & 0 \\ -p^T K_1 & 1 \end{bmatrix} \]

\[ = \begin{bmatrix} 1 & 0 \\ -p^T & 1 \end{bmatrix} \begin{bmatrix} K_{11} & 0 \\ 0 & 1 \end{bmatrix} \quad \text{plane at infinity and the internal calibration from the DIAC first image} \]

\[ 3 + 5 = 8 \text{ unknown} \]

\[ \begin{bmatrix} K_i & R_i \\ T' \end{bmatrix} = [A_i \ a_i] \begin{bmatrix} K_1 & 0 \\ -p^T K_1 & 1 \end{bmatrix} = \begin{bmatrix} A_i K_1 - a_i p^T K_1 & a_i \end{bmatrix} \]

\[ K_i R_i = (A_i - a_i p^T) K_1 \quad \rightarrow \quad R_i = K_i^{-1} (A_i - a_i p^T) K_1 \]
Self-calibration basic equation

\[
\begin{align*}
R_i &= K_i^{-1}(A_i - a_ip^T)K_1 \\
R_i^T &= K_1^T (A_i - a_ip^T)^T K_i^{-T} \\
R_i R_i^T &= I \\
K_i^{-1}(A_i - a_ip^T)K_1 K_1^T (A_i - a_ip^T)^T K_i^{-T} &= I
\end{align*}
\]

\[\text{H\_inf the i frame relative the reference} \quad \text{det(H\_inf) = 1} \quad \text{dual image of the absolute conic view i}\]
Absolute conic $\Omega_\infty$ is a $C \in \Pi_\infty$

Any $x \in \Omega_\infty$ satisfies:

$$x^T \Omega_\infty x = 0$$

$$\Omega_\infty = \begin{bmatrix}
1 & & \\
& 1 & \\
& & 0
\end{bmatrix}$$

$$\begin{cases}
x_1^2 + x_2^2 + x_3^2 = 0 \\
x_4 = 0
\end{cases}$$

Projective transformation of $\Omega_\infty$

$$\omega = (K^T K)^{-1}$$

$$\omega^* = KK^T$$

Dual image of the absolute conic
Self-calibration basic equation

\[
(A_i - a_ip^T)K_1 K_1^T(A_i - a_ip^T)^T = K_i K_i^T
\]

\[
(A_i - a_ip^T)\omega_i^*(A_i - a_ip^T)^T = \omega_i^* \quad \text{i=2...m}
\]

[A_i and a_i are known]

How many unknowns? • 3 from p
• 5 from \(\omega_i\) [per view]

How many equations? 5 independent equations [per view]

Art of self-calibration:
use constraints on \(\omega\) (K) to generate enough equations on the unknowns
Self-calibration – identical Ks

\[(A_i - a_i p^T) \omega^* (A_i - a_i p^T)^T = \omega^* \]

\[\downarrow\]

\[(A_i - a_i p^T) \omega^* (A_i - a_i p^T)^T = \omega^* \]

• For m views, 5(m-1) constraints

• Number of unknowns: 8

\[m \geq 3\] provides enough constraints

To solve the self-calibration problem with identical cameras we need at least 3 views