Robot Vision

TTK4255

Lecture 09 – Structure from Motion

Annette Stahl

(Annette.Stahl@ntnu.no)

Department of Engineering Cybernetics – ITK

NTNU, Trondheim

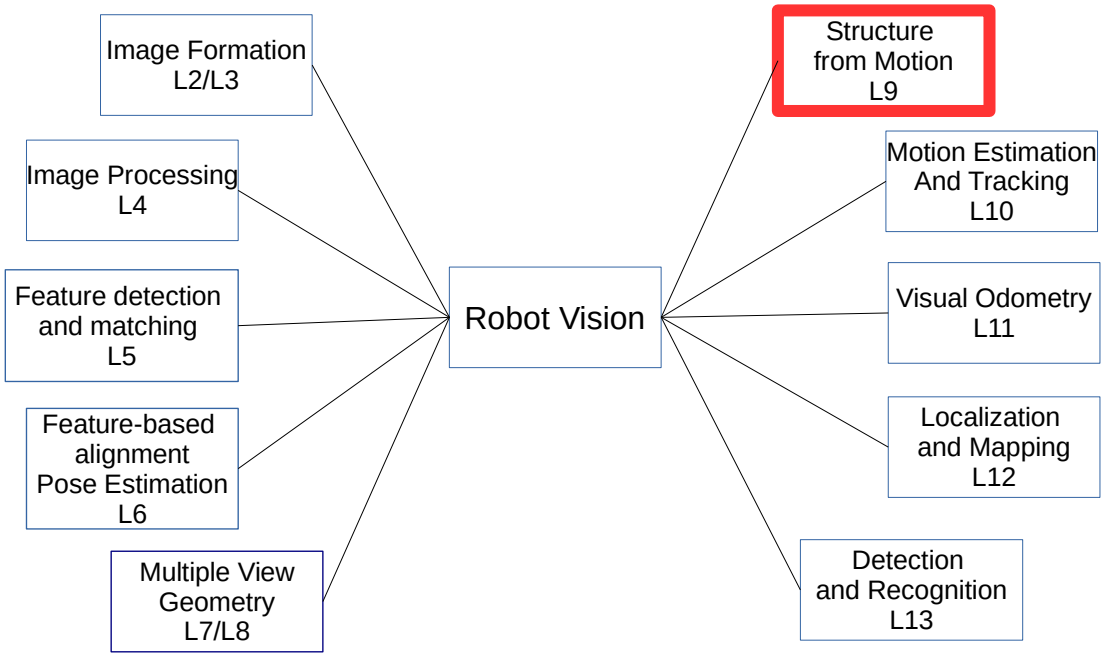
Spring Semester

02. March 2020

Lecture 09 – Structure from Motion

Annette Stahl (Annette.Stahl@ntnu.no)

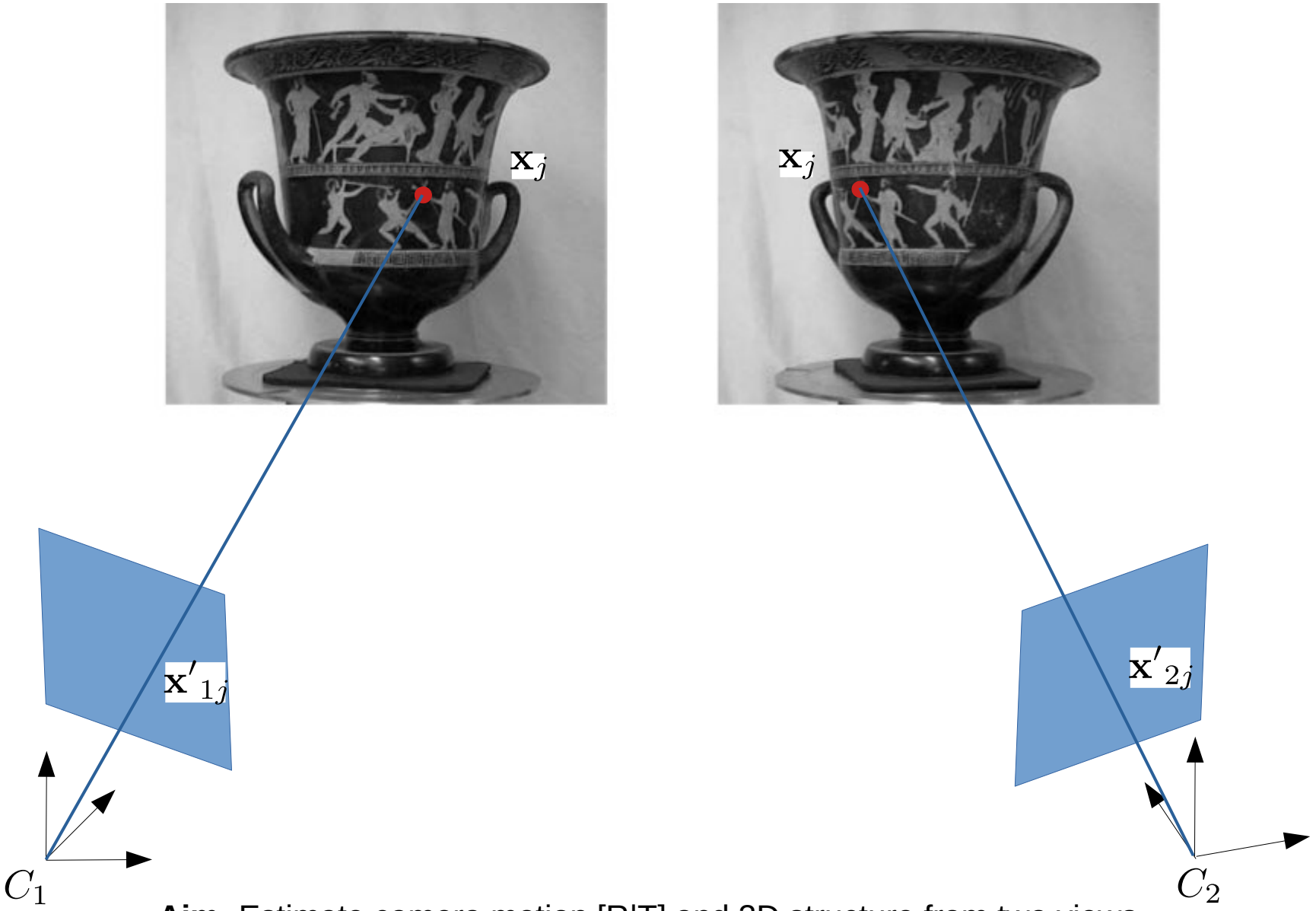
Simen Haugo (Simen.Haugo@ntnu.no)



Outline of the ninth lecture:

- Three-view geometry
- Multiple-view geometry
- Stereo SfM
- Alternative Methods

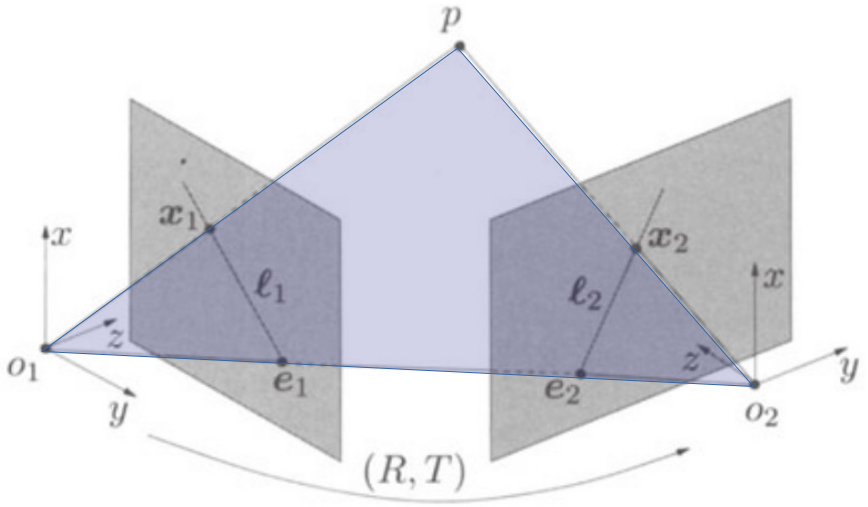
Structure from Motion – Problem Definition



Aim: Estimate camera motion [R|T] and 3D structure from two views

Recap L07: Epipolar Constraint

$$X_r = RX_l + T$$
$$\lambda_r \mathbf{x}_r = R\lambda_l \mathbf{x}_l + T$$
$$\lambda_r [T]_\times \mathbf{x}_r = [T]_\times R\lambda_l \mathbf{x}_l + [T]_\times T$$
$$\mathbf{x}_r^\top \lambda_r [T]_\times \mathbf{x}_r = \mathbf{x}_r^\top [T]_\times R\lambda_l \mathbf{x}_l$$
$$0 = \mathbf{x}_r^\top [T]_\times R\mathbf{x}_l$$
$$0 = \mathbf{x}_r^\top E\mathbf{x}_l$$



- The **Epipolar constraint** holds for every pair of corresponding points
- Two images of the same point $\mathbf{p} = \mathbf{x}$ from two camera positions with relative pose $(R,\ T)$, ($R \in SO(3)$ relative orientation, $T \in \mathbb{R}^3$ relative position) satisfy the epipolar constraint equation.
- Where $E = [T]_\times R \in \mathbb{R}^{3\times 3}$ is the **essential matrix**
- The epipolar constraint gives the relative pose between two cameras
- The Essential Matrix can be decomposed into R and T recalling that four distinct solutions for R and T are possible. [LonguetHiggins1981]

Recap L07

Two-view Structure from Motion

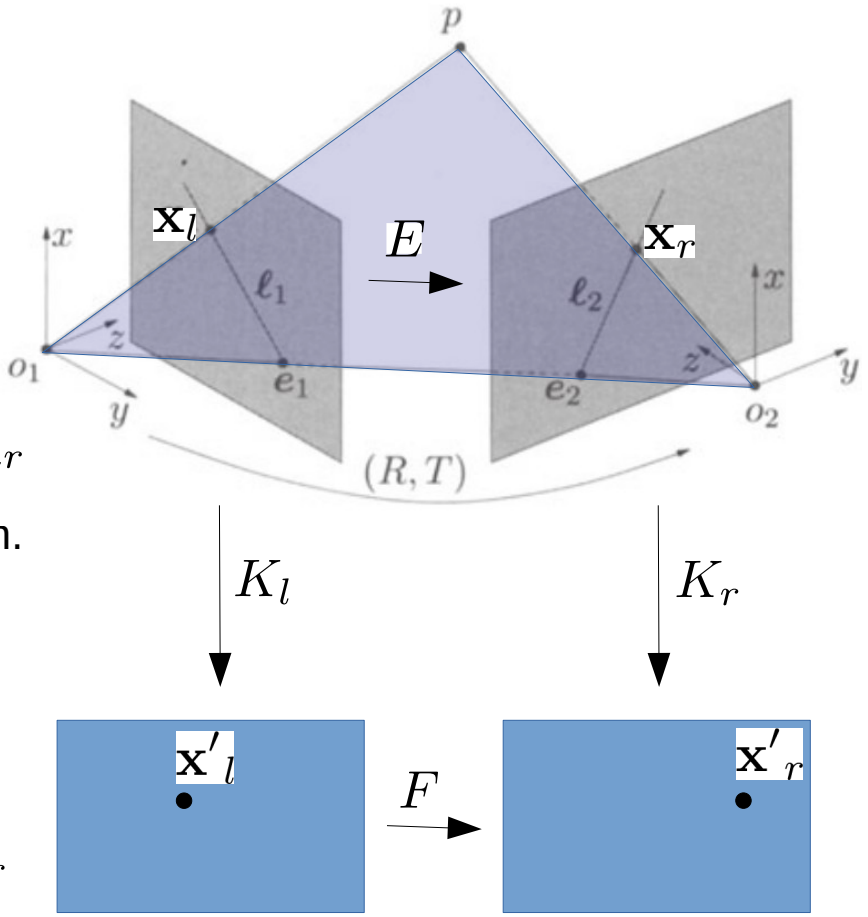
- **Calibrated epipolar geometry**: $K_l,\ K_r$ are known
  - Calibrated epipolar constraint $\mathbf{x}_r^\top E \mathbf{x}_l = 0$
  - **Essential matrix**

    $E = [T]_\times R \in \mathbb{R}^{3\times 3}$

    → estimate E from 5 (7, 8) correspondences $\mathbf{x}_{il} \leftrightarrow \mathbf{x}_{ir}$
- **Uncalibrated epipolar geometry**: $K_l,\ K_r$ are unknown.
  - Uncalibrated epipolar constraint $\mathbf{x'}_r^\top F \mathbf{x'}_l = 0$
  - **Fundamental matrix**

    $F = K_r^{-T}[T]_\times R K_l^{-1} = K_r^{-T} E K_l^{-1}$

    → estimate F from 7 (8) correspondences $\mathbf{x'}_{il} \leftrightarrow \mathbf{x'}_{ir}$
- **Pose (Motion)** from epipolar geomtry
  - Decompose E into R and T (up to scale)
- **3D Structure** from epipolar geometry
  - Triangulation based on known camera matrices



Structure from Motion

Definition: Automatic recovery of

- camera motion (pose estimation between cameras) and
- scene structure (3D reconstruction)

From two and more images.

Structure from Motion (SfM) is a self calibration technique and is also called automatic camera tracking

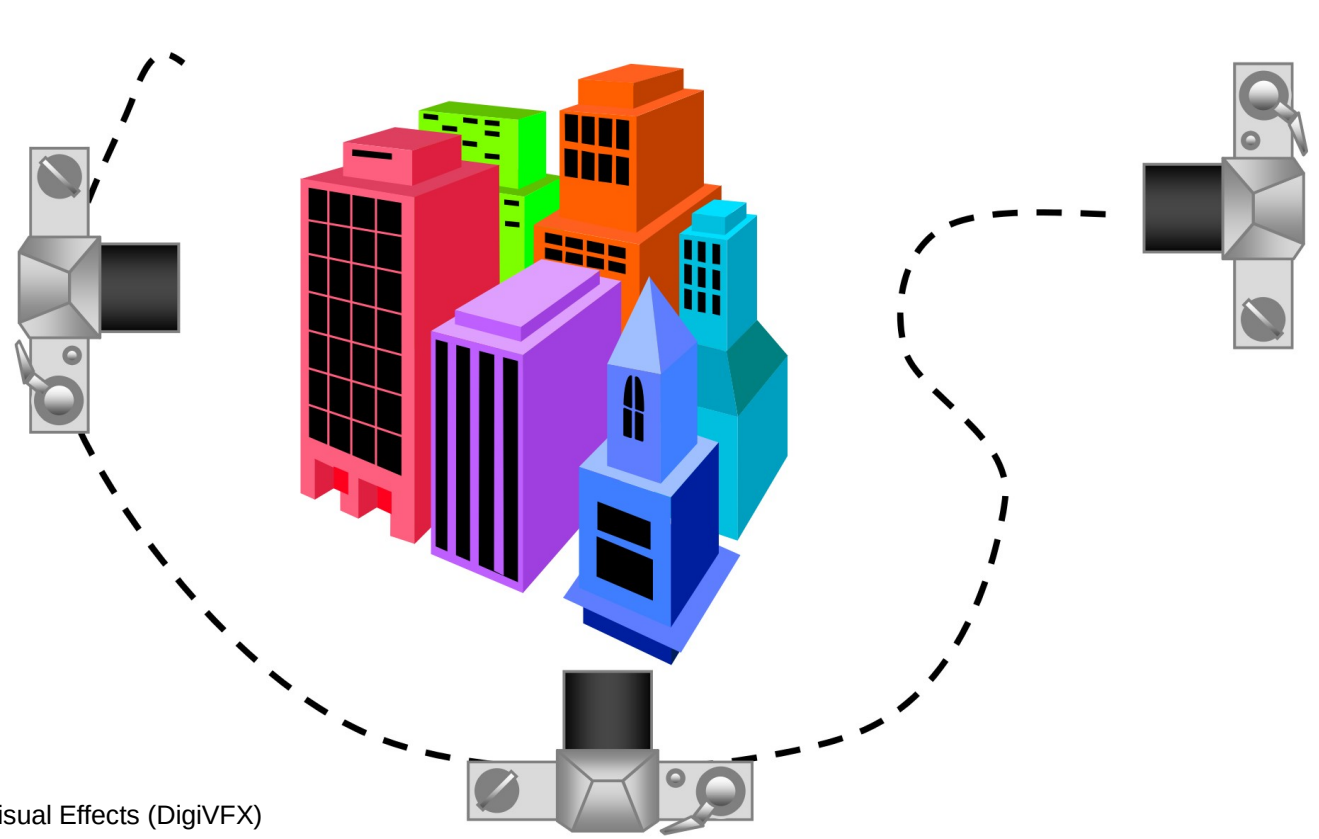
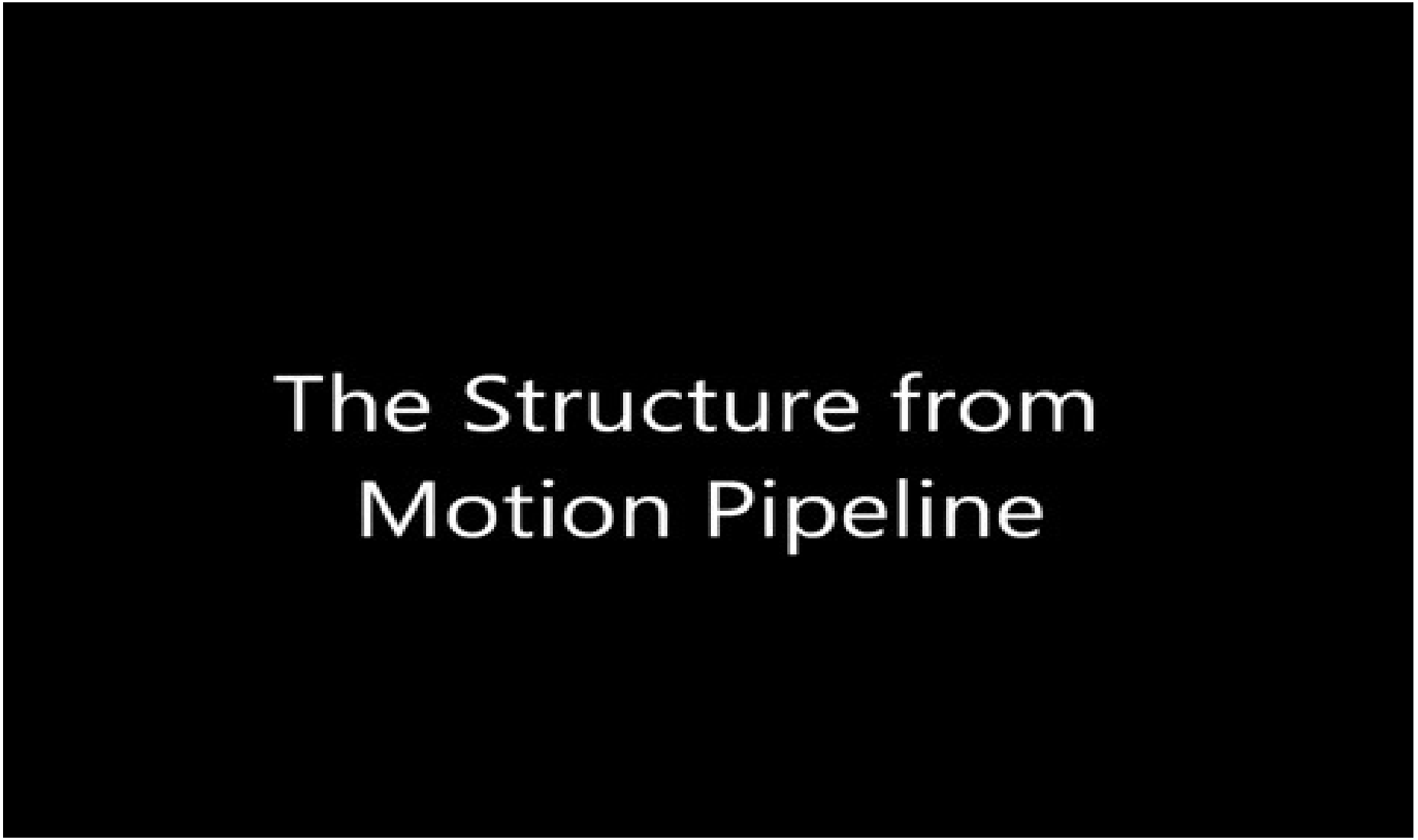


Image courtesy: Digital Visual Effects (DigiVFX)

Structure from Motion



Enqvist, O., Kahl, F., & Olsson, C. (2011). Non-Sequential Structure from Motion. 11thWorkshop on Omnidirectional Vision, Camera Networks and Non-Classical Cameras (OMNIVIS 2011)

Multiple-View Structure from Motion

In general:

Correspondences (matching) → **multiple-view geometry?**

- Must satisfy the epipolar constraint

Reconstruction (structure) → **multiple-views for 3D reconstruction?**

- Sparse 3D reconstruction from triangulation
- Dense 3D reconstruction from stereo processing

Epipolar geometry (motion) → **determine camera poses with multiple views?**

- Calibrated case, the relative pose between cameras can be estimated up to scale by decomposing the essential matrix $E = USV^\top$
- Uncalibrated case, projection matrices P may be estimated from the fundamental matrix F up to projective ambiguity
  $\mathbf{x}_r^\top P_r^\top \hat{F} P_l \mathbf{x}_l = 0 \quad \Rightarrow F = P_r^\top \hat{F} P_l$

Multiple-View Structure from Motion

Correspondences (matching)

- **Two views:**
  - points $\mathbf{x}'_{i1}$ and $\mathbf{x}'_{i2}$ must satisfy epipolar constraint
  - Fundamental matrix $F_{2,1}$ represents this constraint $\mathbf{x}'^{\top}_{2} F_{2,1} \mathbf{x}'^{\top}_{1} = 0$
  - F describes the correspondence between points and epipolar lines $\mathbf{l}'_2 = F_{2,1}\mathbf{x}'_1$

$$\mathbf{l}'_1 = (F_{2,1})^{\top}\mathbf{x}'_2$$

Multiple-View Structure from Motion

Correspondences (matching)

- **Two views:**
  - points $\mathbf{x'}_{i1}$ and $\mathbf{x'}_{i2}$ must satisfy epipolar constraint
  - Fundamental matrix $F_{2,1}$ represents this constraint $\mathbf{x'}_2^\top F_{2,1} \mathbf{x'}_1^\top = 0$
  - F describes the correspondence between points and epipolar lines $\mathbf{l'}_2 = F_{2,1}\mathbf{x'}_1$

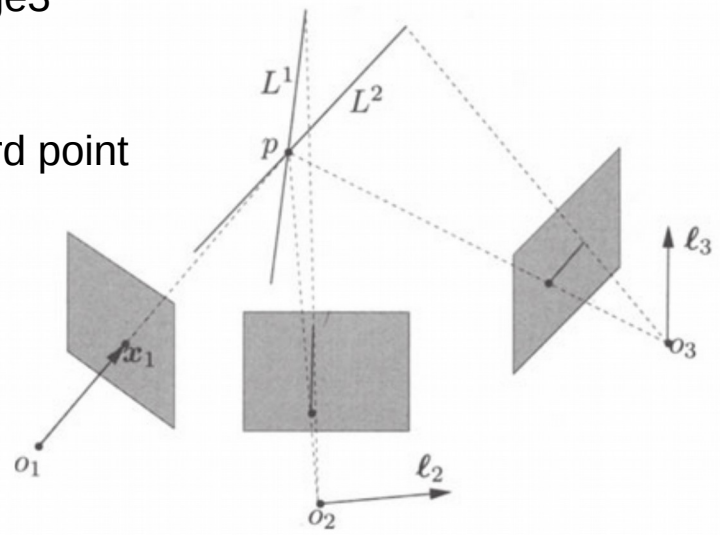
    $\mathbf{l'}_1 = (F_{2,1})^\top \mathbf{x'}_2$
- **Three views:**
  - A point $\mathbf{x'}_2$ in image2 corresponds to lines in image1 and image3
  - A point $\mathbf{x'}_1$ in image1 corresponds to lines in image2 and image3
  - Points $\mathbf{x'}_1$ and $\mathbf{x'}_2$ define a point $\mathbf{x'}_3$ in image3
  - Two of the points out of the set $\{\mathbf{x'}_1, \mathbf{x'}_2, \mathbf{x'}_3\}$ define the third point

    → Three points are connected by a geometric constraint:

    each point can be computed from two other points

    → lines in two views generate a line in a third view

    → three-view geometry: trifocal tensor



Three-view Geometry

Trifocal Tensor

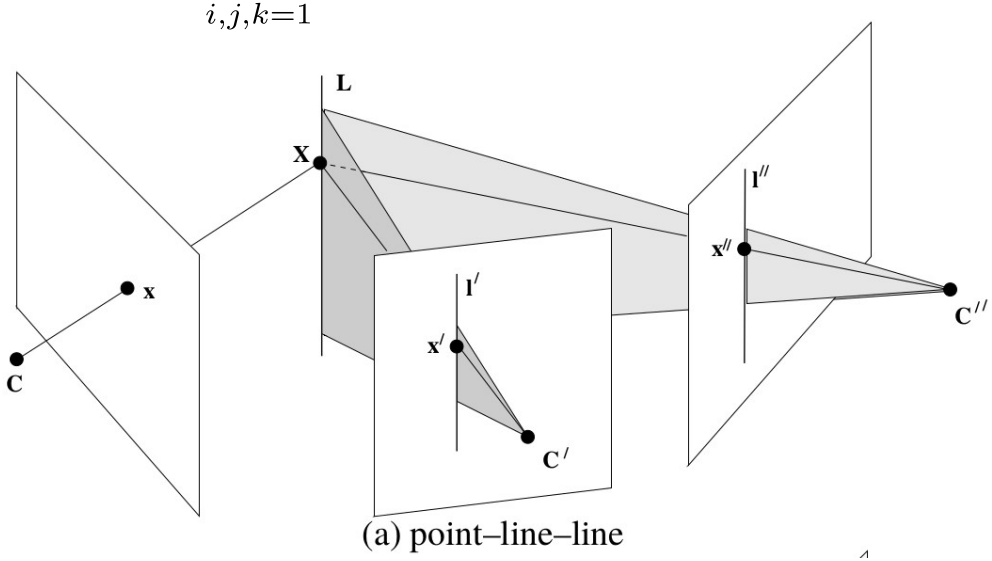
- As of the epipolar constraint for two-view geometry in the case of three-view geometry the fundamental matrix F is replaced by the notion of the **trifocal tensor** $\mathcal{T}$
- Like the fundamental matrix, the trifocal tensor depends (nonlinearly) on the motion parameters R and T.
- It consists of 3x3x3 entries
- the fundamental matrix has 7dof, whereas the trifocal tensor has 18dof

Three-view Geometry

Trifocal Tensor

- From a point-correspondence $\mathbf{x} \leftrightarrow \mathbf{x}' \leftrightarrow \mathbf{x}''$ choose any lines l' and l'' passing through the points x' and x'' generate a relation

$$\mathcal{T}(\mathbf{x}, \mathbf{l}', \mathbf{l}'') = \sum_{i,j,k=1}^{3,3,3} \mathcal{T}(i,j,k)\mathbf{x}(i), \mathbf{l}'(j), \mathbf{l}''(k) = 0$$



(a) point–line–line

- It is possible to choose two independent lines passing through x', and two others passing through x'', one can obtain four independent equations in this way.

  → A total of seven point correspondences are sufficient to compute the trifocal tensor linearly

  → It can be computed from a minimum of six point correspondences using a non-linear method.

Three-view Geometry

Trifocal Tensor

- As with the fundamental matrix, once the trifocal tensor is known, it is possible to **extract the three camera matrices** from it, and thereby **obtain a reconstruction of the scene points and lines** → reconstruction is unique **only up to a 3D projective transformation**
- **Advantages** to using such a three-view method for reconstruction:
  - It is possible to use a **mixture of line and point correspondences** to compute the projective reconstruction. With two views, only point correspondences can be used.
  - Using three views **gives greater stability to the reconstruction**, and avoids unstable configurations that may occur using only two views for the reconstruction.

Three-view Geometry

Trifocal Tensor

- This tensor governs the relationship between points and lines in three views (HZ,Ch. 15.1):
  - Point-point-point
  - Point-point-line
  - Point-line-line
  - Point-line-point
  - Line-line-line
- It may be used to transfer a two-view point/line correspondence into a point/line in a third view
- point transfer can be done directly from the epipolar constraints

$$\mathbf{x'}_3 = (F_{3,1}\mathbf{x'}_1) \times (F_{3,2}\mathbf{x'}_2)$$

  → fails for points in the plane defined by the three camera centers
  = the trifocal plane (→ epipolar lines will coincide)

- The trifocal tensor allows point transfer also for points in the trifocal plane
- Uncertainty in feature points transfer to uncertainty in the epipolar lines
- Reliability of the predicted point depends on the angle between the epipolar lines → large angle is preferable

Three-view Geometry

Correspondences (matching)

- More views enables us to **reveal and remove more mismatches** than we can do in the two-view case
- More views enables us to **predict correspondences** that can be tested with or without the use of descriptors
- **Uncertainties** in these predictions will in general **decrease** with the number of views

N-view Structure from Motion

Given are m images of n fixed 3D points, estimate the m projection matrices P and the n points $\mathbf{x}_j$ from the mn correspondences $\mathbf{x'}_{ij} \leftrightarrow \mathbf{x'}_{kj}, j = 1, 2, ..., n$

We can solve for structure and motion when $2mn \geq 11m + 3n - 15$

In the general/uncalibrated case, cameras and points can only be recovered up to a projective ambiguity

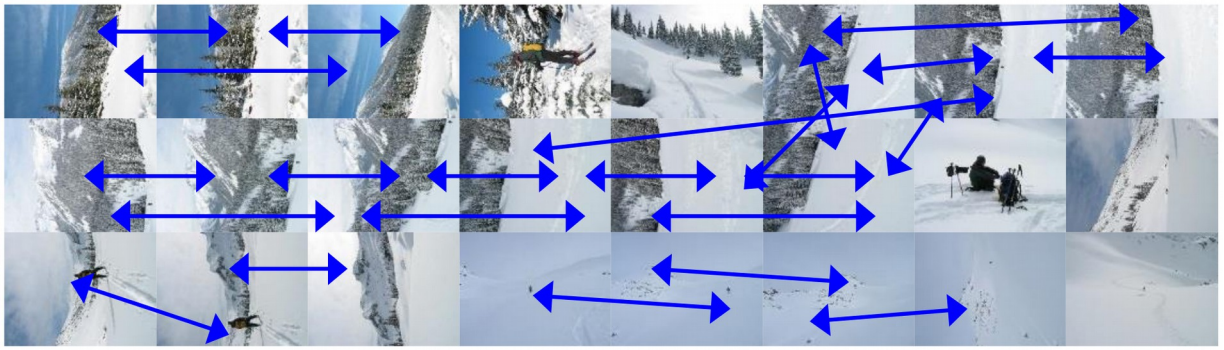
In the calibrated case, they can be recovered up to a similarity (scale) known as Euclidean/metric reconstruction

Multiple-View Structure from Motion

- Compute initial structure and motion
  - Hierarchical SfM
  - Incremental/Sequential SfM
- Refine Simultaneously structure and motion using Bundle Adjustment (BA)

Hierarchical SfM

1. Features are extracted and matched between nearby frames



2. Identify clusters consisting of three nearby frames:

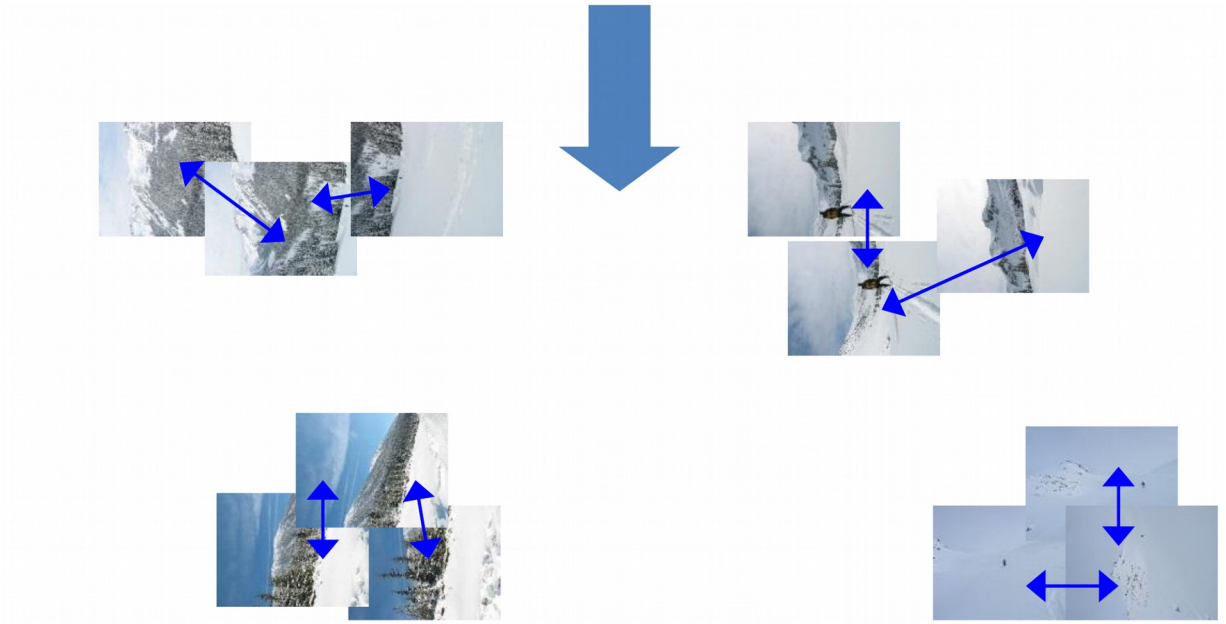


Image courtesy: ETH Zuerich

Hierarchical SfM

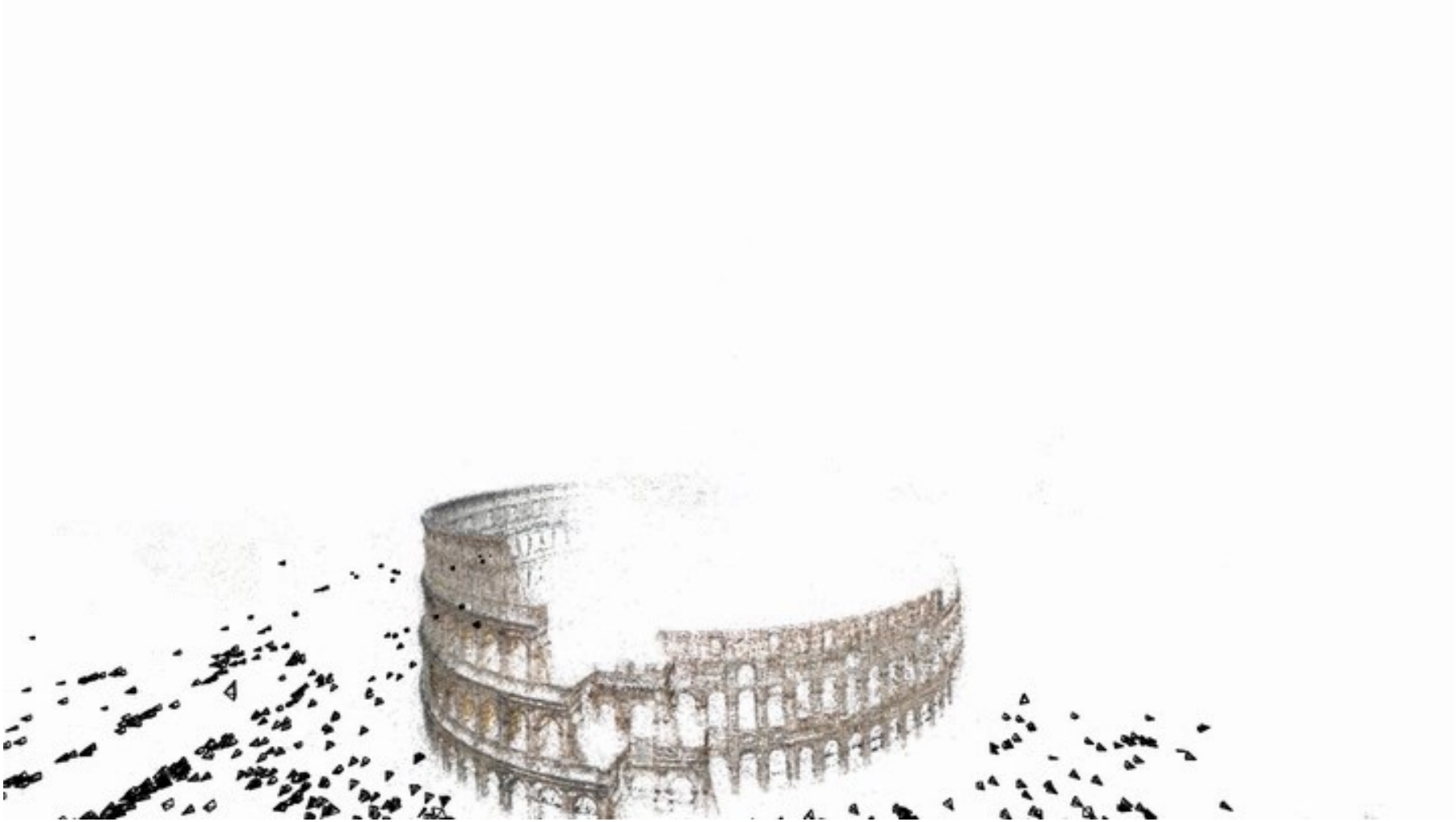
3. Compute SfM from three views
   - Compute SfM between first and second view and build point cloud
   - Then merge third view by running a 3-point RANSAC algorithm between point cloud and third view

4. Merge clusters pairwise and refine (BA) for both structure and motion

Example:

- Reconstruction from 150,000 images from Flickr associated with the tags “Rome”
- 4m 3D points. Cloud of 496 computers. 21 hours of computation.
- Paper: “Building Rome in a Day”, ICCV’09: http://grail.cs.washington.edu/rome/ University of Washington, 2009

Hierarchical SfM



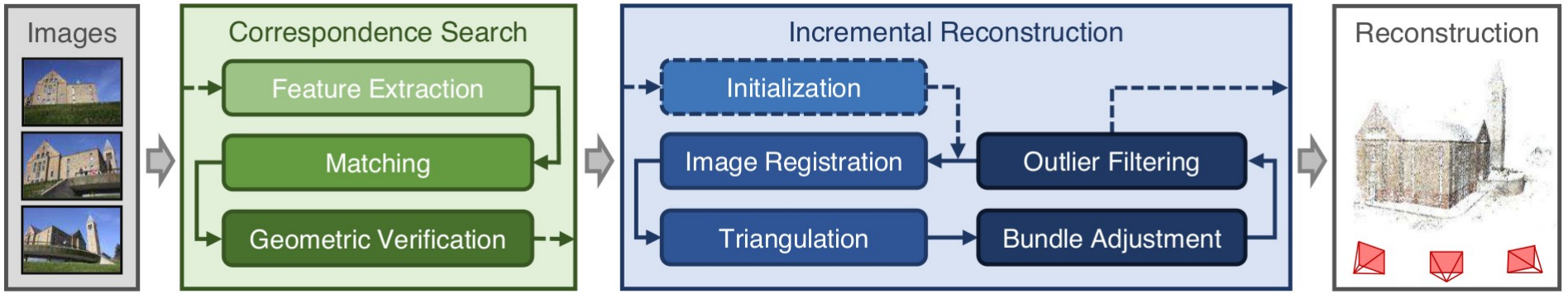
The Colosseum, 2,106 images, 819,242 points

Building Rome in a Day. Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz and Richard Szeliski. International Conference on Computer Vision, 2009, Kyoto, Japan.

Multiple-View Structure from Motion

- Compute initial structure and motion
  - Hierarchical SfM
  - Incremental/Sequential SfM
- Refine Simultaneously structure and motion using Bundle Adjustment (BA)

Incremental Structure from Motion



- **Correspondence search:** feature extraction, matching and geometric verification

  → scene graph (initialization - foundation for the reconstruction stage: seeds the model with a carefully selected two-view reconstruction)

- **Incremental reconstruction:**
  - incrementally registering new images
  - triangulating scene points
  - refining the reconstruction using bundle adjustment (BA)
  - filtering outliers

J. L. Schönberger and J. Frahm, "Structure-from-Motion Revisited," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 4104-4113.

Incremental SfM

Input to the incremental reconstruction: **scene graph** (with images as nodes and verified pairs of images as edges.

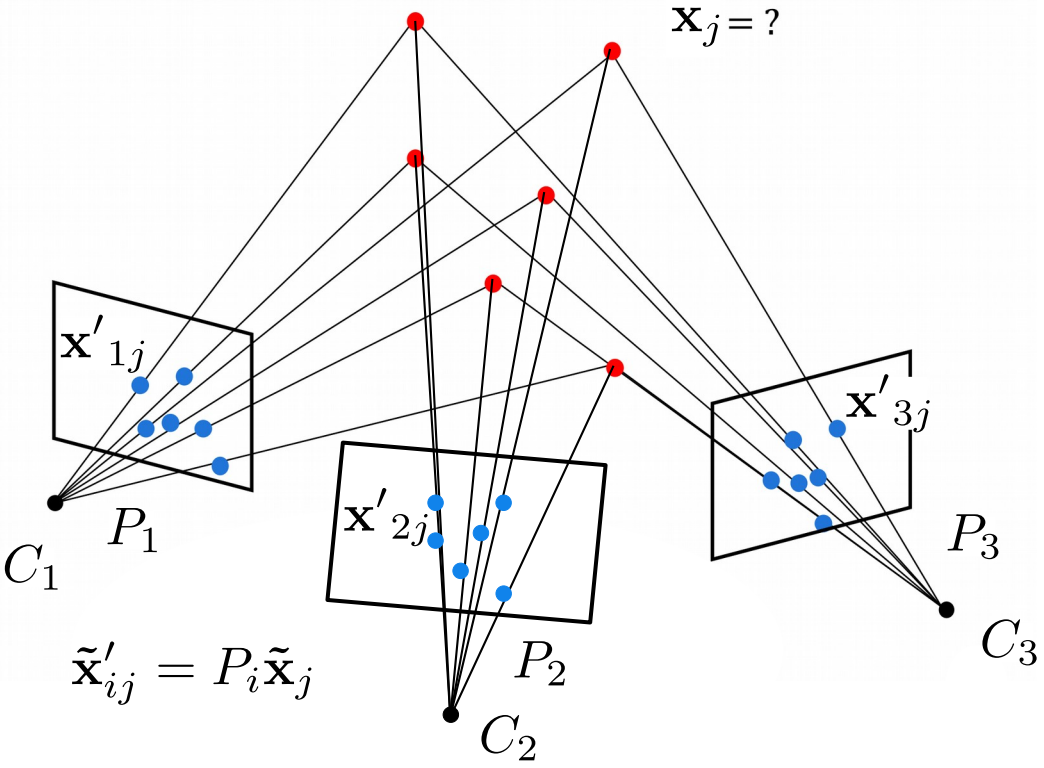
Output to the incremental reconstruction: pose estimates P for registered images and reconstructed scene structure as a set of points $\mathbf{x}$

Initialization of motion from two images

- $F \rightarrow (P_1, P_2)$
- $E \rightarrow (P_1, P_2) = (K_1[I|0], K_2[R|T])$

Note: choosing a *suitable* initial pair is critica
- reconstruction might never recover!

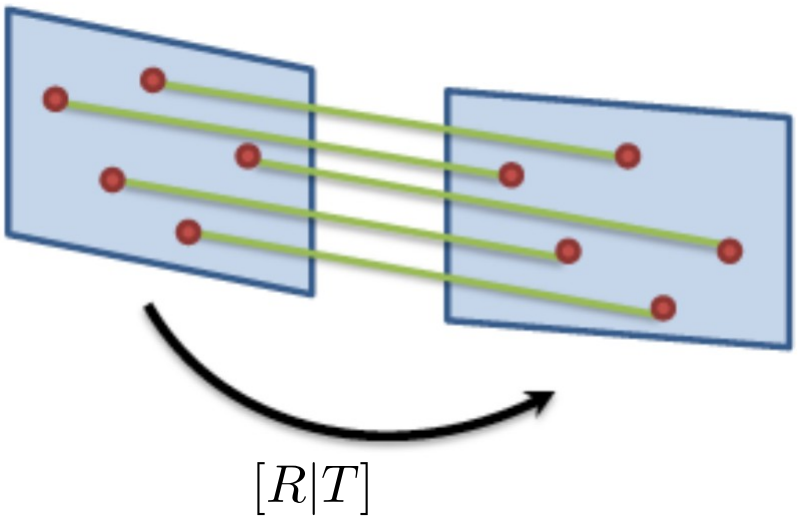
Initialization of 3D structure from triangulatio



Incremental SfM

Two view initialization:

- 5-point algorithm (minimal solver)
- 8-point linear algorithm
- 7-point algorithm



Note:

- **Initializing from a dense location** in the image graph with many overlapping cameras results in a more robust and accurate reconstruction due to increased redundancy
- **Initializing from a sparser location** results in lower run times, since BAs deal with overall sparser problems accumulated over the reconstruction process.

J. L. Schönberger and J. Frahm, "Structure-from-Motion Revisited," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 4104-4113.

Incremental SfM

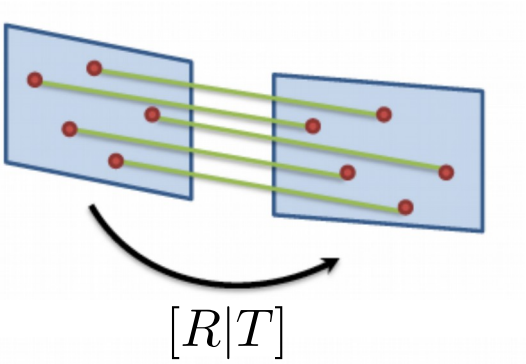
Input to the incremental reconstruction: scene graph (with images as nodes and verified pairs of images as edges.

Output to the incremental reconstruction: pose estimates P for registered images and reconstructed scene structure as a set of points x

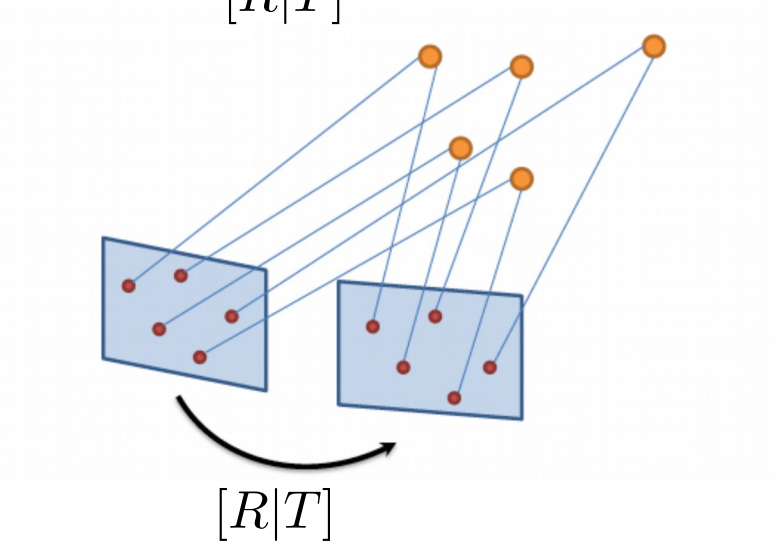
Initialization of motion from two images

- $F \rightarrow (P_1, P_2)$
- $E \rightarrow (P_1, P_2) = (K_1[I|0], K_2[R|T])$

Note: choosing a *suitable* initial pair is critical!
- reconstruction might never recover!



Initialization of 3D structure from triangulation

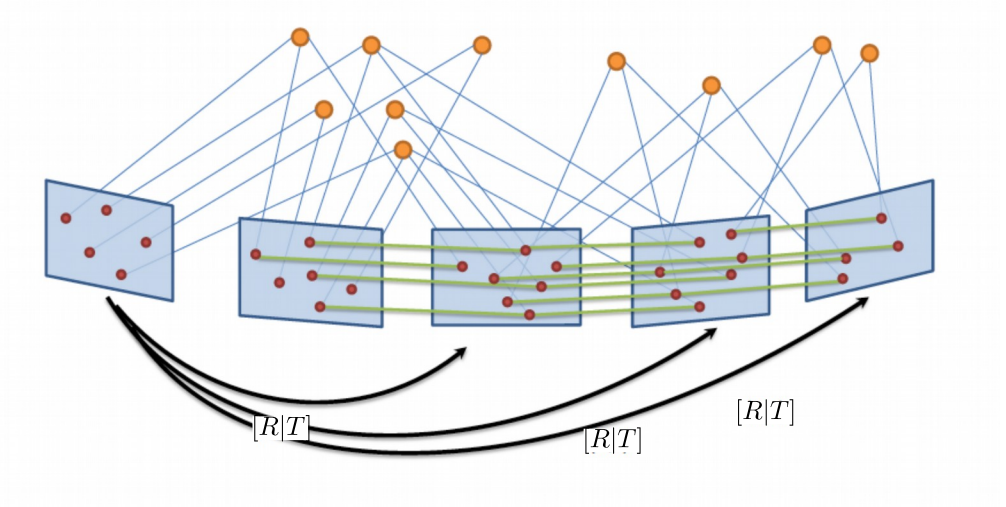


J. L. Schönberger and J. Frahm, "Structure-from-Motion Revisited," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 4104-4113.

Incremental SfM

For each additional view:

- **Image registration:**
  Determine the projection matrix P_i e.g from 2D-3D correspondences solving the PnP problem. The set P is thus extended by the pose of the newly registered image. (RANSAC for **outlier contamination**)



- **Triangulation:**
  Refine and extend the 3D structure. A newly registered image must observe existing scene points.
  - It may also increase scene overage by extending the set of points through triangulation
  - A new scene point can be triangulated and added as soon as at least one more image is registered
  - Note: Triangulation is a crucial step in SfM as it increases the stability of the existing model through redundancy and enables registration of new images by providing additional 2D-3D correspondences.

- **Bundle Adjustment:**
  The resulting **structure and motion** should be refined

Multiple-View Structure from Motion

- Compute initial structure and motion
  - Incremental/Sequential SfM
  - Hierarchical SfM
- Refine Simultaneously structure and motion using Bundle Adjustment (BA)

Bundle Adjustment (BA)

- Image registration and triangulation are separate procedures: **But** their products are highly correlated: **uncertainties in the camera pose propagate to triangulated points and vice versa**

  → additional triangulation may improve the initial camera pose through increased redundancy

  → without further refinement Sfm drifts quickly to a non-recoverable state

- BA is the joint non-linear refinement of camera parameters (motion) and point parameters (structure) that minimize the reprojection error

$$E = \sum_{i=1}^{m} r_i^2(\theta) = \sum_{i=1}^{m} \|\pi_i(g(\theta)) - \tilde{\mathbf{x}}_\mathbf{i}'\|^2$$

  with $(\pi_i(g(\theta)) - \tilde{\mathbf{x}}_\mathbf{i}') = (P\tilde{\mathbf{x}}_\mathbf{i} - \tilde{\mathbf{x}}_\mathbf{i}')$

  BA is used after linear estimation of R and T (e.g., after 8-point algorithm)

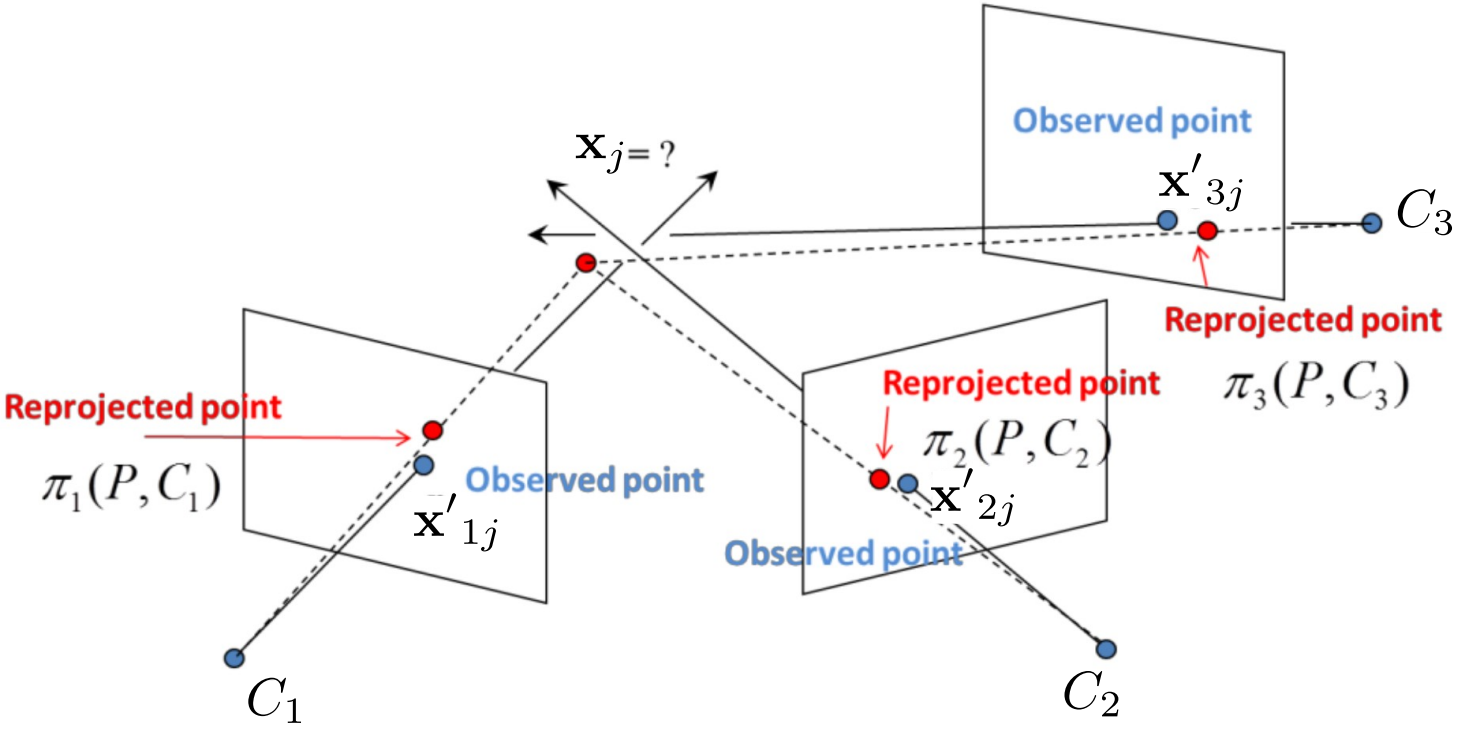
  Computes P_i by minimizing the Sum of Squared Reprojection errors.

- Can be minimized using Levenberg–Marquardt (more robust than Gauss-Newton to local minima)
- In order to not get stuck in local minima, the initialization should be close the minimum

Bundle Adjustment (BA) for Multiple-View

Minimizes the Sum of Squared Repojection Errors over each view

$$E = \sum_{i=1}^{m} r_i^2(\theta) = \sum_{i=1}^{m} \|\pi_i(g(\theta)) - \tilde{\mathbf{x}}_\mathbf{i}'\|^2$$



Robust Lost Function

To **prevent that large reprojection errors** which can negatively influence the optimization, a more robust norm ρ() is used instead of the L2-norm :

$$E = \sum_{i=1}^{m} \rho_i(\pi_i(g(\theta)) - \tilde{\mathbf{x}}_\mathbf{i}')$$

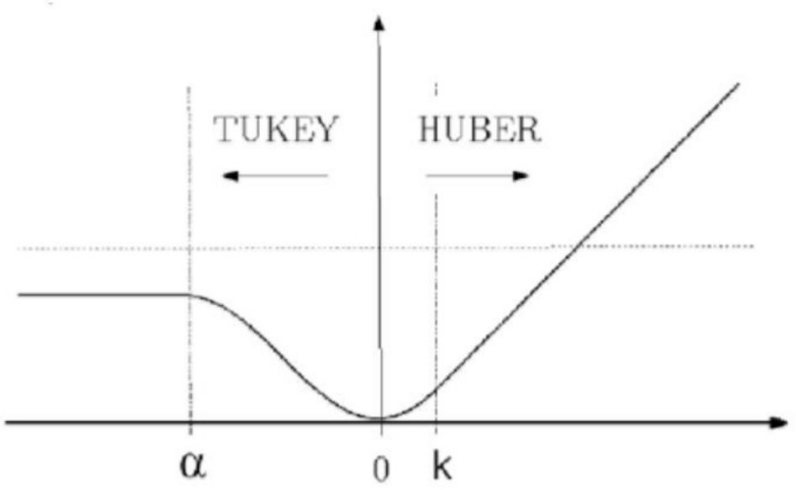
ρ() is a robust cost function (Huber or Tukey) to penalize wrong matches:

- **Huber norm**:

$$\rho(x) = \begin{cases} x^2 & \text{if } |x| \leq k \\ k(2|x| - k) & \text{if } |x| \geq k \end{cases}$$

- **Tukey norm**:

$$\rho(x) = \begin{cases} \alpha^2 & \text{if } |x| \geq \alpha \\ \alpha^2 \left(1 - \left(1 - \left(\frac{x}{\alpha}\right)^2\right)^3\right) & \text{if } |x| \leq \alpha \end{cases}$$



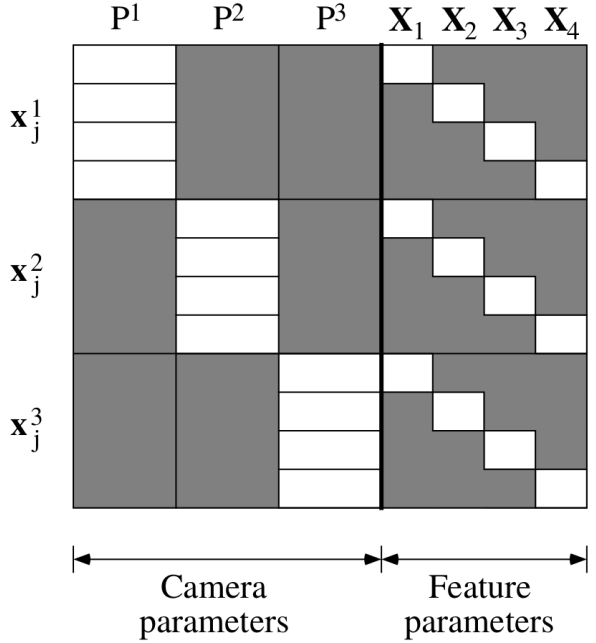
Sparse Bundle Adjustment

- For each iteration, iterative minimization methods need to determine a vector of changes to be made in the parameter vector
- In Levenberg-Marquardt each such step is determined from the normal equation

$$(J^\top J + \lambda diag(J^\top J))\Delta x = -J^\top r$$

  where J is the Jacobian matrix of the cost function and r is the vector of residuals (errors)

- For the bundle adjustment problem the Jacobian matrix has a sparse structure that can be exploited in computations



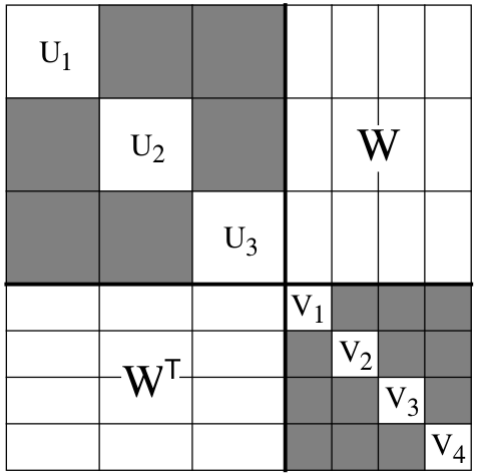


Fig. A6.1. *Form of the Jacobian matrix for a bundle-adjustment problem consisting of* 3 *cameras and* 4 *points.*

Hartley, Zissermann. Multiple-View Geometry

Efficient Bundle Adjustment

The Jacobian $\mathtt{J} = [\partial\hat{\mathbf{x}}/\partial\mathbf{P}]$ has a block structure of the form $\mathtt{J} = [\mathtt{A}|\mathtt{B}]$ with

$$\mathtt{A} = \left[\partial\hat{\mathbf{x}}/\partial\mathbf{a}\right]$$

$$\mathtt{B} = \left[\partial\hat{\mathbf{x}}/\partial\mathbf{b}\right]$$

The normal equations

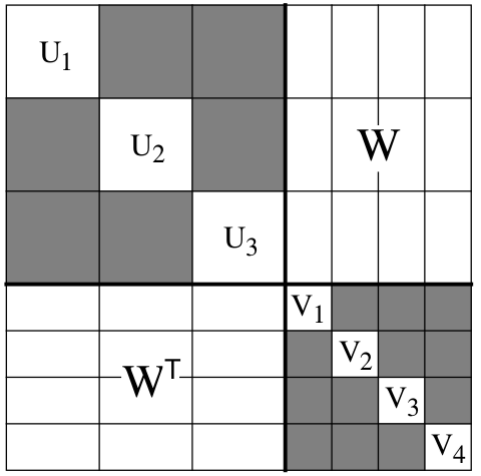
$$\mathtt{J}^\mathsf{T}\Sigma_\mathbf{X}^{-1}\mathtt{J}\boldsymbol{\delta} = \mathtt{J}^\mathsf{T}\Sigma_\mathbf{X}^{-1}\boldsymbol{\epsilon}$$

$$\left[\begin{array}{c|c} \mathtt{A}^\mathsf{T}\Sigma_\mathbf{X}^{-1}\mathtt{A} & \mathtt{A}^\mathsf{T}\Sigma_\mathbf{X}^{-1}\mathtt{B} \\ \hline \mathtt{B}^\mathsf{T}\Sigma_\mathbf{X}^{-1}\mathtt{A} & \mathtt{B}^\mathsf{T}\Sigma_\mathbf{X}^{-1}\mathtt{B} \end{array}\right] \left(\begin{array}{c} \boldsymbol{\delta}_\mathbf{a} \\ \hline \boldsymbol{\delta}_\mathbf{b} \end{array}\right) = \left(\begin{array}{c} \mathtt{A}^\mathsf{T}\Sigma_\mathbf{X}^{-1}\boldsymbol{\epsilon} \\ \hline \mathtt{B}^\mathsf{T}\Sigma_\mathbf{X}^{-1}\boldsymbol{\epsilon} \end{array}\right)$$

Sparse Bundle Adjustment

Rewrite

$$\begin{bmatrix} \mathtt{U}^* & \mathtt{W} \\ \mathtt{W}^\mathsf{T} & \mathtt{V}^* \end{bmatrix} \begin{pmatrix} \boldsymbol{\delta}_\mathbf{a} \\ \boldsymbol{\delta}_\mathbf{b} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\epsilon}_\mathtt{A} \\ \boldsymbol{\epsilon}_\mathtt{B} \end{pmatrix}$$



Multiply both sides on the left by

$$\begin{bmatrix} \mathtt{I} & -\mathtt{W}\mathtt{V}^{*-1} \\ 0 & \mathtt{I} \end{bmatrix}$$

Resulting in

$$\begin{bmatrix} \mathtt{U}^* - \mathtt{W}\mathtt{V}^{*-1}\mathtt{W}^\mathsf{T} & \mathtt{0} \\ \mathtt{W}^\mathsf{T} & \mathtt{V}^* \end{bmatrix} \begin{pmatrix} \boldsymbol{\delta}_\mathbf{a} \\ \boldsymbol{\delta}_\mathbf{b} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\epsilon}_\mathtt{A} - \mathtt{W}\mathtt{V}^{*-1}\boldsymbol{\epsilon}_\mathtt{B} \\ \boldsymbol{\epsilon}_\mathtt{B} \end{pmatrix}$$

The top half of this set of equations is

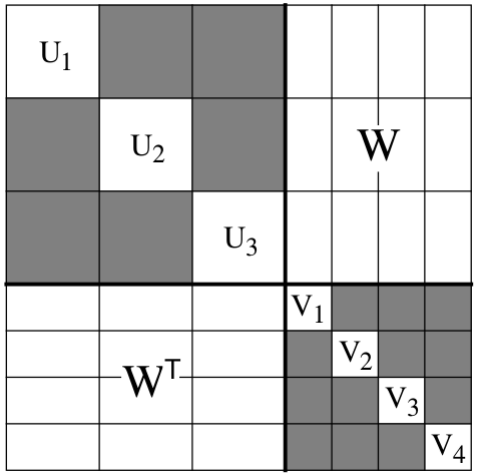
$$(\mathtt{U}^* - \mathtt{W}\mathtt{V}^{*-1}\mathtt{W}^\mathsf{T})\boldsymbol{\delta}_\mathbf{a} = \boldsymbol{\epsilon}_\mathtt{A} - \mathtt{W}\mathtt{V}^{*-1}\boldsymbol{\epsilon}_\mathtt{B}$$

These equations may be solved to find delta a. Subsequently the value of delta b may be found by back-substitution, given

$$\mathtt{V}^*\delta_\mathbf{b} = \boldsymbol{\epsilon}_\mathtt{B} - \mathtt{W}^\mathsf{T}\boldsymbol{\delta}_\mathbf{a}$$

Sparse Bundle Adjustment

Schur Complement trick



Block diagonal matrix is easy to invert

$$(\mathtt{U}^* - \mathtt{W}\mathtt{V}^{*-1}\mathtt{W}^\mathsf{T})\boldsymbol{\delta}_\mathbf{a} = \boldsymbol{\epsilon}_\mathtt{A} - \mathtt{W}\mathtt{V}^{*-1}\boldsymbol{\epsilon}_\mathtt{B}$$

Small number of camera parameters also inexpensive easy to invert

$$\mathtt{V}^*\delta_\mathbf{b} = \boldsymbol{\epsilon}_\mathtt{B} - \mathtt{W}^\mathsf{T}\boldsymbol{\delta}_\mathbf{a}$$

Bundle Adjustment - Literature

S. Agarwal et al, Building Rome in a Day, 2011
- Cluster of 62-computers
- 150 000 unorganized images from Rome
- ~37 000 image registered
- Total processing time ~21 hours
- SfM time ~7 hours

J. Heinly et al, Reconstructing the World in Six Days, 2015
- 1 dual processor PC with 5 GPU's (CUDA)
- ~96 000 000 unordered images spanning the globe
- ~1.5 000 000 images registered
- Total processing time ~5 days
- SfM time ~17 hours

SBA – Sparse Bundle Adjustment
- A generic sparse bundle adjustment C/C++
- package based on the Levenberg-Marquardt algorithm
- Code (C and Matlab mex) available at http://www.ics.forth.gr/~lourakis/sba/
- CVSBA is an OpenCV wrapper for SBA www.uco.es/investiga/grupos/ava/node/39/

Bundle Adjustment - Literature

Ceres
- By Google (used in production since 2010)
- A C++ library for modeling and solving large, complicated optimization problems like SfM
- Homepage: www.ceres-solver.org
- Code available on GitHub https://github.com/ceres-solver/ceres-solver

GTSAM – Georgia Tech Smoothing and Mapping
- – A C++ library based on factor graphs that is well
- suited for SfM ++
- – Code (C++ library and Matlab toolbox) available at
- https://borg.cc.gatech.edu/

g 2 o – General Graph Optimization
- Open source C++ framework for optimizing graph-
- based nonlinear error functions
- https://openslam-org.github.io/g2o.html
- Code available on GitHub https://github.com/RainerKuemmerle/g2o

Bundle Adjustment - Literature

Bundler

- A structure from motion system for unordered image collections written in C and C++
- SfM based on a modified version SBA (default) or Ceres
- Homepage: http://www.cs.cornell.edu/~snavely/bundler/
- Code available on GitHub https://github.com/snavely/bundler_sfm

RealityCapture

- A state-of-the-art photogrammetry software that automatically extracts accurate 3D models from images, laser-scans and other input
- Homepage: https://www.capturingreality.com/

VisualSfM

- A GUI application for 3D reconstruction using structure from motion
- Output works with other tools that performs dense 3D reconstruction
- Homepage: http://ccwu.me/vsfm/

Passive Stereo

Passive (traditional) stereo main idea:
use corresponding points to estimate the location of a 3D point by triangulation.

Key challenge:
correspondence problem: how to know whether a point actually corresponds to a point in another image?

In the following: alternative techniques for 3D scene reconstruction

Multi-View Stereo Reconstruction

Approaches to Multiple-view Stereo

- Active stereo
- Can be very accurate

- Volumetric stereo
  - View-independent representation
  - Need silhouette extraction
  - Accuracy depends on the density of the grid
  - High computational and memory costs

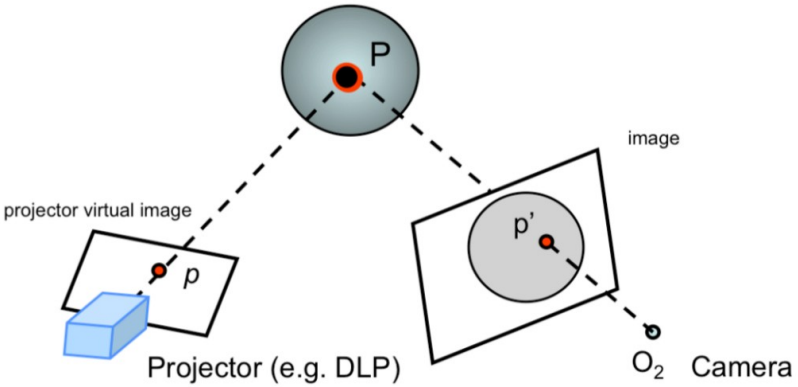
- Surface expansion
  - Sparse to dense feature correspondences

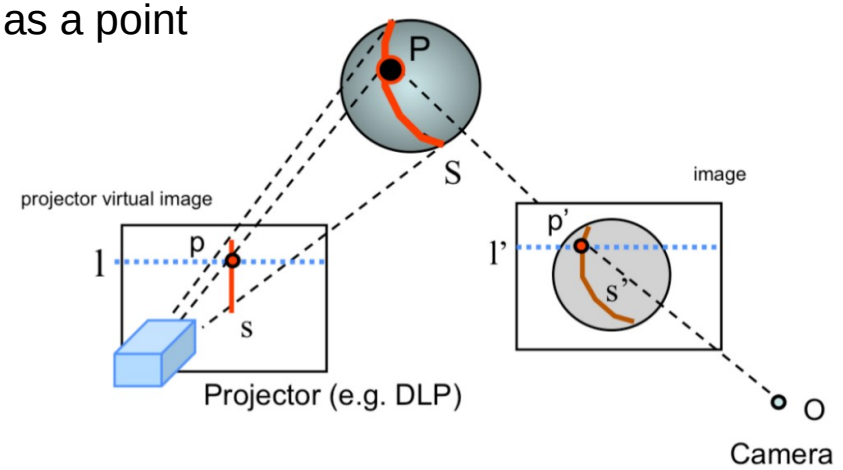
Active Stereo

Mitigate the correspondence problem in traditional stereo.

Main idea: replace one of the two cameras with a device that interacts with the 3D environment

- usually by projecting a pattern onto the object that is easily identifiable from the second camera.
- This new projector-camera pair defines the same epipolar geometry that we introduced for camera pairs, whereby the image plane of the replaced camera is replaced with a projector virtual plane.
- The projector is used to project a point in the virtual plane onto the object in 3D space, producing a point in 3D space
- This 3D point should be observed in the second camera as a point
- Because we know what we are projecting (e.g. the position of p in the virtual plane, the color and intensity of the projection, etc.), we can easily discover the corresponding observation in the second camera.



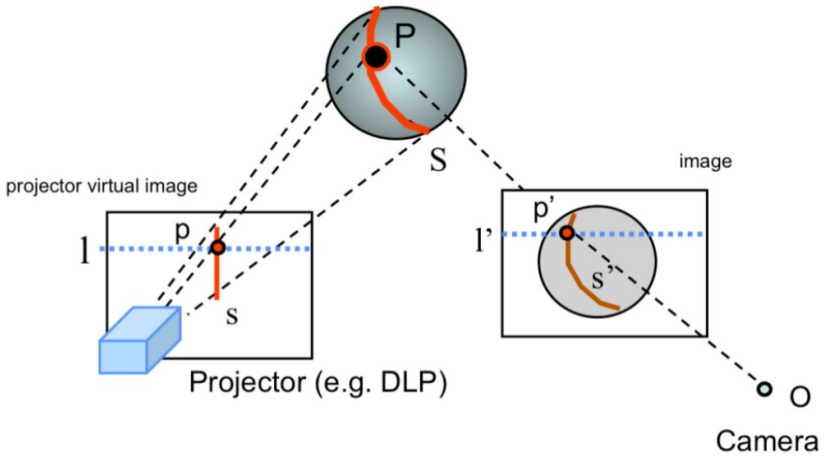


Courtesy: Kenji Hata and Silvio Savarese. Active and Volumetric Stereo

Active Stereo

Strategy:
1. Project from the virtual plane a vertical stripe instead of a single point.
2. Very similar to the point case, where the line is projected to a stripe in 3D space and observed as a line in the camera.
3. If the projector and camera are parallel or rectified,then we can discover the corresponding points easily by simply intersecting the points with the horizontal epipolar lines.
4. From the correspondences, use triangulation to reconstruct all the 3D points on the stripe.
5. By swiping the line across the scene and repeating the process, the entire shape of all visible objects in the scene is recovered.

Note: the projector and the camera need to be calibrated! The camera is calibrated by using a calibration rig. Then, by projecting known stripes onto the calibration rig, and using the corresponding observations in the newly calibrated camera, set up constraints for estimating the projector intrinsic and extrinsic parameters.



Active Stereo

Less-costly alternative to the projector:

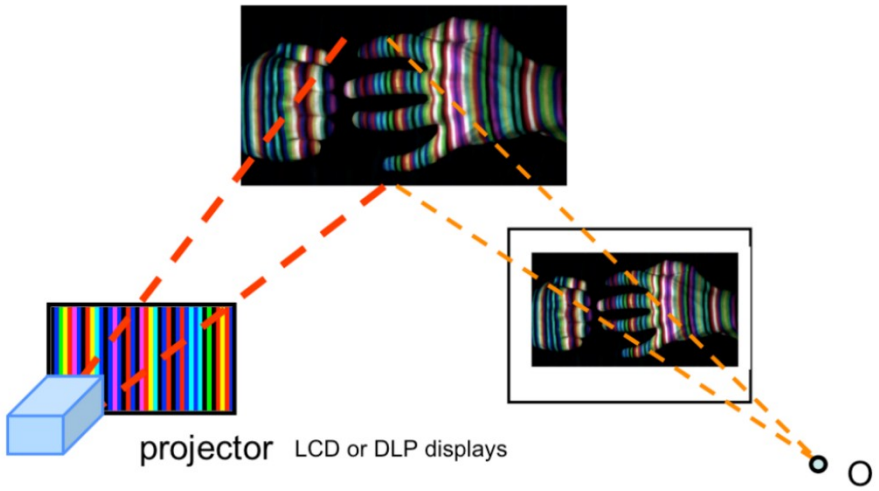
- Leverage shadows to produce active patterns to the object we want to recover.
- Place a stick between the object and a light source at a known position → project a stripe onto the object
- Moving the stick allows us to project different shadow stripes onto the object and recover the object in a similar manner as before.
- Much cheaper, but less accurate (requires very good calibration between the stick, camera, and light source, while needing to maintain a tradeoff between the length and thinness of the stick's shadow)

Active Stereo

- projecting a single stripe onto objects is that it is slow, as the projector needs to swipe across the entire object.
- It cannot capture deformations in real time.

Extension:

- project a known pattern of different stripes to the entire visible of the object, instead of a single stripe.
- The colors of these stripes are designed in such a way that the stripes can be uniquely identified from the image.
- This concept powered many versions of modern depth sensors, such as the original version of the Microsoft Kinect.
- In practice, these sensors use infrared laser projectors , which allow it to capture video data in 3D under any ambient light conditions.



Courtesy: Kenji Hata and Silvio Savarese. Active and Volumetric Stereo

Volumetric Stereo

Alternative approach to both traditional stereo and active stereo

Volumetric stereo inverts the problem of using correspondences to find 3D structure

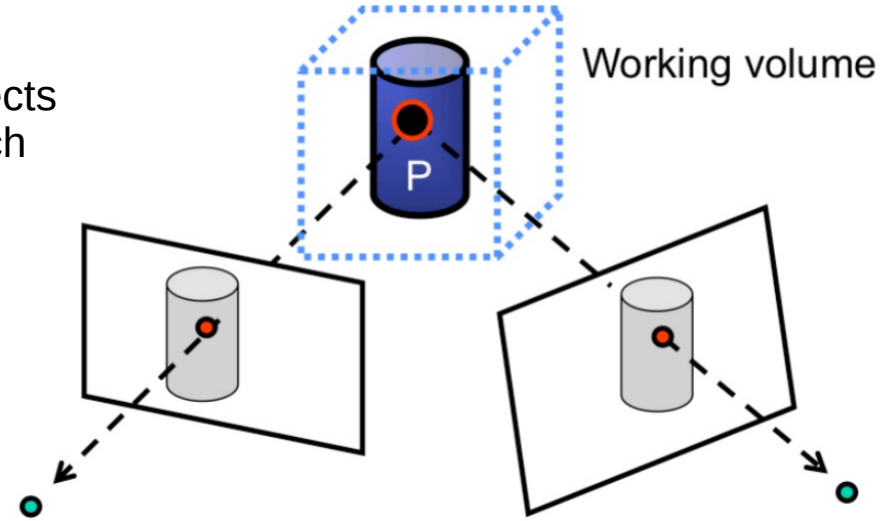
Assumption: 3D point we are try to estimate is within a contained, known volume; volume is discretized as a 3D voxel grid

Aim: Assign RGB values to voxels photo-consistent with images

- Project the hypothesized 3D point back into the calibrated cameras and validate whether these projections are consistent across the multiple views

  → A photo-consistent scene exactly reproduces input images from the same viewpoints

- used for recovering the 3D models of specific objects as opposed to recovering models of a scene, which may be unbounded



Courtesy: Kenji Hata and Silvio Savarese. Active and Volumetric Stereo

Volumetric Stereo – Consistent Observations

Volumetric stereo methods tend to first define what it means to be “consistent” when a 3D point is reprojected in the contained volume back into the multiple image views.

Techniques for consistent observations:

- space carving
- shadow carving
- voxel coloring

Volumetric Stereo - Space Carving

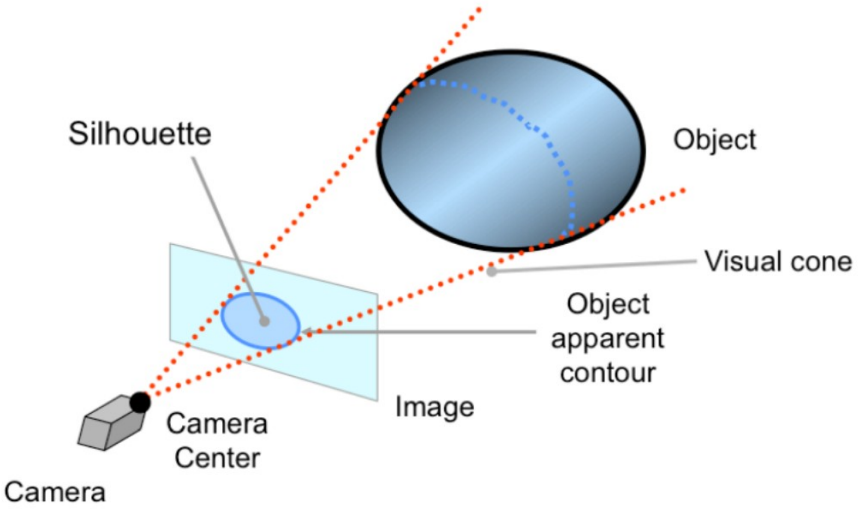
Idea: The contours of an object provide a rich source of geometric information about the object.

Each camera observes some visible portion of an object, from which a contour can be determined.

When projected into the image plane, this contour encloses a set of pixels known as the **silhouette of the object** in the image plane.

Space carving uses the silhouettes of objects from multiple views to enforce consistency.

Concept of a visual cone:
It is the enveloping surface defined by the camera center and the object contour in the image plane. By construction, it is guaranteed that the object will lie completely in both the initial volume and the visual cone.



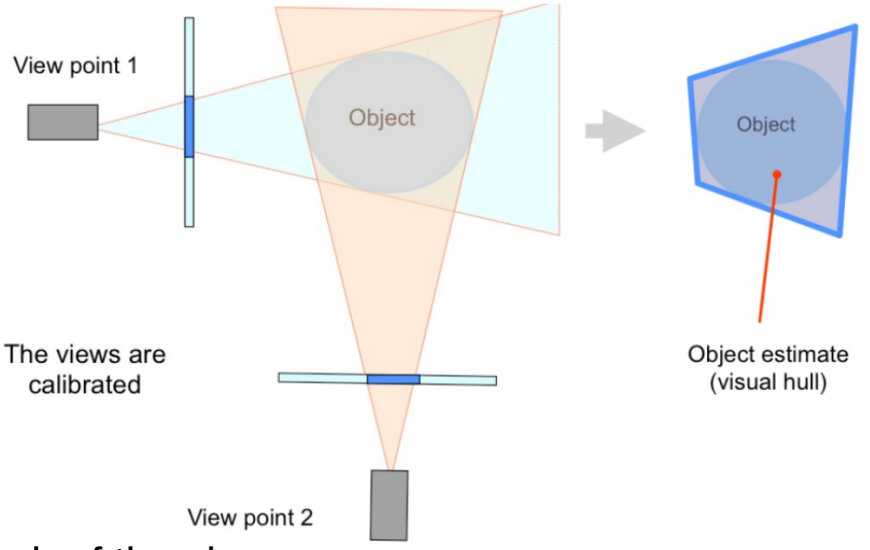
Courtesy: Kenji Hata and Silvio Savarese. Active and Volumetric Stereo

Volumetric Stereo – Visual Hull

Multiple views: compute **visual cones** for each view.

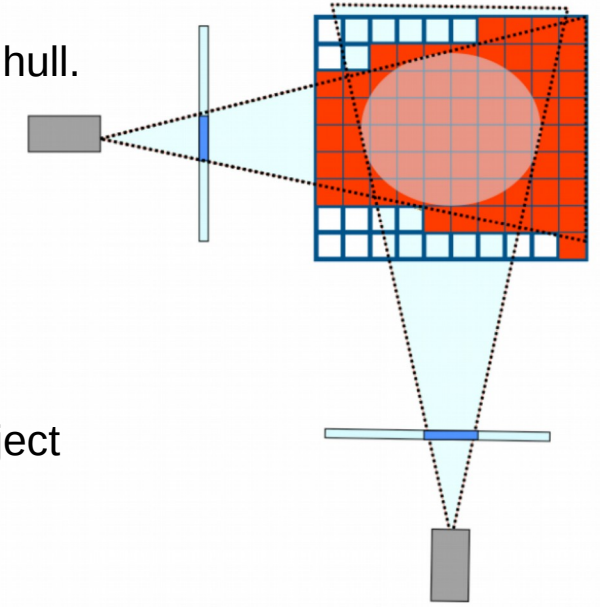
If the object resides in each of these visual cones,
then it must lie in the intersection of these visual cones

→ Such an intersection is called a **visual hull.**



- Divide this volume into small units known as voxels
  → **voxel grid.**
- Take each voxel in the voxel grid and project it into each of the views.
- If the voxel is not contained by the silhouette in a view, then it is discarded.

Outcome: Voxels that are contained within the visual hull.



Limitations:
- Scales linearly with the number of voxels
- Costly → Octrees can mitigate this problem
- Depends on number of views
- Incapable of modeling certain concavities of an object

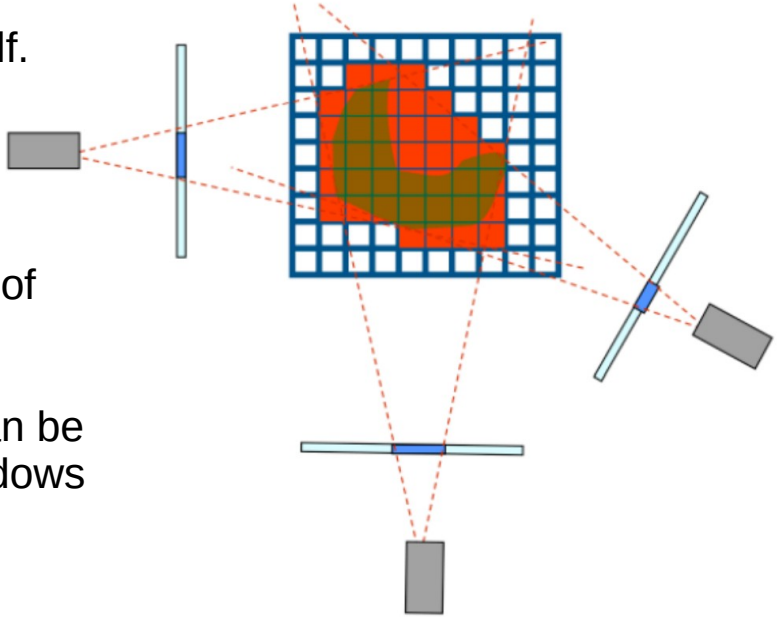
Courtesy: Kenji Hata and Silvio Savarese. Active and Volumetric Stereo

Volumetric Stereo – Volume Intersection

Self-shadows are the shadows that an object projects on itself.

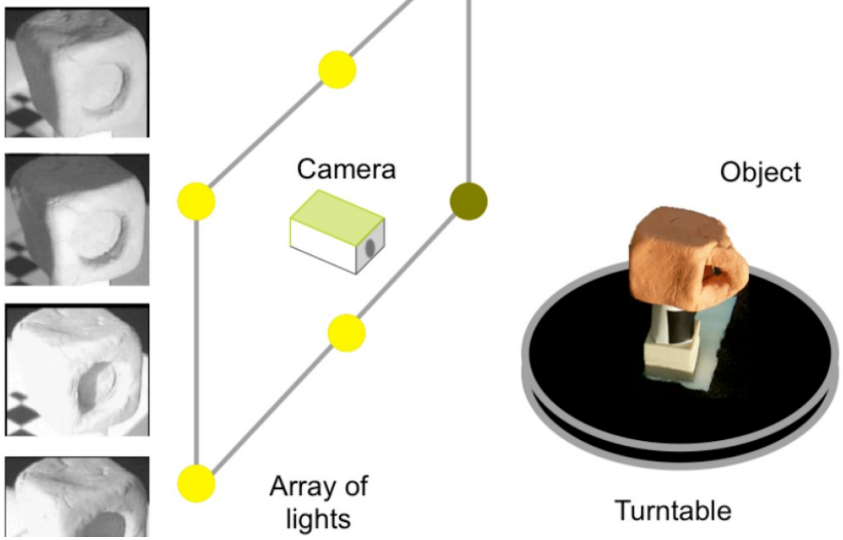
For the case of concave objects, an object will often cast self-shadows in the concave region.

Idea: Shadow carving augments space carving with the idea of using self-shadows to better estimate the concavities.



- An array of lights in known positions around the camera can be turned on and off → used to make the object cast self-shadows

- Start with an initial voxel grid
- In each view turn on and off each light in the array surrounding the camera.
- Each light will produce a different self-shadow on the object.
- identify the shadow in the image plane
  → find the voxels that are in the visual cone of the shadow.
- A voxel that is part of both visual cones cannot be part of the object to eliminate voxels in the concavity.



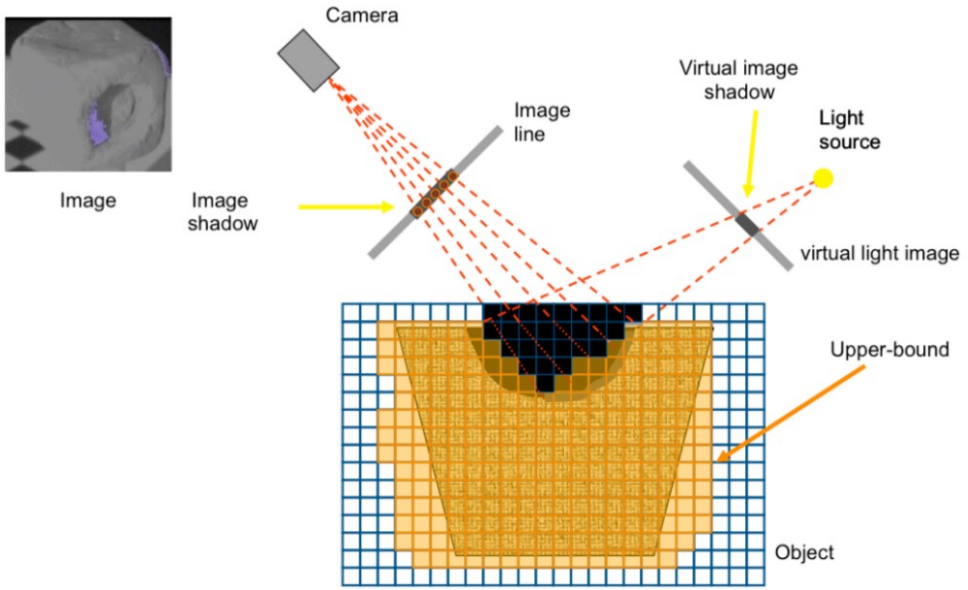
Courtesy: Kenji Hata and Silvio Savarese. Active and Volumetric Stereo

Volumetric Stereo - Shadow Carving

- Shadow carving relies on a new consistency check that removes voxels that are in the self-shadow visual cone of the camera and the visual cone of the light.

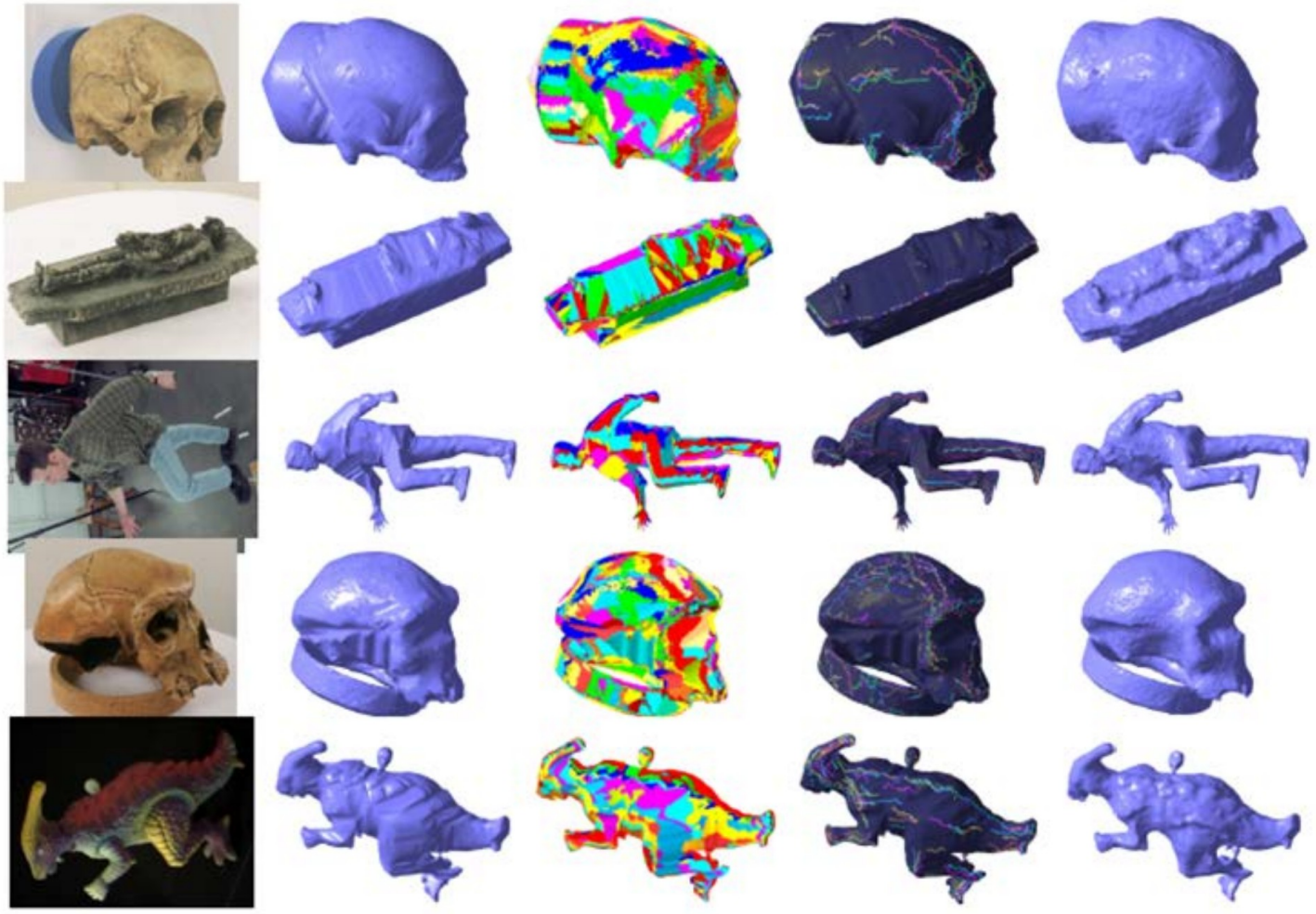
- Results depends on both the number of views and the number of light sources.

- Disadvantages:
  cannot handle cases where the object contains reflective or low albedo regions



Courtesy: Kenji Hata and Silvio Savarese. Active and Volumetric Stereo

Volumetic Stereo - Carved visual hulls



Yasutaka Furukawa and Jean Ponce, Carved Visual Hulls for Image-Based Modeling, ECCV 2006.

Volumetric Stereo – Voxel Colouring

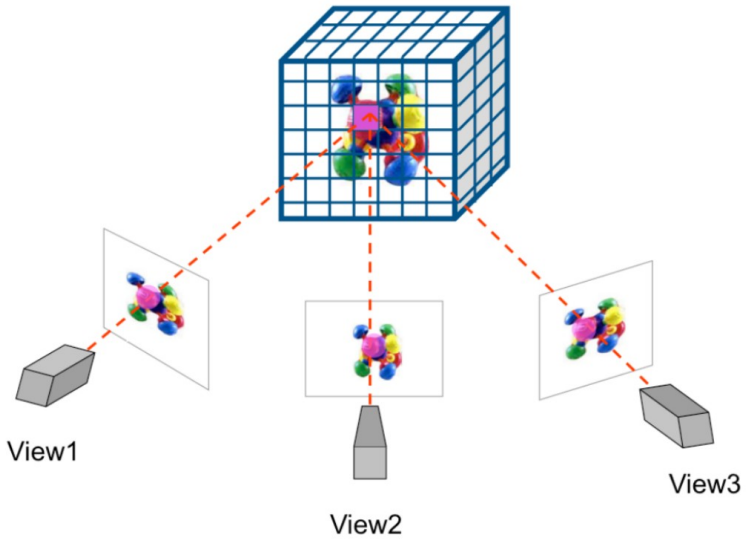
Idea: Use color consistency instead of contour consistency in space carving.

- Given are images from multile views of an object that we want to reconstruct.

- For each voxel, we look at its corresponding projections in each of the images and compare the color of each of these projections.
- If the colors of these projections match then we mark the voxel as part of the object.

Color conistency check: set a threshold between the color similarity between the projections.

Critical assumption for any color consistency check: the object being reconstructed must be **Lambertian**, which means that the perceived luminance of any part of the object does not change with viewpoint location or pose.



Courtesy: Kenji Hata and Silvio Savarese. Active and Volumetric Stereo

Surface Expansion

Extract features and acquire a sparse set of initial matches

Iteratively expand matches to nearby locations

Use visibility constraints to filter out false matches

Perform surface reconstruction

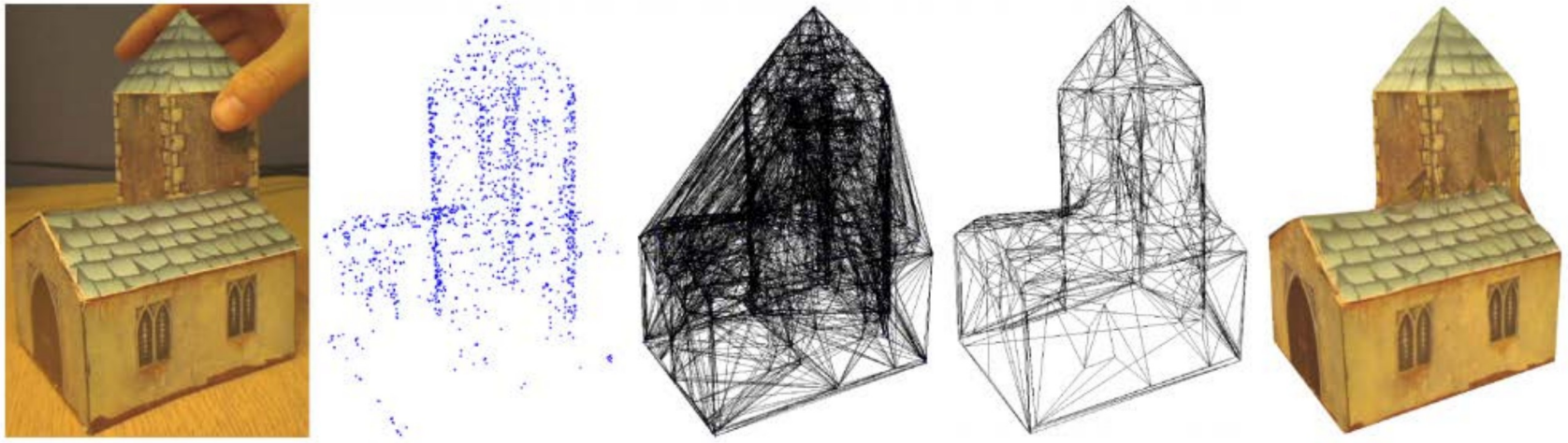
Yasutaka Furukawa and Jean Ponce, Accurate, Dense, and Robust Multi-View Stereopsis, CVPR 2007.

Surface reconstruction from point clouds

Exploit the known imaging geometry

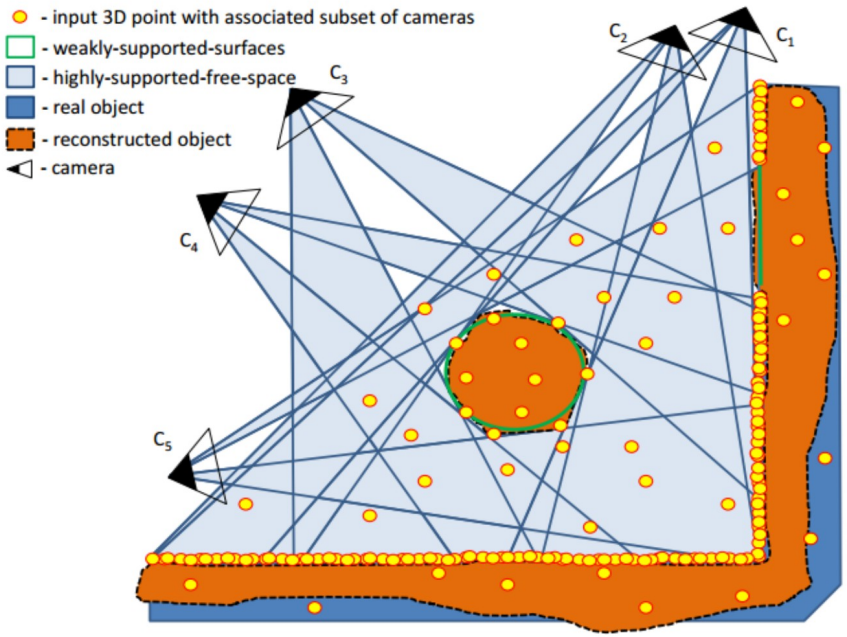
Remove surfaces that we see through

A kind of visual hull and space carving for point clouds



Qi Pan, Gerhard Reitmayr, Tom Drummond, ProFORMA: Probabilistic Feature-based On-line Rapid Model Acquisition, BMVC 2009.

Surface reconstruction from point clouds



Michal Jancosek and Tomas Pajdla, Multi-View Reconstruction Preserving Weakly-Supported Surfaces, CVPR 2011.

Summary

Three-view Geometry

- Trifocal Tensor

N-view Geometry

Stereo Multiple-view

Alternative Approaches to Multiple View Stereo
- Active Stereo
- Volumetric Stereo
- Surface Expansion
- Surface Reconstruction from Point Clouds

Understanding Check L07 and L09

What's the minimum number of correspondences required for calibrated SFM and why?

Are you able to derive the epipolar constraint?

Are you able to define the essential matrix?

Are you able to derive the 8-point algorithm?

How many rotation-translation combinations can the essential matrix be decomposed into?

Are you able to provide a geometrical interpretation of the epipolar constraint?

Are you able to describe the relation between the essential and the fundamental matrix?

Why is it important to normalize the point coordinates in the 8-point algorithm?

Describe one possible way to achieve this normalization.

Are you able to describe the normalized 8-point algorithm?

Why do we need RANSAC?

After how many iterations can RANSAC be stopped to guarantee a given success probability?

How do we apply RANSAC to the 8-point algorithm, DLT?

How can we reduce the number of RANSAC iterations for the SFM problem?

Why do we need Bundle Adjustment?

Are you able to define Bundle Adjustment (via mathematical expression and illustration)?

Are you able to describe hierarchical and sequential SFM for monocular VO?

Are there alternative approaches to passive stereo vision, can you name some and roughly describe the idea behind it?

Literature

- HZ Chapter 15.1
- HZ Chapter A6.3
- HZ Chapter A6.6