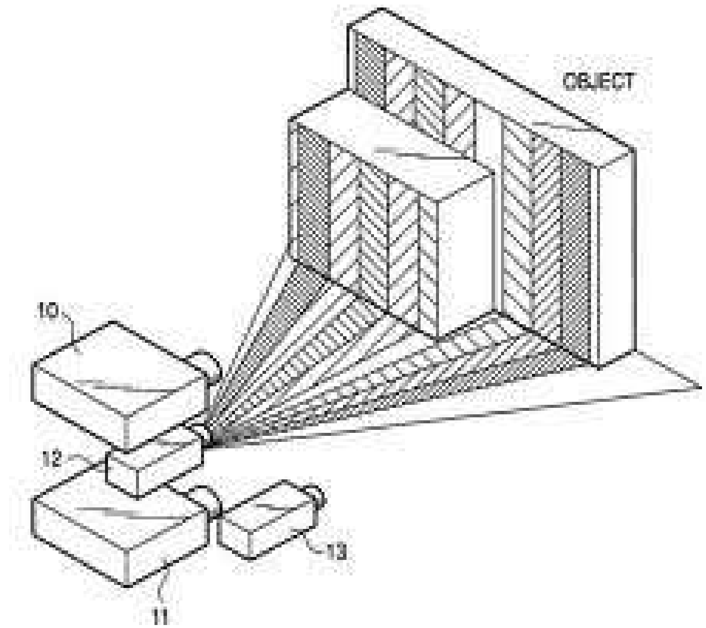
Structured Lighting

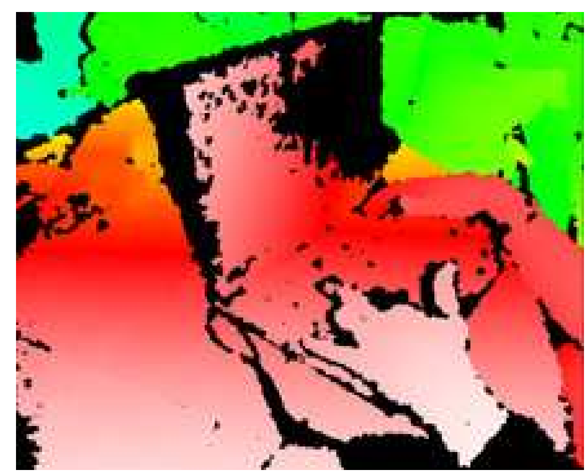
- Finding correspondences is hard by itself
- Can we help it by projecting patterns onto the world? **Structured Lights!**
- Lightstrip range finders, etc.
- Combination of sinusoids sometimes to get dense matches
- *Active vision*, as it changes the appearance
- The light projected need not be in the visible spectrum



Xbox Kinect

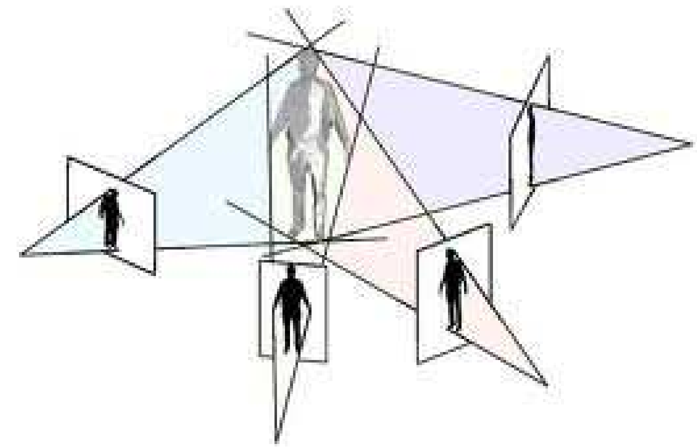
IR-based range sensor for Xbox

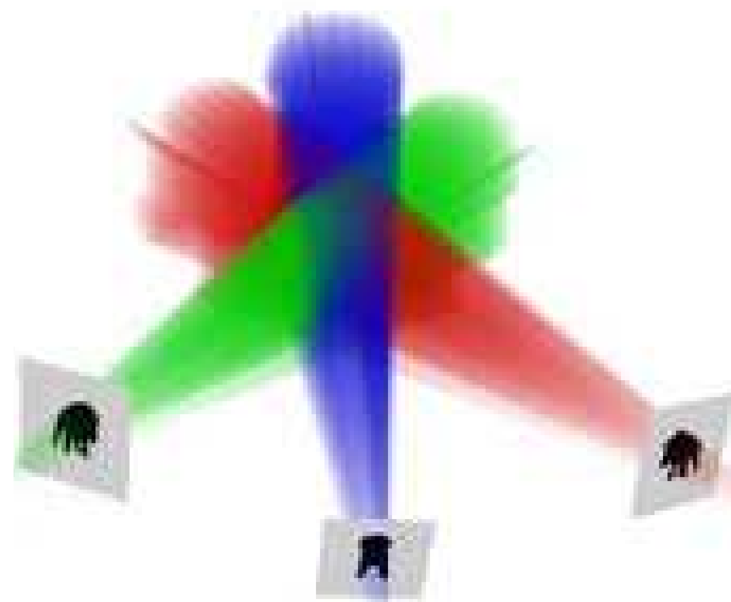
- Aligned depth and RGB images at 640×480
- Original goal: Interact with games in full 3D
- Computer vision happy with real-time depth and image
  - Games, HCI, etc
  - Action recognition
  - Image based modelling of dynamic scenes
- Fastest selling electronic appliance ever!!
- Other products that use PrimeSense sensor



Visual Hull

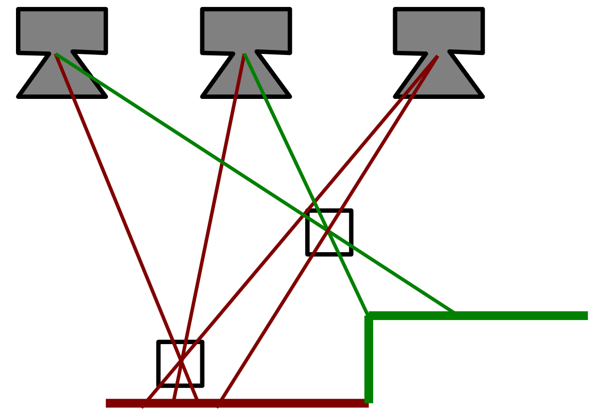
- Object silhouette represents a generalized cone with the camera centre as the apex
- Intersect these cones for multiple views in the 3D space
- **Visual Hull**, like convex hull
- Cannot get fine details like concavities
- Gives a very good, approximate shape, without scene modification!





Space Carving

- Reason directly in a volumetric **voxel** space
- If a voxel is filled, it projects to similar colours in all cameras
- If a voxel is empty, its projections will have different appearances
- *Colour consistency:* filled or empty?
- Assume all filled initially; carve out empty ones by going over the images, guessing visibility, etc.



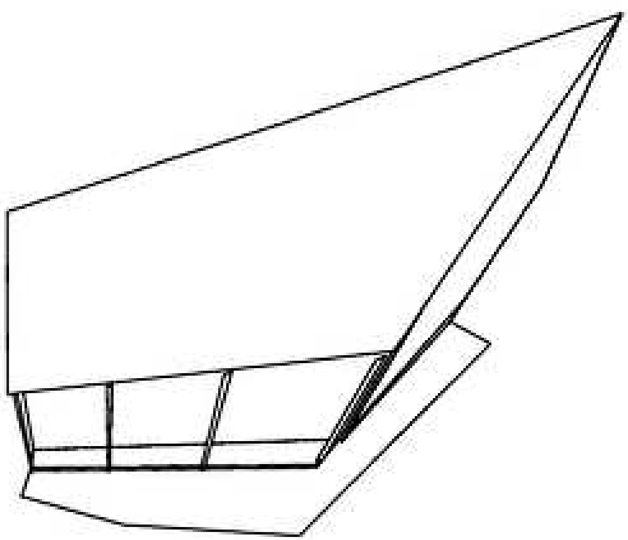
Stratified 3D Reconstruction

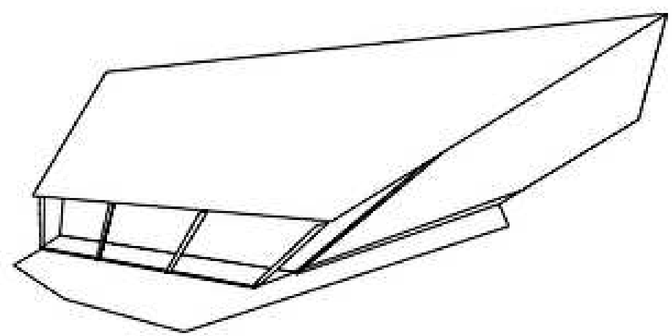
Structure from Two Views

- Estimate fundamental matrix from point matches
- Decompose $\mathbf{F}$ into canonical cameras: $\mathbf{P} = [\mathbf{I}|\mathbf{0}]$ and $\mathbf{P}' = [\mathbf{M}|\mathbf{m}]$. Epipole $\mathbf{e}'$ is the left null-space of $\mathbf{F}$. Second camera centre is: $\mathbf{C}' = -\mathbf{M}^{-1}\mathbf{e}'$
- Rays for matching $(\mathbf{x}, \mathbf{x}')$ are: $[0\ 0\ 0\ 1]^{\mathbf{T}} + \lambda[\mathbf{x}^{\mathbf{T}}\ \ \mathbf{0}]^{\mathbf{T}}$ and $\mathbf{C}' + \gamma[(\mathbf{P}'^{+}\mathbf{x}')^{\mathbf{T}}\ \ 0]^{\mathbf{T}}$. They meet at a 3D point $\mathbf{X}$ as they lie on a plane
- $\mathbf{X}$ can be found by equating the two equations to solve for λ, γ (3 equations in 2 unknowns).

The structure recovered is *projective*. Points $\mathbf{X}$ are within an unknown, general, 4×4 projective transformation of the true Euclidean structure

Projective Structure of House





3D Projective Geometry

- Points represented using 4-vectors $\mathbf{X} = [x_1 \; x_2 \; x_3 \; x_4]^{\mathbf{T}}$
- Planes also represented by 4-vectors, such that $\pi^{\mathbf{T}}\mathbf{X} = \mathbf{0}$ is the plane equation.
- Lines represented using two 4-vectors: line joining two points or intersection of two plans.
- Quadrics are general order-2 surfaces: $\mathbf{X}^{\mathbf{T}}\mathbf{Q}\mathbf{X} = \mathbf{0}$
- Lines have points at infinity. Points at infinity on a plane lie on the line at infinity for it. All points/lines at infinity lie on the **plane at infinity** denoted by π_∞

Hierarchy of Transformations

- Plane at infinity π_∞ is **fixed** for, and only for, **affine** transformations.
- The **absolute conic** Ω_∞ on π_∞ is the intersection of all spheres with π_∞. In canonical coordinates, $x_1^2 + x_2^2 + x_3^2 = 0 = x_4$. Thus, $\mathbf{\Omega}_\infty = \mathbf{I}$ on π_∞.
- **Absolute dual quadric** $\mathbf{Q}_\infty^*$ is defined using planes. Planes enveloping it meet π_∞ in lines that are tangent to Ω_∞.

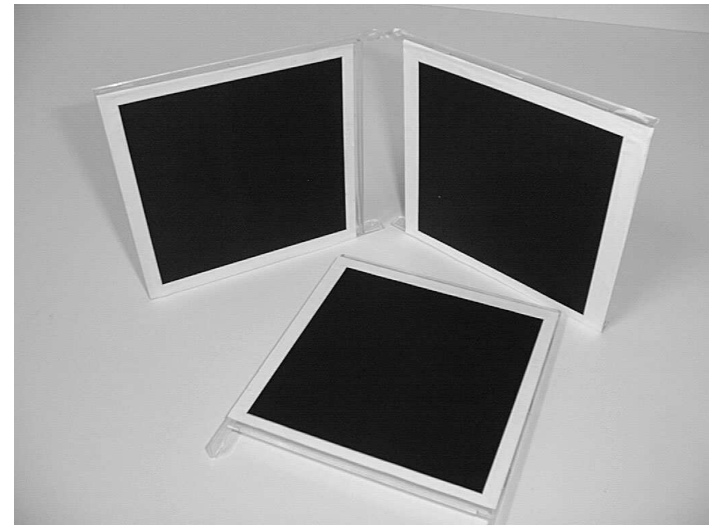
  $\mathbf{Q}_\infty^*$ in Euclidean/canonical coords: $\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0}^\mathbf{T} & 0 \end{bmatrix}$
- $\mathbf{Q}_\infty^*$ and $\mathbf{\Omega}_\infty$ are **fixed** under **similarity** transformations.

Image of the Absolute Conic

- A camera projects points $[\mathbf{d} \;\; \mathbf{0}]^{\mathbf{T}}$ on π_∞ to $\mathbf{x} = \mathbf{KRd}$. Thus, $\mathbf{KR}$ is the homography from π_∞ to the image.
- Image of the Absolute Conic (IAC) is given by: $\omega = (\mathbf{KK^T})^{-1} = \mathbf{K^{-T}K^{-1}}$. Dual-IAC is: $\omega^* = KK^{\mathbf{T}}$
  $\omega = \mathbf{H^{-T}\Omega_\infty H^{-1}} = \mathbf{(KR)^{-T}I(KR)^{-1}} = \mathbf{K^{-T}RR^{-1}K^{-1}}$ as $\mathbf{\Omega_\infty} = \mathbf{I}$ on π_∞.
- Properties of IAC ω
  - ω depends only on $\mathbf{K}$ and not on $\mathbf{R}, \mathbf{t}$.
  - Angle between two rays: $\cos\theta = \mathbf{x_1^T}\omega\mathbf{x_2}/(\cdots)$
  - Two rays are othogonal iff $\mathbf{x_1^T}\omega\mathbf{x_2} = \mathbf{0}$.
  - We can calibrate the camera given ω, using Cholesky factorization.
  - Circular points of all planes lie on Ω_∞ and are imaged onto intersection of plane's $\mathbf{l}_\infty$ and ω

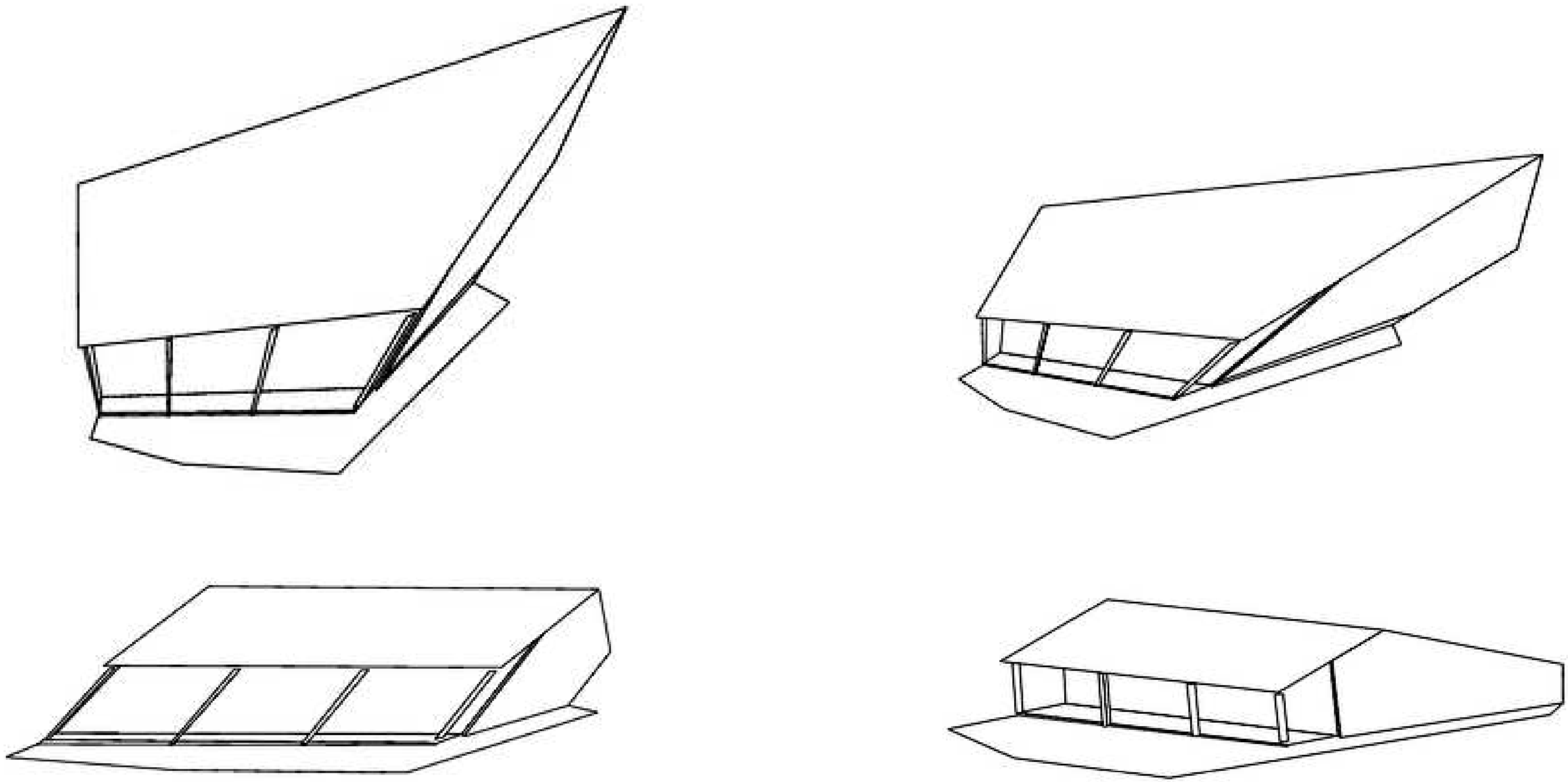
Projective to Affine Structure

- The ambiguity is affine if the plane at infinity or $\mathbf{H}_\infty$ is known.
- Plane at infinity can be recovered in the image using 3 parallel line pairs in different directions. Each gives a point on the plane at infinity.



- π_∞ can be obtained in many ways: from distance ratios, perpendicular lines, etc.
- If $\mathbf{H}_\infty$ is known, $\mathbf{P} = [\mathbf{I}|\mathbf{0}]$ and $\mathbf{P}' = [\mathbf{H}_\infty|\mathbf{e}']$ are affine cameras. Triangulation using these will give structure upto an affine ambiguity

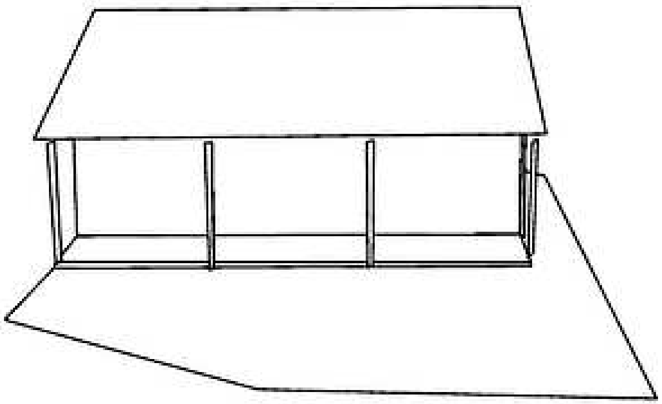
To Affine Structure of House

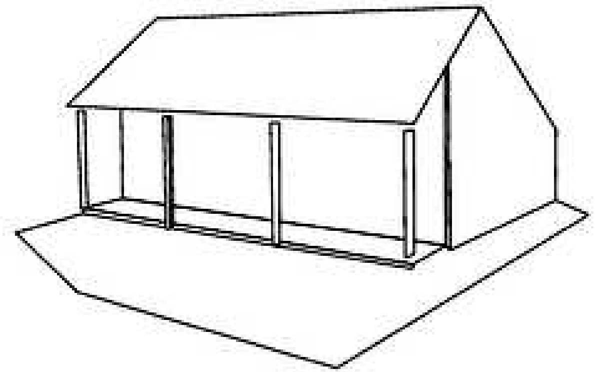


Affine to Metric Structure

- Metric ambiguity if image of the absolute conic ω is known.
- Map ω to $\mathbf{I}$ on π_∞ using $\mathbf{H}$. The same $\mathbf{H}$ will transform affine structure to metric structure.
- IAC can be recovered from parallel lines, perpendicular lines/planes in the image, or a combination of those.
- IAC needs to be known only in one view. It can be transferred to the other using $\mathbf{H}_\infty$
- From IAC, get $\mathbf{K}$ using $\omega^{\text{-}\mathbf{1}} = \mathbf{K}\mathbf{K}^{\mathbf{T}}$. Essential matrix $\mathbf{E} = \mathbf{K}^{\prime\mathbf{T}}\mathbf{F}\mathbf{K}$. Decomposed it into metric cameras with 4-way ambiguity. Resolve by identifying a point in front of the camera. Triangulation will now result in metric structure of the scene.

To Metric Structure of House



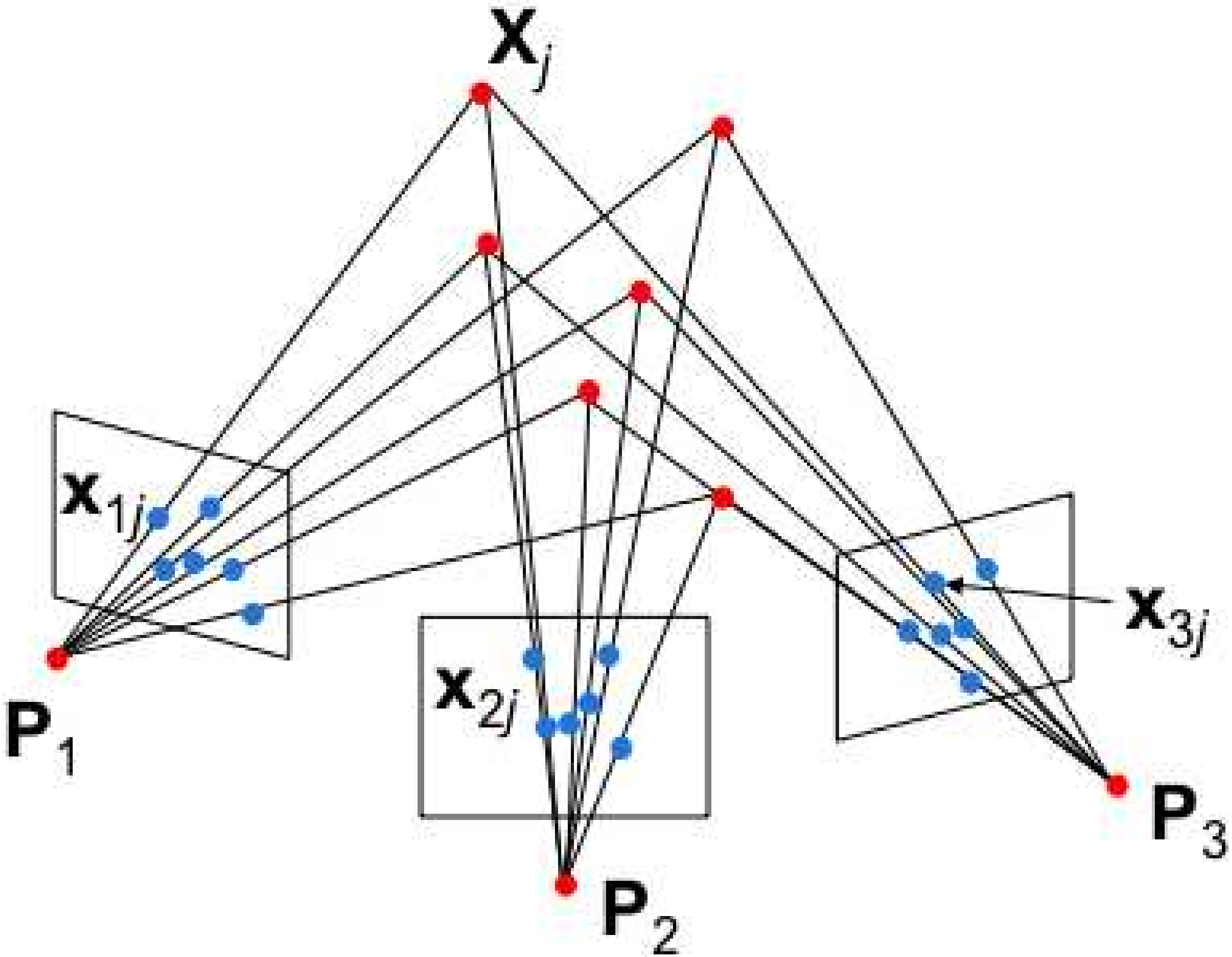


Direct Metric Structure

- Projective structure can be converted to metric using ground truth points whose metric structure is known.
- Assume recovered projective $\mathbf{X_i}$ has metric coordinates $X_{\mathbf{E}i}$. A homography exists such that: $X_{\mathbf{E}i} = \mathbf{H}\mathbf{X}_i$
- The above gives 3 equations. With 5 or more points, the homography $\mathbf{H}$ can be recovered.
- Apply the $\mathbf{H}$ to all recovered points (and camera) to get metric structure.

Structure from Motion Using Bundle Adjustment

Points in Multiple Views



Structure From Motion

- m cameras and n points, with correspondences
- Unknown: m matrices P_i and n coordinates, X_j
- We have: $x_{ij} = P_i X_j, \;\; 1 \leq i \leq m, \; 1 \leq j \leq n$
- $2mn$ equations in total (2 for each match)
- Can be solved if $2mn > 11m + 3n$
- However, under projective, $\mathbf{PX} = (\mathbf{PQ})(\mathbf{Q}^{-1}\mathbf{X})$.
  A projective ambiguity will remain
- Projective structure if $2mn > 11m + 3n - 15$
- Affine structure if $2mn > 11m + 3n - 12$
- Metric structure if $2mn > 11m + 3n - 7$
- Affine/Metric structure only by enforcing affine/metric constraints

Bundle Adjustment

Given m views of n 3D points, with unknown $\mathbf{P_i}$ and $\mathbf{X_j}$.
Ideally, $x_{ij} = P^i X_j$

Minimize the re-projection error over all camearas/views:

$$\min_{P_i, X_j} \sum_{i=1}^{m} \sum_{j=1}^{n} dist(x_{ij}, P_i X_j)^2$$

A non-linear optimization problem. Can be solved using the Levenberg-Marquardt procedure directly.

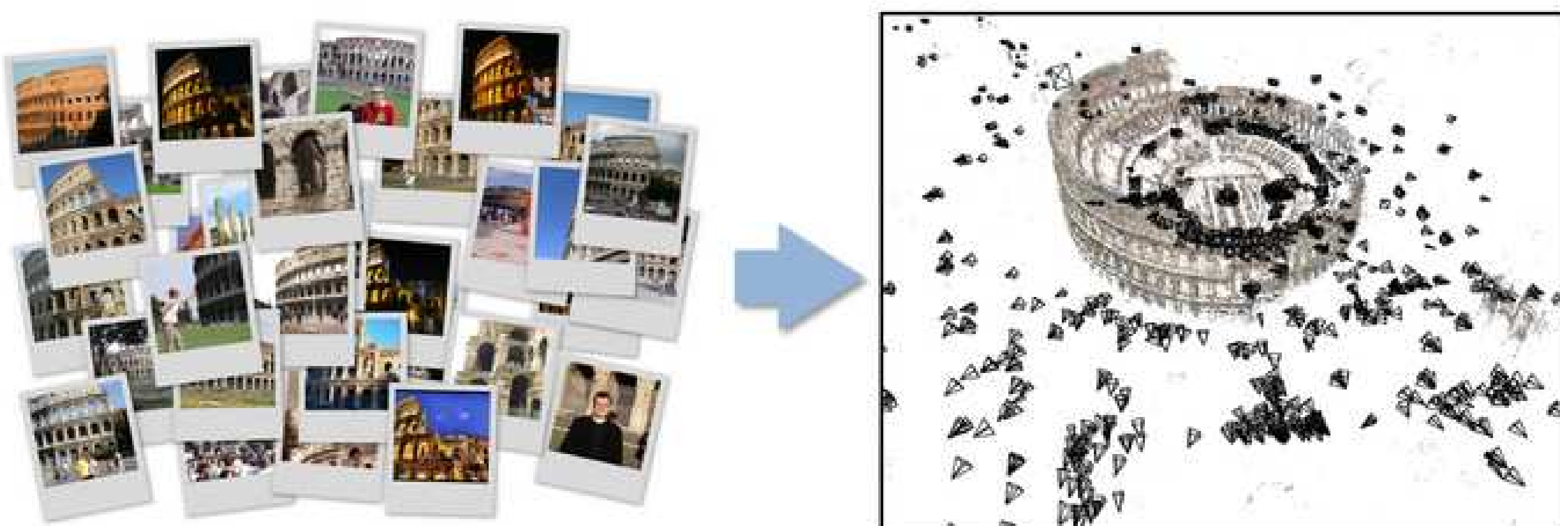
Called **bundle adjustment**. Known to photogrammetry community for a long time

Needs good initialization as the complex non-linear optimization problem can get stuck in local minima

SfM from Community Photo-Collections

PhotoTourism or PhotoSynth

- An automatic process, starting with independent images of a scene/monument/object. The images could be from a video sequence.
- Of particular interest has been SfM from Community Photo Collections (CPC), which are images that can be downloaded from flickr/picasa by giving a keyword like “Taj Mahal”.



SfM: Steps

1. Download images for the place of interest!!
2. Extract descriptors from interest points on all images
3. Match points in pairs of images using Approximate Nearest Neighbours
4. Refine matches using Geometric Verification: Epipolar relationship, etc.
5. Form tracks of points across images. Transitively connect matches to get long matching “tracks”
6. Build image connectivity graph based on common points
7. Perform incremental SfM using the image connectivity graph and bundle adjustment

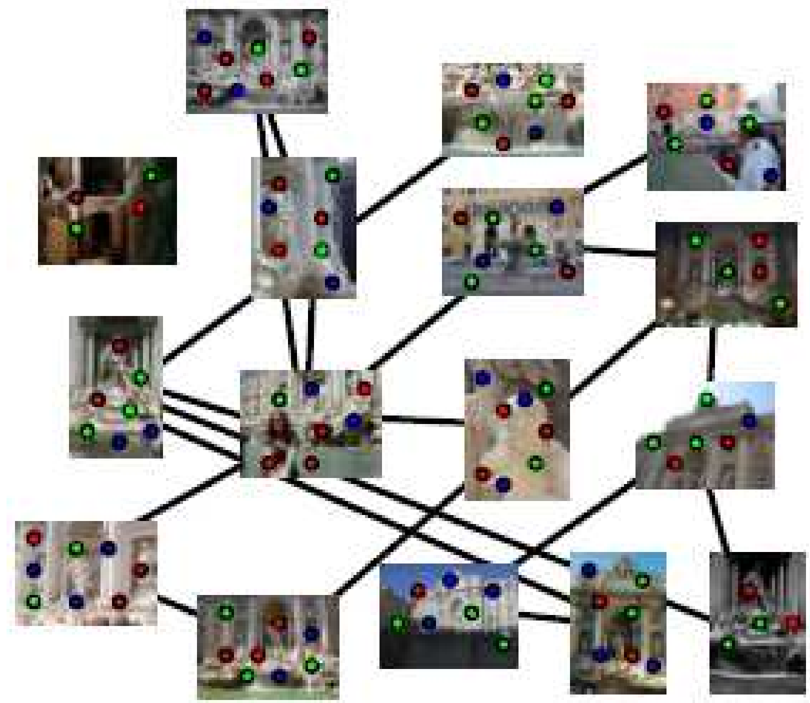
Matching Points Across Images

- Extract interest points $\mathbf{p}_{ij}$ in each image I_i and descriptors s_{ij} for it. SIFT is popular. A few thousand in a typical image
- Match interest points in image pairs. An approximate nearest neighbour approach is used, with a ratio test
- Point $\mathbf{p}_{ij}$ matches point $\mathbf{p}_{kl}$ iff:
  (a) dist(s_{ij}, s_{kl}) is minimum over all points in I_k and
  (b) dist$(s_{ij}, s_{kl}) < r \times$ dist(s_{ij}, s_{km}) where p_{km} is the second closest point in I_k. r is typically 0.6
- Discard all points involved in case of multiple matches

Geometric Verification and Tracks

- Find fundamental matrix between pairs of cameras using RANSAC
- Refine matches by eliminating those not satisfying epipolar relation
- Propagate matches using transitivity to generate **tracks** which represent the same world point in multiple images
- Form image connectivity graph. Two images have an edge if they share a point
- Densely connected regions represent a parts of the scene visible to a large number of views. Sparse, leaf regions denote low sampling of parts of the scene

Features and Match Graph



Form **tracks** of points by transitively connecting matches.
They represent the same 3D point in multiple images.

Track Statistics

For a typical large data set with approx 3000 images:

- 1.5 million tracks
- 75-80% with of length 2
- 98% of length less than 10
- A few tracks of length more than 100

Remember: Only 2D feature matching and verification has been done so far, but we seem to have come far!!

Structure from Motion

- $11m + 3n$ parameters for m cameras and n points. $2mn$ equations mapping each point in each camera
- Recover camera and structure. Minimize reprojection error across all of them using a non-linear minimization step. This needs good initialization
- Not possible to do them all together. So, start with one pair of cameras and incrementally add more cameras
- Adjust points and cameras to reduce global reprojection error after new cameras are added

Modern digital cameras store a lot of metadata in the images as EXIF tags, including the focal length!

Assume: Only focal length is the unknown intrinsic parameter!

Incremental SfM

- Find a strong starting pair of cameras. These should have a large number of points in common and a large baseline
- Find a pair with a large number of matches. Compute a planar homography from the matches. The pair is good if the error from the homography is *high*!
- Select the pair with the lowest percentage of inliers to homography using RANSAC
- Estimate the essential matrix for the camera pair
- Reconstruct cameras and common points using the essential matrix
- Perform **bundle adjustment** to minimize reprojection error

Adding Views

- While there are more connected cameras
  - Pick an image that sees most number of 3D points so far
  - Estimate pose of the camera using DLT and known 3D points. Perform a local bundle adjustment to correct new camera pose
  - Triangulate new points (if any) and add to the collection
  - Perform a global bundle adjustment on all cameras and points, using a non-linear optimization step
- Can remove outlier tracks altogether
- Can add a small group of camera views together, instead of one at a time

Bundle Adjustment

- Find $\mathbf{P}, \mathbf{X}$ that minimizes (with visibility indicator w_{ij})

$$g(\mathbf{P}, \mathbf{X}) = \sum_{i}^{m} \sum_{j}^{n} w_{ij} \, ||\mathbf{p}_{ij} - \mathbf{P}_i \mathbf{X}_j||^2$$

- Write it as $g(\mathbf{P}, \mathbf{X}) = ||\mathbf{A} - \mathcal{P}(\mathbf{P}, \mathbf{X})||^{\mathbf{2}}$ where $\mathcal{P}$ is the non-linear camera projection function
- Linearize $\mathcal{P}(\mathbf{P}, \mathbf{X}) = \mathcal{P}(\mathbf{P_0}, \mathbf{X_0}) + \mathbf{J} \begin{bmatrix} \Delta\mathbf{P} \\ \Delta\mathbf{X} \end{bmatrix}$
- Iterative solution using Levenberg-Marquardt method
- A sparse problem as the indicator w_{ij} of point j being visible in camera/image i is sparse.

Practical Aspects

- Heavy computations. Several days to reconstruct 500 images. About half of that time is for the bundle adjustment step
- Several optimizations have been worked on recently. Typical papers:
  “Building Rome in a Day”, ICCV 2009 (U of W)
  “Building Rome on a Cloudless Day”, ECCV 2010 (UNC)
- Combinatorics of pairwise matching is also huge. Use image search approaches to reduce the potential numbers

Building Rome in a Day

Agarwal, Simon, Seitz, Szeliski. ICCV 2009

- Over a million images of the city of Rome
- Pair-wise matching can take 15 years at 2 pairs/sec
- Find 40 most similar words using ...
- Query expansion to increase graph density
- Full bundle adjustment may run till end of time (nearly!)
- Use skeletal graphs to capture overall structure; perform bundle adjustment in local clusters
- Reconstructed Rome in **24 hours** on a 1000-node cluster!!

A recent local experiment for a 400-image Hampi dataset:

Extracting SIFT: **54 minutes**, Image matching: **17.2 hrs**
Bundle adjustment: **12.6 hrs**!!

Videos and Links

Videos from the Rome dataset: Colosseum and St. Peters

Check out **PhotoSynth** from Microsoft

For more on SfM: **http://grail.cs.washington.edu** and **http://www.cs.cornell.edu/projects/bigsfm/**

CVIT's page relating to India Digital Heritage project: **http://idh.iiit.ac.in/** and **http://monarch.iiit.ac.in/**

3D Structure Recovery in Practice

Studios to Record Events

Virtualized Reality (CMU, 1995)

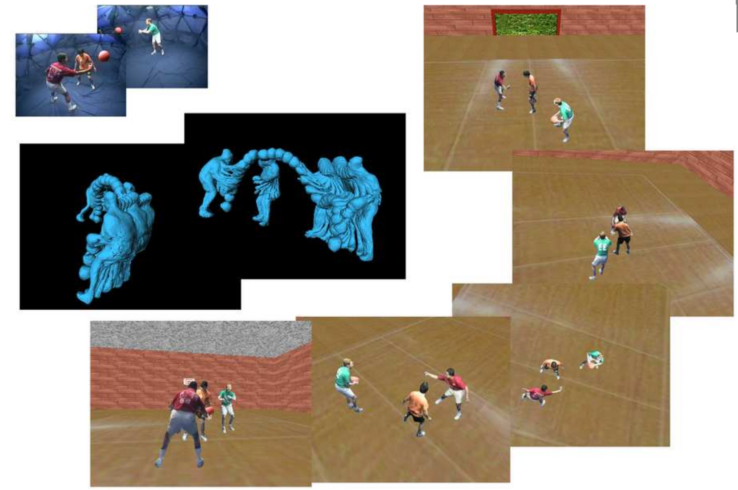
- A 51-camera recording setup
- Off-line digitization
- Multi-baseline stereo
- Merge depth maps to get structure

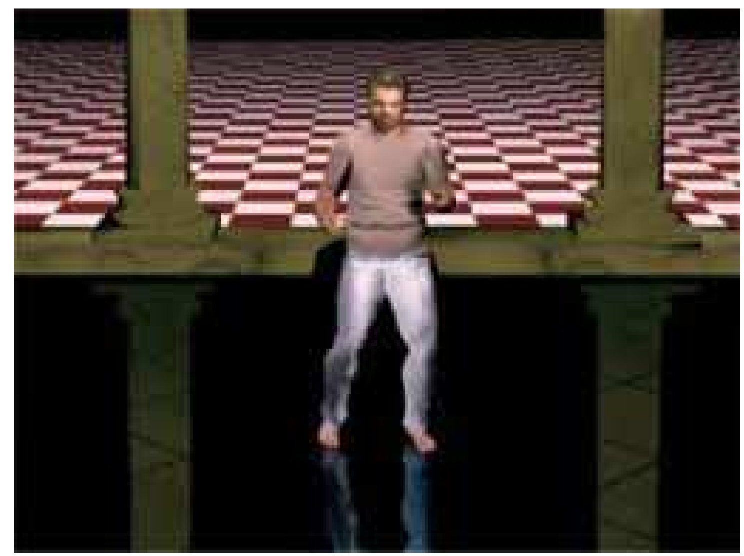
Free-Viewpoint Video (ETH, MPI)

- Multicamera setup, all digital
- Visual hull for quick structure

123D from Autodesk

- Submit your photographs
- Get a 3D model!!





Structure Recovery: Conclusions

- A problem that has been solved somewhat well
- Many challenges remain, but many have been tackled
- Next generation movies: Watch it from a viewpoint of your choice, decided at view time!!

Thank You!

Thanks to the Computer Vision community worldwide
for many of the figures!!