AN OVERVIEW OF 3D RECONSTRUCTION: FROM FUNDAMENTALS TO MODERN INSIGHTS

SHAIFALI PARASHAR

CNRS RESEARCH SCIENTIST

LYON, FRANCE

CONTACT: SHAIFALI.PARASHAR@LIRIS.CNRS.FR

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

cnrs

LIRIS

ÉCOLE CENTRALE LYON

3D RECONSTRUCTION OF RIGID OBJECTS

Images from [Snavely et al., 2016]

Structure-from-Motion [Longuet-Higgins, 1981]
Use multiple, registered, calibrated images to obtain a 3D structure from the projective relations between images

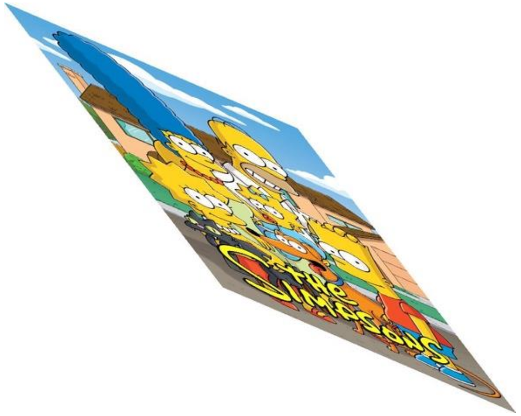
3D STRUCTURE AND IMAGES

Given a rigid object (P) under rigid motion (R,T) with new position Q = RP +T.
Various rigid motions may produce quite different image transformations

Image(P)

Image(Q1)

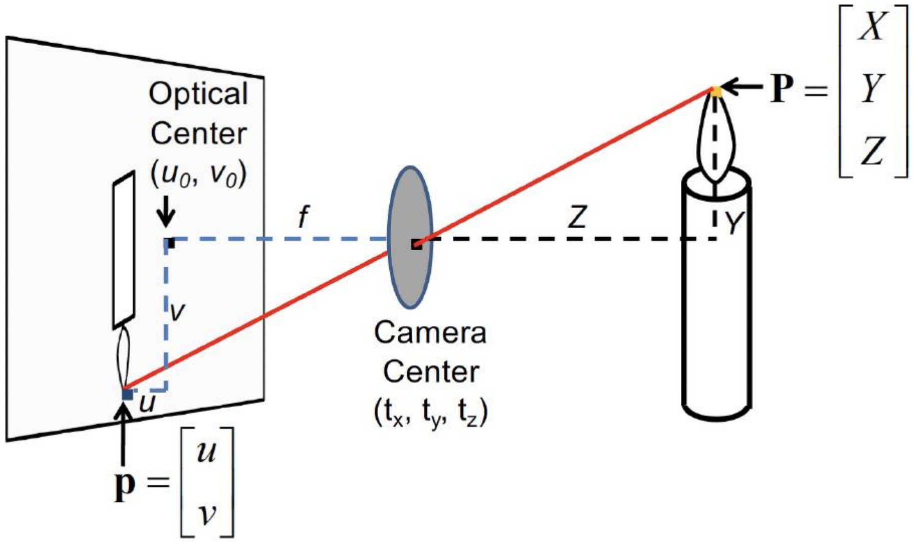
Image(Q2)



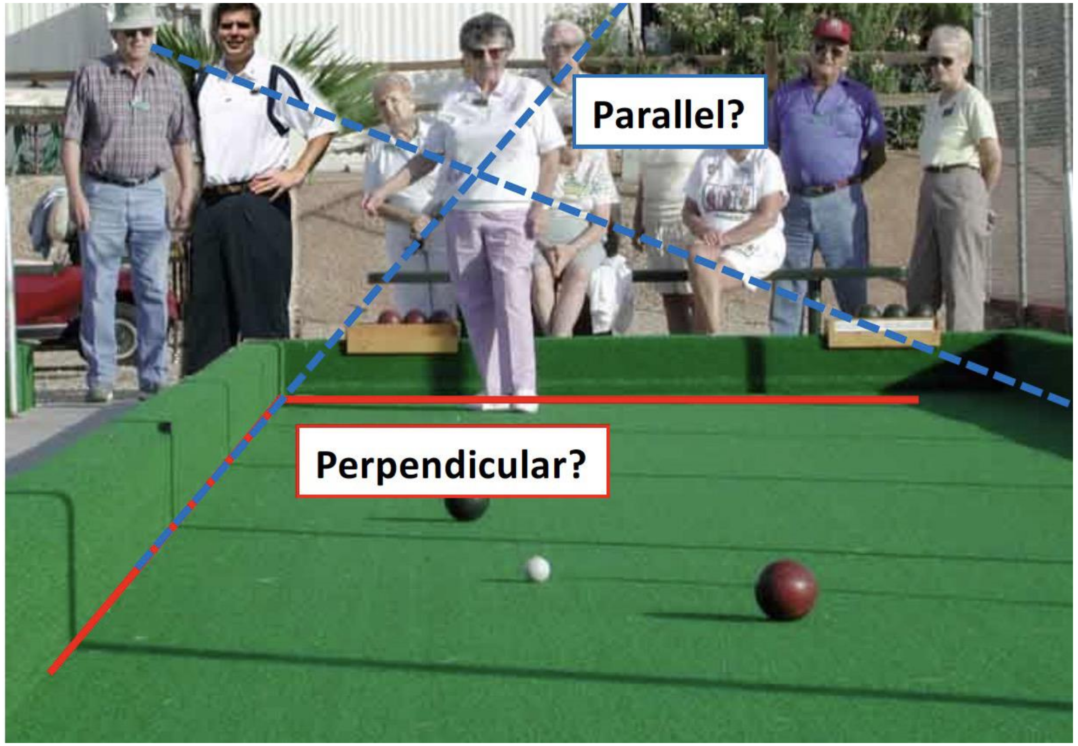
Image(Q3)

Image(Q4)

IMAGE FORMATION IS INVERSELY PROPORTIONAL TO DEPTH



$$\mathbf{p} = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -f\dfrac{X}{Z} \\ -f\dfrac{Y}{Z} \end{bmatrix}$$



*Shift $\mathbf{p}$ with optical center to be precise

3D STRUCTURE AND IMAGES

Image(P)

Image(Q1)

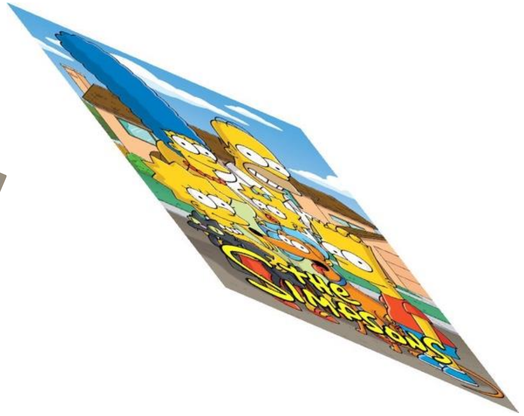
lengths and angles are preserved

Euclidean

Image(Q2)

angles are preserved

Similarity



Image(Q3)

parallelism is preserved

Affine

Image (Q4)

collinearity is preserved

Projective

3D STRUCTURE AND IMAGES

Image(P) | Image(Q1) | Image(Q2) | Image(Q3)

lengths and angles are preserved

angles are preserved

parallelism is preserved

Euclidean: $\mathbf{A}=\mathbf{R}$

Similarity: $\mathbf{A}=s\mathbf{R}$

Affine: $\mathbf{A}$

Transforming $\mathbf{x}$ at Image(P) yields $\overline{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{t}$

$$\begin{pmatrix} \overline{\mathbf{x}} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{A} & t \\ 0_{1\times 2}^\top & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}$$

$\mathbf{R}$: rotation
s: scalar
t: translation

3D STRUCTURE AND IMAGES

Image(P)	Image(Q1)	Image(Q2)	Image(Q3)	Image (Q4)
	lengths and angles are preserved	angles are preserved	parallelism is preserved	collinearity is preserved
	Euclidean: $\mathbf{A}=\mathbf{R}$	Similarity: $\mathbf{A}=s\mathbf{R}$	Affine: $\mathbf{A}$	Projective=??

Transforming $\mathbf{x}$ at Image(P) yields $\overline{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{t}$

$$\begin{pmatrix} \overline{\mathbf{x}} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{A} & t \\ 0_{1\times 2}^{\top} & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}$$

$\mathbf{R}$: rotation
s: scalar
t: translation

3D STRUCTURE AND IMAGES

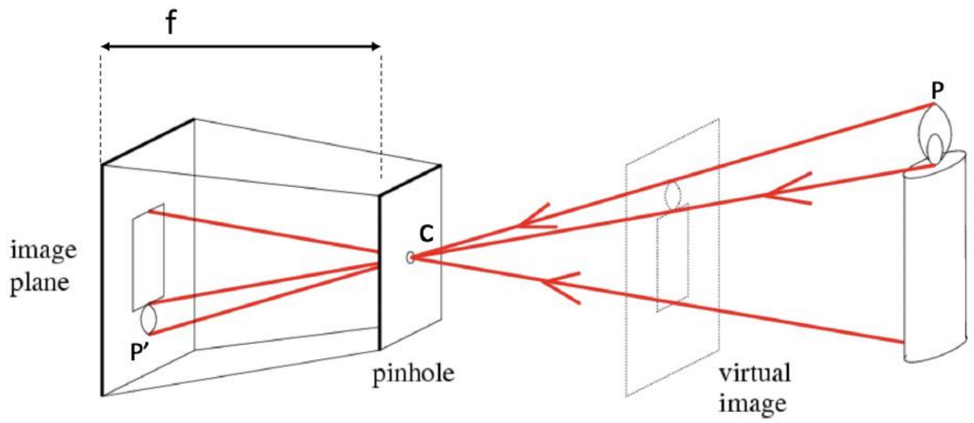
Image(P)	Image(Q1)	Image(Q2)	Image(Q3)	Image (Q4)
	lengths and angles are preserved	angles are preserved	parallelism is preserved	collinearity is preserved
	Euclidean: $\mathbf{A}=\mathbf{R}$	Similarity: $\mathbf{A}=s\mathbf{R}$	Affine: $\mathbf{A}$	Projective=??

$$\begin{pmatrix} \overline{\mathbf{x}} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{A} & t \\ \mathbf{v}^{\top}_{1\times 2} & v \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}$$

Use homogeneous coordinates.

3D STRUCTURE AND IMAGES

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad \begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$



Homogeneous image coordinates can jointly represent all 3D points along a camera ray

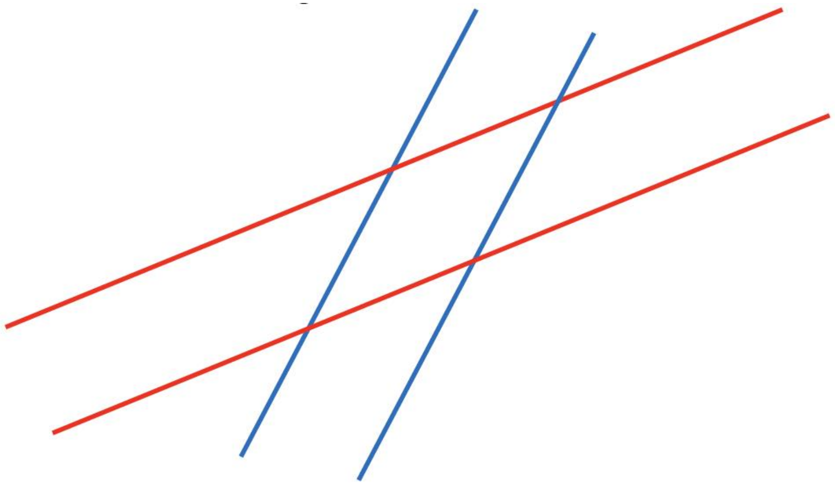
$$\begin{pmatrix} \overline{\mathbf{x}} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{A} & t \\ \mathbf{v}^{\top}_{1\times 2} & v \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}$$

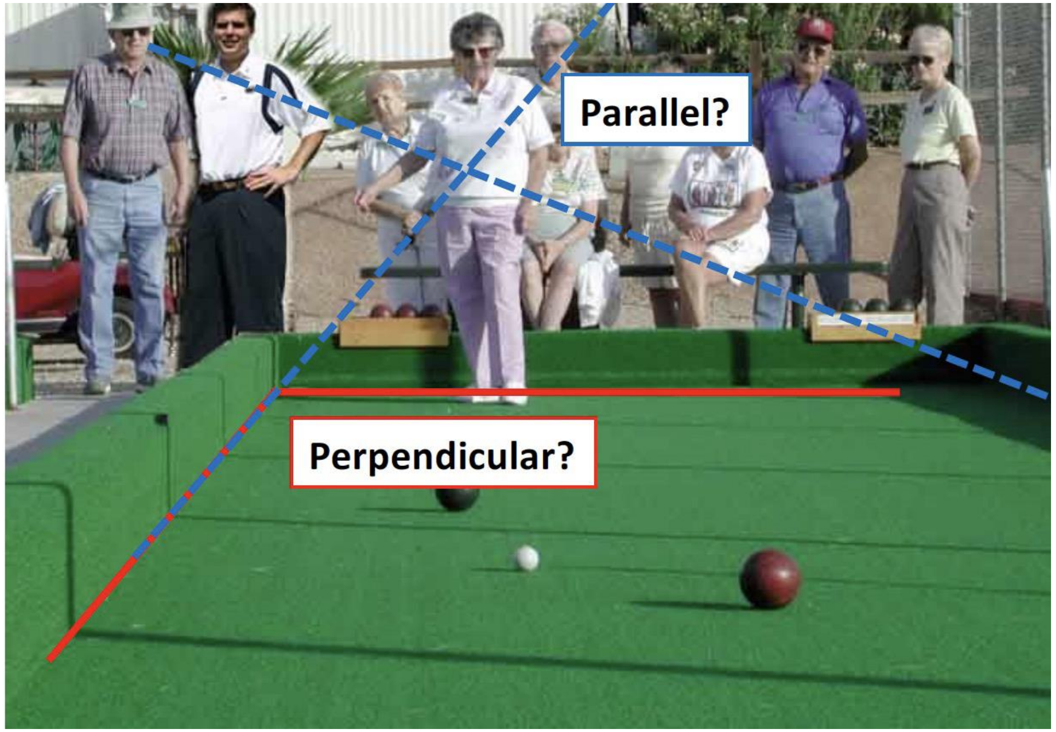
A division with $\mathbf{v}^{\top}_{1\times 2}\,\mathbf{x}+v$ on the right hand side is implicit

3D STRUCTURE AND IMAGES

Intersection of parallel lines

Cartesian: (Inf,Inf)
Homogeneous: (x,y,0)





3D STRUCTURE AND IMAGES

Image(P)	Image(Q1)	Image(Q2)	Image(Q3)	Image (Q4)
	lengths and angles are preserved	angles are preserved	parallelism is preserved	collinearity is preserved
	Euclidean: $\mathbf{A}=\mathbf{R}$	Similarity: $\mathbf{A}=s\mathbf{R}$	Affine: $\mathbf{A}$	Projective (Homography)

$$\begin{pmatrix} \bar{\mathbf{x}} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{A} & t \\ \mathbf{v}_{1\times 2}^{\top} & v \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}$$

Projective (Homography) > Affine > Similarity > Euclidean

HOMOGRAPHY ESTIMATION: DIRECT LINEAR TRANSFORMATION

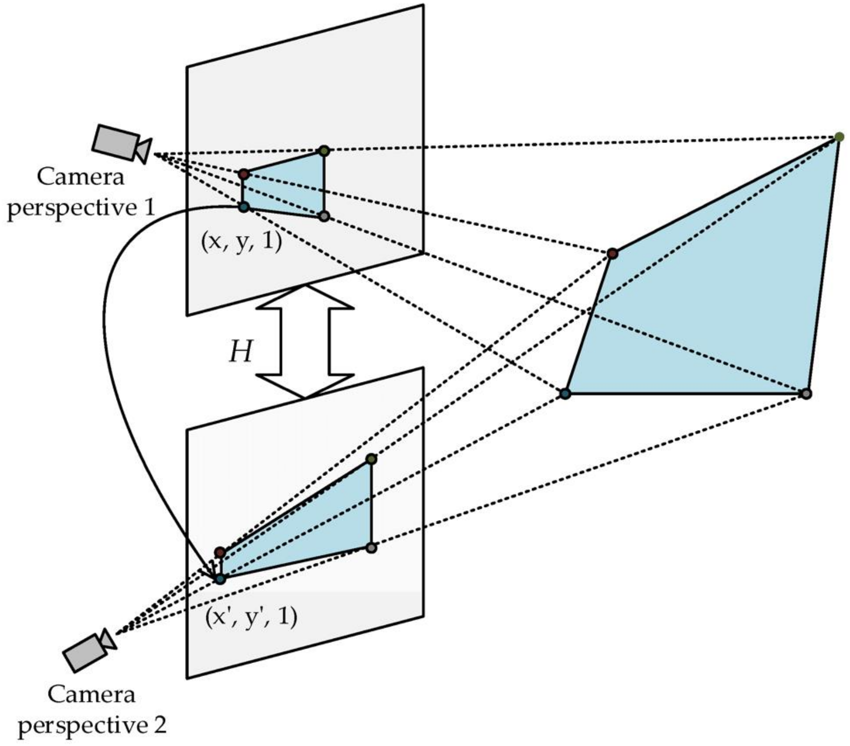
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$x' = \frac{x'}{1} = \frac{h_{11}x+h_{12}y+h_{13}}{h_{31}x+h_{32}y+h_{33}}$$

$$y' = \frac{y'}{1} = \frac{h_{21}x+h_{22}y+h_{23}}{h_{31}x+h_{32}y+h_{33}}$$

$$0 = (h_{11}x + h_{12}y + h_{13}) - (h_{31}x'x + h_{32}x'y + h_{33}x')$$

$$0 = (h_{21}x + h_{22}y + h_{23}) - (h_{31}y'x + h_{32}y'y + h_{33}y')$$



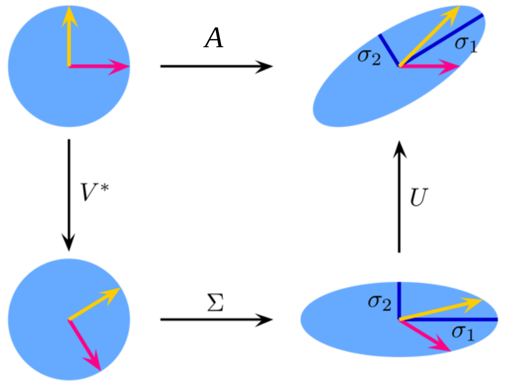
HOMOGRAPHY ESTIMATION: DIRECT LINEAR TRANSFORMATION

$$0 = Ah = \begin{bmatrix} x & y & 1 & 0 & 0 & 0 & -x'x & -x'y & -x' \\ 0 & 0 & 0 & x & y & 1 & -y'x & -y'^y & -y' \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix}$$

Solution: SVD of A.

SVD(A) = USV^T

S: singular values
V: eigenvectors
The smallest eigenvector is the solution.



$A = U \cdot \Sigma \cdot V^*$

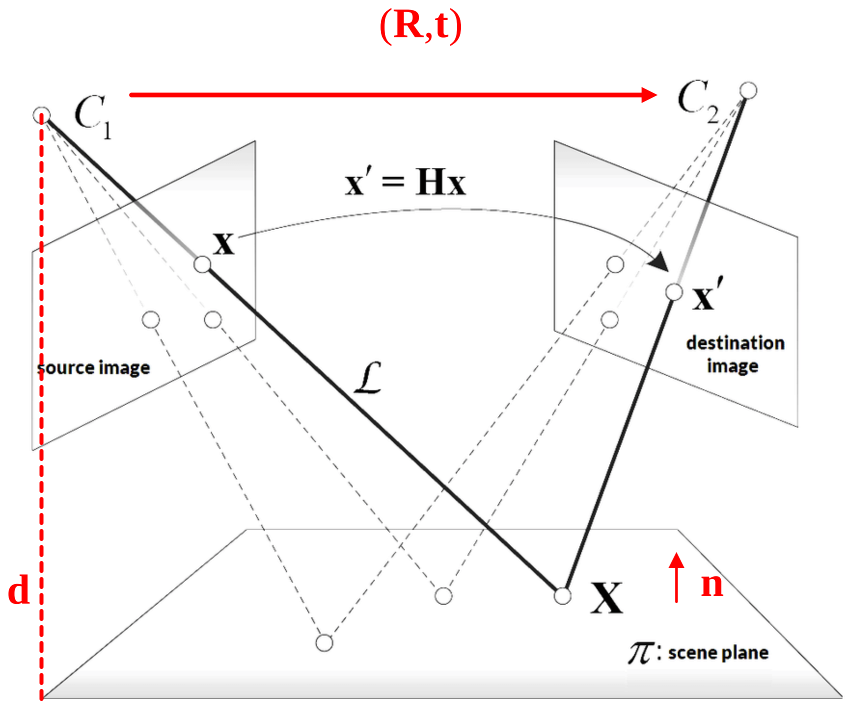
To solve, minimise $||Ah||$ such that $||h||=1$

3D PLANAR STRUCTURE AND HOMOGRAPHY

Assuming calibrated cameras

$\mathbf{H} = \mathbf{R} + \mathbf{t}\mathbf{n}^T/\mathbf{d}$

n: plane normal (unit)
d: distance of the plane from camera center



3D PLANAR STRUCTURE AND HOMOGRAPHY

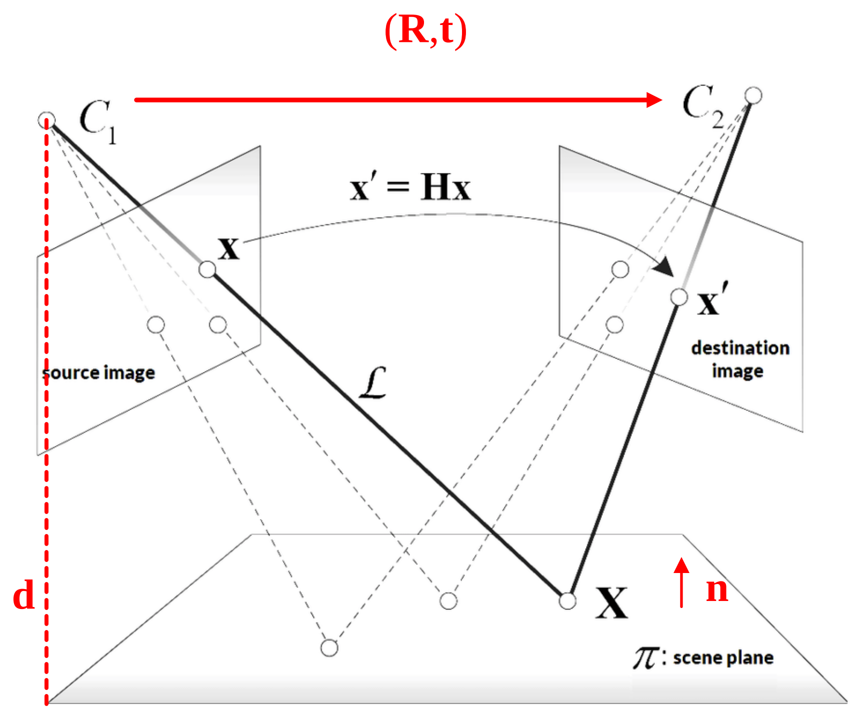
Assuming calibrated cameras

$\mathbf{H} = \mathbf{R} + \mathbf{t}\mathbf{n}^T/\mathbf{d}$

n: plane normal (unit)
d: distance of the plane from camera center

$s\mathbf{H}\mathbf{X} = \mathbf{R}\mathbf{X} + \mathbf{t}$
$\mathbf{n}^T\mathbf{X} - \mathbf{d} = 0$

$\mathbf{H} \sim= s\mathbf{H}$



3D PLANAR STRUCTURE AND HOMOGRAPHY

Homography Decomposition

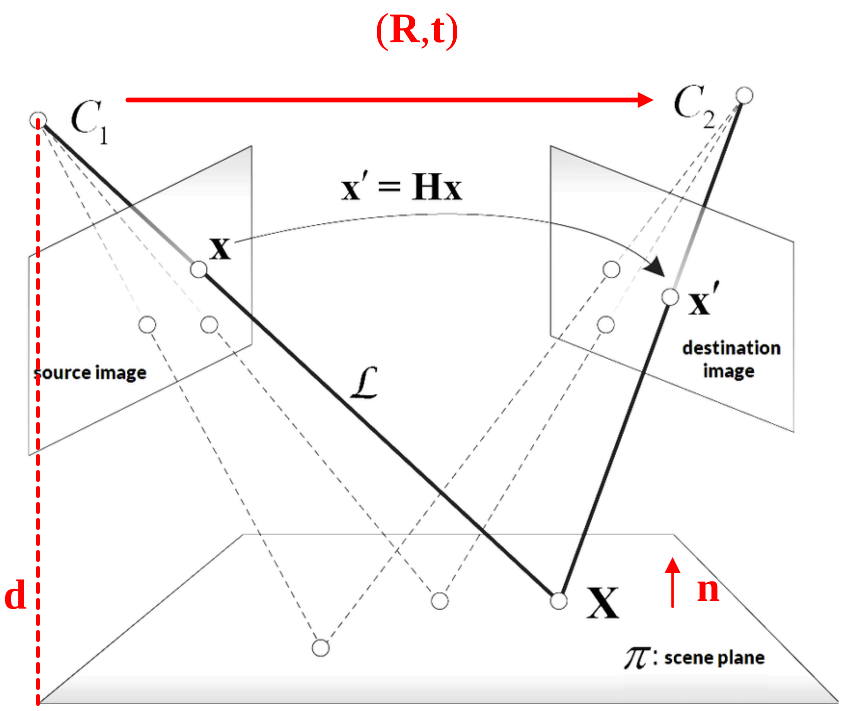
Given $\mathbf{H}$ computed with DLT, one can obtain closed-form solution to normal

$$\mathbf{S} = \mathbf{H}^\top \mathbf{H} - \mathbf{I} = \begin{bmatrix} s_{11} & s_{12} & s_{13} \\ s_{12} & s_{22} & s_{23} \\ s_{13} & s_{23} & s_{33} \end{bmatrix} \qquad M_{\mathbf{S}_{11}} = - \begin{vmatrix} s_{22} & s_{23} \\ s_{23} & s_{33} \end{vmatrix} = s_{23}^2 - s_{22}s_{33} \geq 0$$

$$\mathbf{n}'_a(s_{11}) = \begin{bmatrix} s_{11} \\ s_{12} + \sqrt{M_{\mathbf{S}_{33}}} \\ s_{13} + \epsilon_{23}\sqrt{M_{\mathbf{S}_{22}}} \end{bmatrix}; \quad \mathbf{n}'_b(s_{11}) = \begin{bmatrix} s_{11} \\ s_{12} - \sqrt{M_{\mathbf{S}_{33}}} \\ s_{13} - \epsilon_{23}\sqrt{M_{\mathbf{S}_{22}}} \end{bmatrix}$$

$$\mathbf{n}'_a(s_{22}) = \begin{bmatrix} s_{12} + \sqrt{M_{\mathbf{S}_{33}}} \\ s_{22} \\ s_{23} - \epsilon_{13}\sqrt{M_{\mathbf{S}_{11}}} \end{bmatrix}; \quad \mathbf{n}'_b(s_{22}) = \begin{bmatrix} s_{12} - \sqrt{M_{\mathbf{S}_{33}}} \\ s_{22} \\ s_{23} + \epsilon_{13}\sqrt{M_{\mathbf{S}_{11}}} \end{bmatrix}$$

$$\mathbf{n}'_a(s_{33}) = \begin{bmatrix} s_{13} + \epsilon_{12}\sqrt{M_{\mathbf{S}_{22}}} \\ s_{23} + \sqrt{M_{\mathbf{S}_{11}}} \\ s_{33} \end{bmatrix}; \quad \mathbf{n}'_b(s_{33}) = \begin{bmatrix} s_{13} - \epsilon_{12}\sqrt{M_{\mathbf{S}_{22}}} \\ s_{23} - \sqrt{M_{\mathbf{S}_{11}}} \\ s_{33} \end{bmatrix}$$

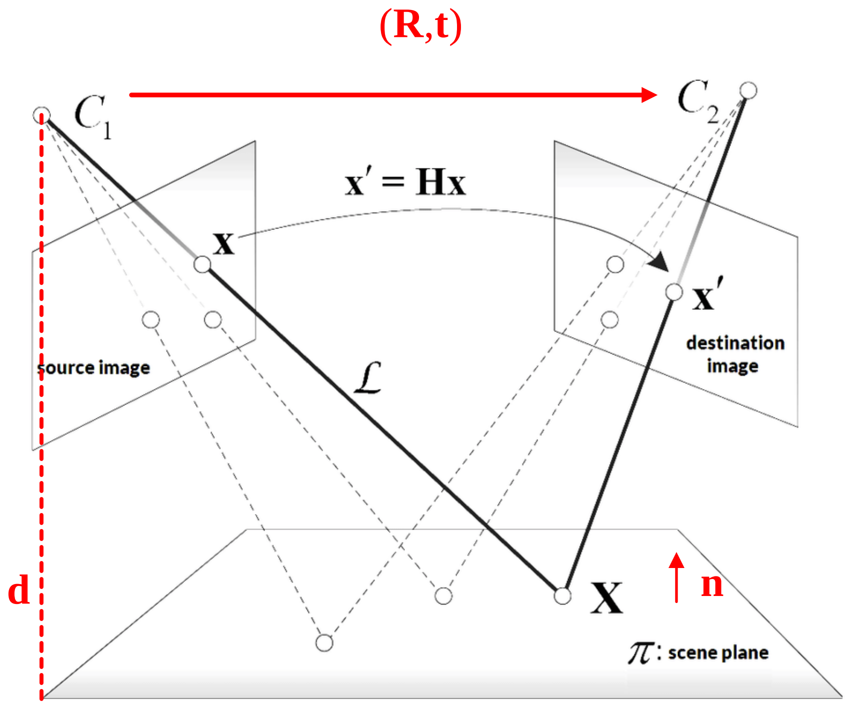


[Malis et Vargas, 2007].Deeper understanding of the homography decomposition for vision-based control .

3D PLANAR STRUCTURE AND HOMOGRAPHY

Given $\mathbf{H}$ computed with DLT, one can obtain closed-form solution to normal

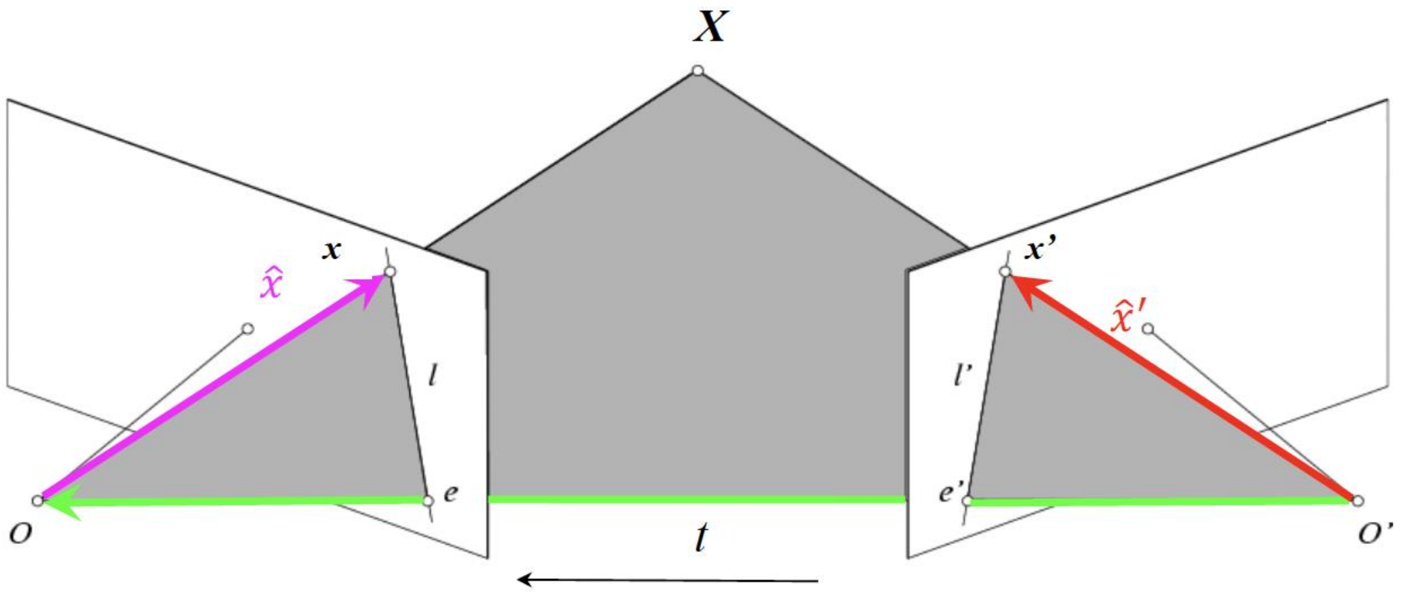
4 possible solutions: $\pm\mathbf{n}_a$, $\pm\mathbf{n}_b$ and their corresponding rotations and normals
2 can be discarded by sign
2 feasible solution remain; one can easily figure out the right one



[Malis et Vargas, 2007].Deeper understanding of the homography decomposition for vision-based control .

3D NON-PLANAR STRUCTURE AND HOMOGRAPHY

Epipolar Constraint

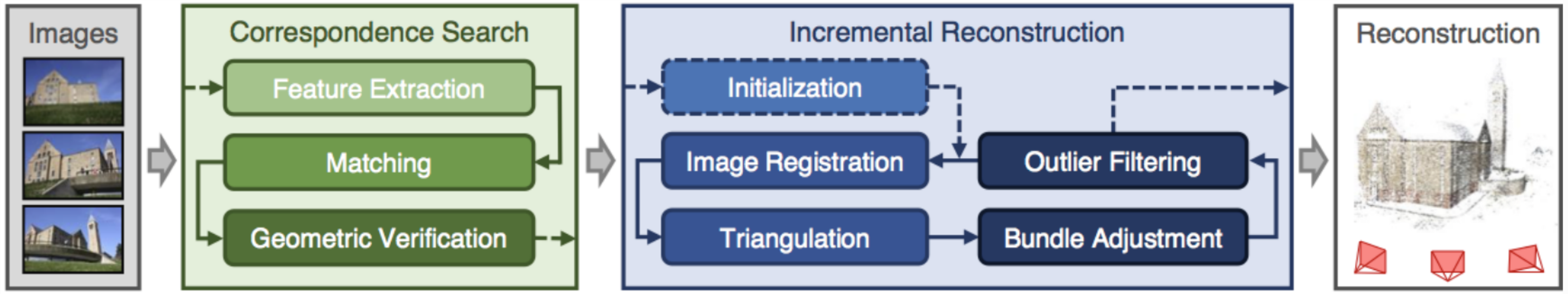


$$\hat{x} = K^{-1}x = X \qquad \hat{x}' = K'^{-1}x' = X'$$

$$\hat{x} = R\hat{x}' + t \Rightarrow \hat{x} \cdot [t \times (R\hat{x}')] = 0$$

3D RECONSTRUCTION: STRUCTURE FROM MOTION

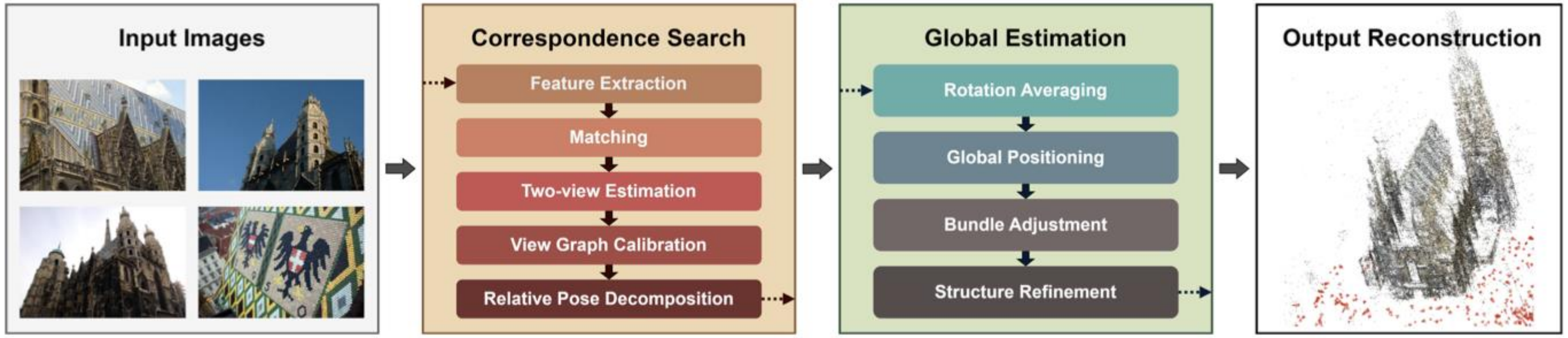
The COLMAP method [Schonberger and Frahm, CVPR 2016]



COLMAP's incremental Structure-from-Motion pipeline.

3D RECONSTRUCTION: STRUCTURE FROM MOTION

The GLOMAP method [Pan et al, ECCV 2024]



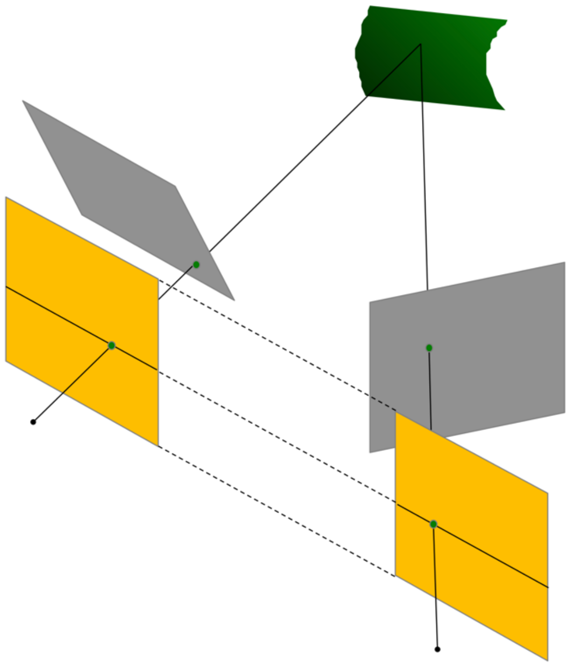
Joint camera and structure recovery
Considerably faster than COLMAP

3D RECONSTRUCTION: SPARSE TO DENSE

Use Multi-view stereo to get per-pixel depth using disparity

Convert 2-view setup to stereo setup ($R = I$, $T = [t,0,0]$)

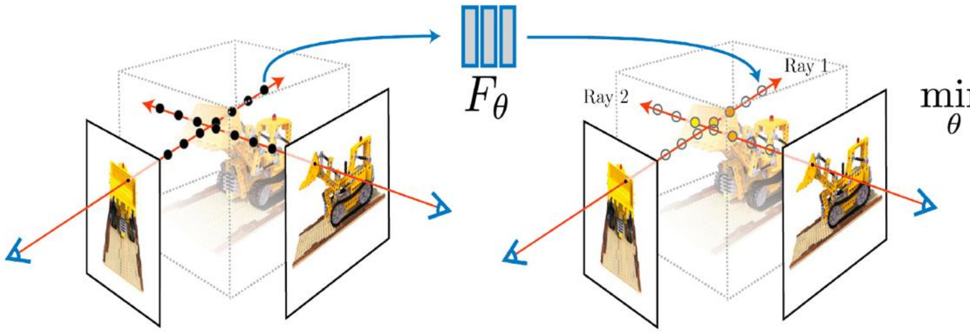
Limits correspondence search to x-direction



3D RECONSTRUCTION: SPARSE TO DENSE

NeRF [Mildenhall et al, ECCV 2020]

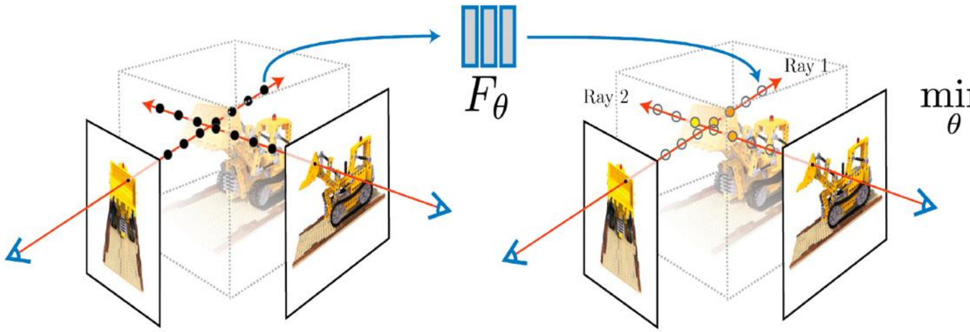
Scene-specific but high quality



$$\min_\theta \sum_i ||\,\mathrm{render}_i(F_\theta) - I_i||^2$$

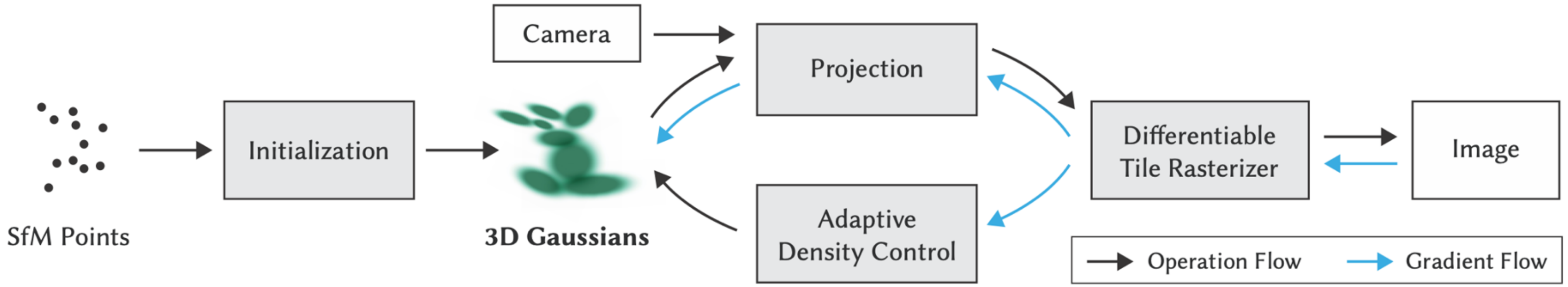
3D RECONSTRUCTION: SPARSE TO DENSE

NeRF [Mildenhall et al, ECCV 2020]



$$\min_{\theta} \sum_{i} ||\, \text{render}_i(F_\theta) - I_i ||^2$$

Scene-specific but high quality

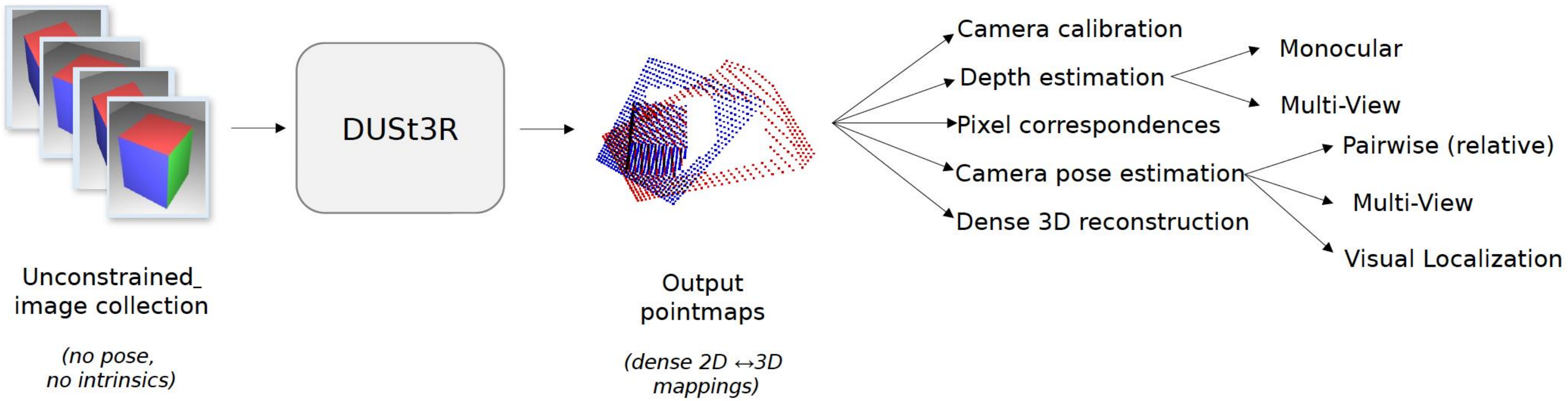


Gaussian Splatting [Kerbl et al, ToG, 2023]

Scene-specific
high(er) quality
computationally efficient

3D RECONSTRUCTION: FULLY SUPERVISED WITH SIMPLE REGRESSION LOSS

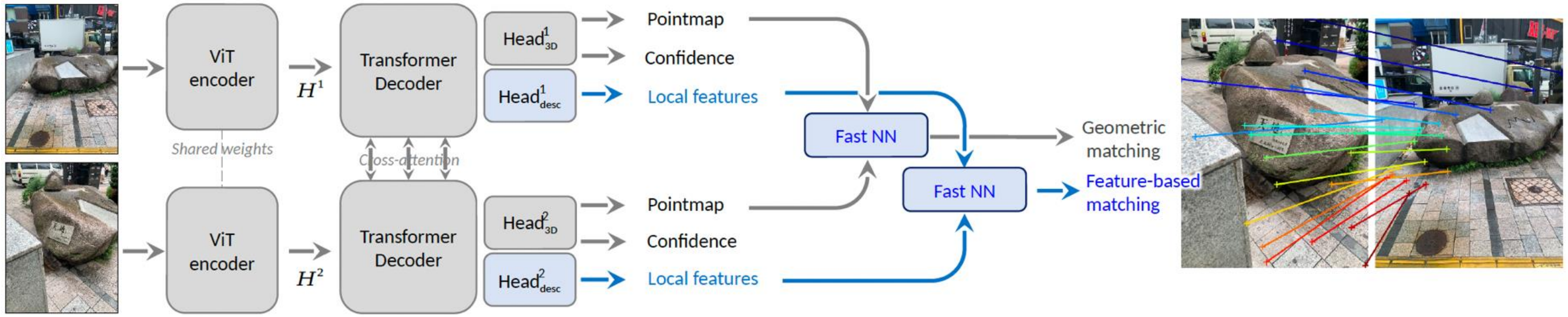
DUST3R [Wang et al, CVPR 2024]



Joint recovery of camera and scene
No vision-based losses are used
A simple data-driven methodology
Not at par with sota on SfM in terms of accuracy

3D RECONSTRUCTION: SELF SUPERVISED WITH GEOMETRIC LOSS

MAST3R [Leroy et al, ECCV 2024]



Built on Dust3R
Global alignment of scene
Loss minimizing the 2D reprojection error of 3D points in all cameras
At par with traditional SfM methods

3D RECONSTRUCTION: SUMMARY

Traditional (sparse)

COLMAP
- incremental SfM
- need calibrated cameras

GLOMAP
- global SfM
- no camera intrinsics needed

Traditional (Dense)

Multi View Stereo (traditional)
- stereo rectification
- generic

NERF/Gaussian Splatting (non-tradtional)
- scene specific
- computationally expensive

Non-traditional (Dense)

Dust3R
- fully supervised
- not much use of camera geometry

Mast3R
- self-supervised with global positioning
- geometry-aware, data-oriented learning

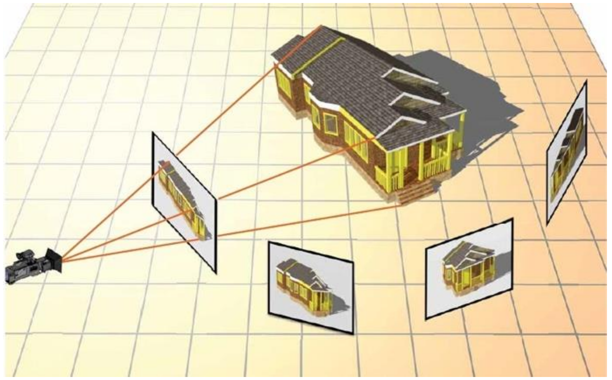
A reliably accurate recovery of the scene from both traditional and non-traditional methods

WHAT ABOUT DEFORMABLE OBJECTS?

Rigid Objects



Structure-from-Motion (SFM)



Can perfectly model the camera motion

Deformable (Non-Rigid) Objects

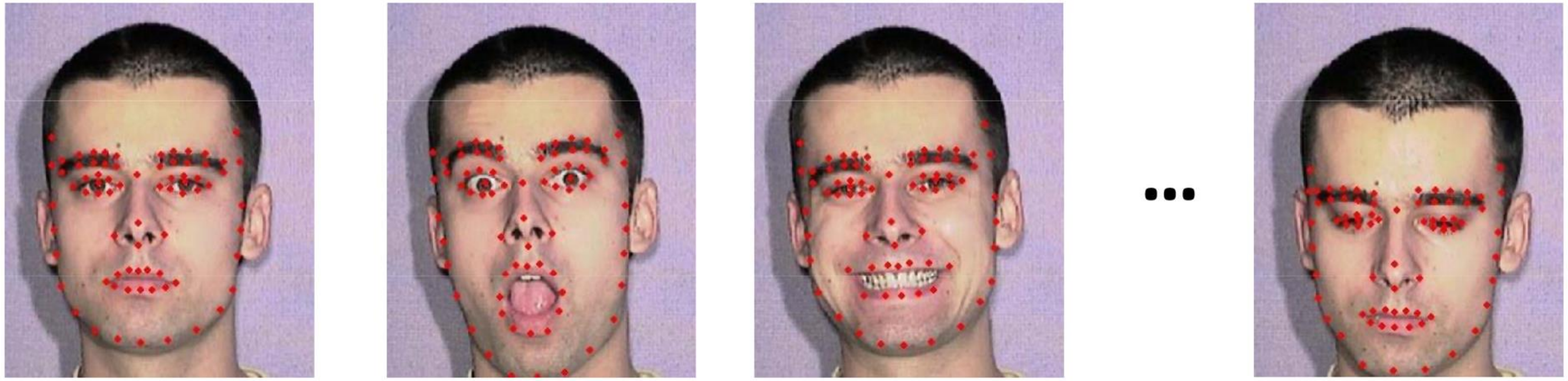


Non-Rigid Structure-from-Motion (NRSFM)

Confusion: Camera motion and/or object deformation

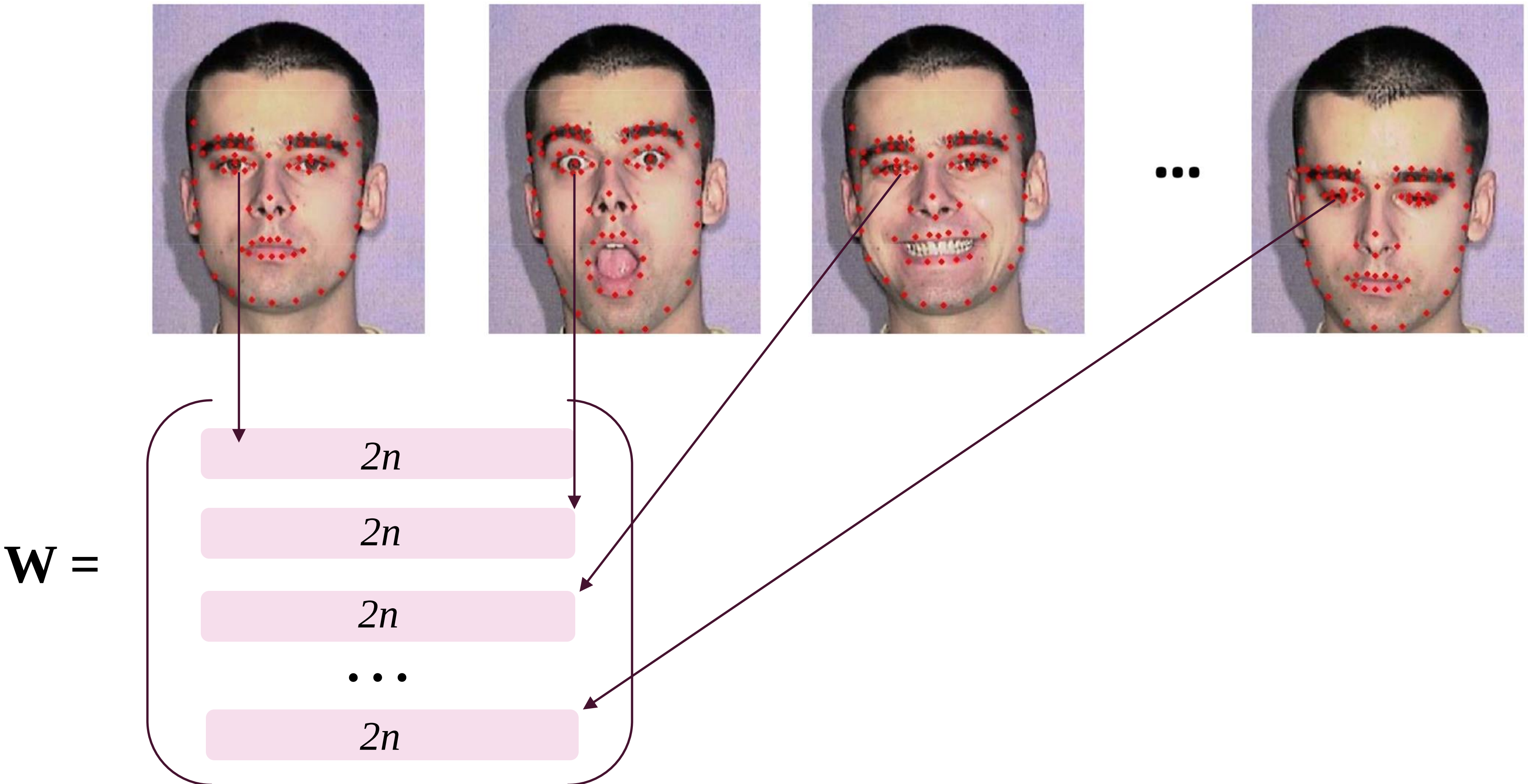
3D RECONSTRUCTION OF DEFORMABLE OBJECTS USING STATISTICS

[Bregler et al, 2000]



3D RECONSTRUCTION OF DEFORMABLE OBJECTS USING STATISTICS

[Bregler et al, 2000]



3D RECONSTRUCTION OF DEFORMABLE OBJECTS USING STATISTICS

[Bregler et al, 2000]

$$\mathbf{W} = \underbrace{\begin{pmatrix} 2n \\ 2n \\ 2n \\ \cdots \\ 2n \end{pmatrix}}_{2m \text{ X } n} = \underbrace{\begin{pmatrix} 2X3 & & \\ & \ddots & \\ & & 2X3 \end{pmatrix}}_{2m \text{ X } 3n} \underbrace{\begin{pmatrix} 3n \\ 3n \\ 3n \\ \cdots \\ 3n \end{pmatrix}}_{3m \text{ X } n} = \mathbf{R}\ \mathbf{S}$$

n points, m images
W: observation matrix
R: camera matrix
S: shape matrix

Assuming orthographic camera, solve for each image, $W=RS$ such that $RR^{\mathrm{T}} = \mathrm{I}$

3D RECONSTRUCTION OF DEFORMABLE OBJECTS USING STATISTICS

[Bregler et al, 2000]

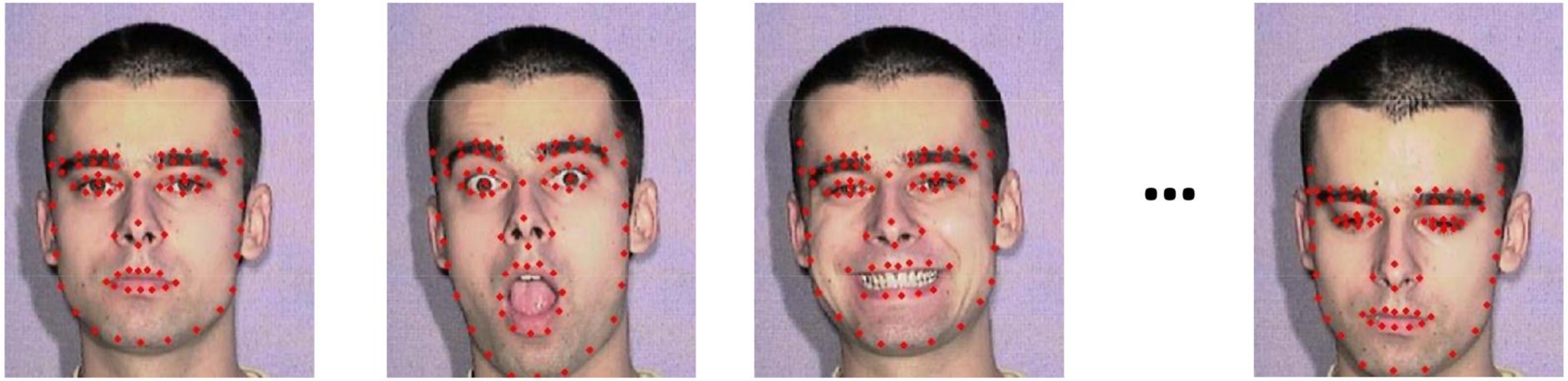
$$\mathbf{W} = \underbrace{\begin{pmatrix} 2n \\ 2n \\ 2n \\ \cdots \\ 2n \end{pmatrix}}_{2m\ X\ n} = \underbrace{\begin{pmatrix} 2X3 & & \\ & \ddots & \\ & & 2X3 \end{pmatrix}}_{2m\ X\ 3n} \underbrace{\begin{pmatrix} 3n \\ 3n \\ 3n \\ \cdots \\ 3n \end{pmatrix}}_{3m\ X\ n} = \mathbf{R}\ \mathbf{S}$$

n points, m images
W: observation matrix
R: camera matrix
S: shape matrix

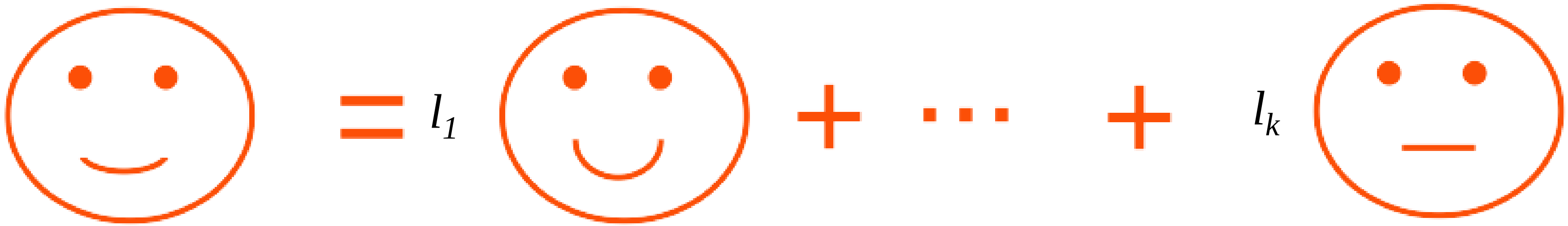
Severely ill-posed, given $2m\ X\ n$ observations we need to solve for $6m + 3m$ variables

3D RECONSTRUCTION OF DEFORMABLE OBJECTS USING STATISTICS

[Bregler et al, 2000]



Assumption: shapes lie in a low-dimensional space ($k << m$)



3D RECONSTRUCTION OF DEFORMABLE OBJECTS USING STATISTICS

[Bregler et al, 2000]

$$\mathbf{W} = \begin{pmatrix} W_1 \\ W_2 \\ W_3 \\ \cdots \\ W_m \end{pmatrix} = \begin{pmatrix} l_{11}R_1 & & l_{1k}R_1 \\ & \ddots & \\ l_{m1}R_m & & l_{mk}R_m \end{pmatrix} \begin{pmatrix} B_1 \\ B_2 \\ \cdots \\ B_k \end{pmatrix} = \mathbf{RS}$$

$2m \ X \ n$ $\qquad$ $2m \ X \ 3k$ $\qquad$ $3k \ X \ n$

3D RECONSTRUCTION OF DEFORMABLE OBJECTS USING STATISTICS

[Bregler et al, 2000]

$$\mathbf{W} = \begin{pmatrix} W_1 \\ W_2 \\ W_3 \\ \cdots \\ W_m \end{pmatrix} = \begin{pmatrix} l_{11}R_1 & & l_{1k}R_1 \\ & \ddots & \\ l_{m1}R_m & & l_{mk}R_m \end{pmatrix} \begin{pmatrix} B_1 \\ B_2 \\ \cdots \\ B_k \end{pmatrix} = \mathbf{RS}$$

$2m\ X\ n$ $\qquad$ $2m\ X\ 3k$ $\qquad$ $3k\ X\ n$

$$\Omega(R, L, B) = \sum_{i=1}^{f} \left\| W_i - R_i \sum_{i=1}^{k} l_{id} B_d \right\|^2 \quad \text{subject to} \quad R_i R_i^\top = I_2$$

Solve using truncated SVD

3D RECONSTRUCTION OF DEFORMABLE OBJECTS USING STATISTICS

[Bregler et al, 2000]

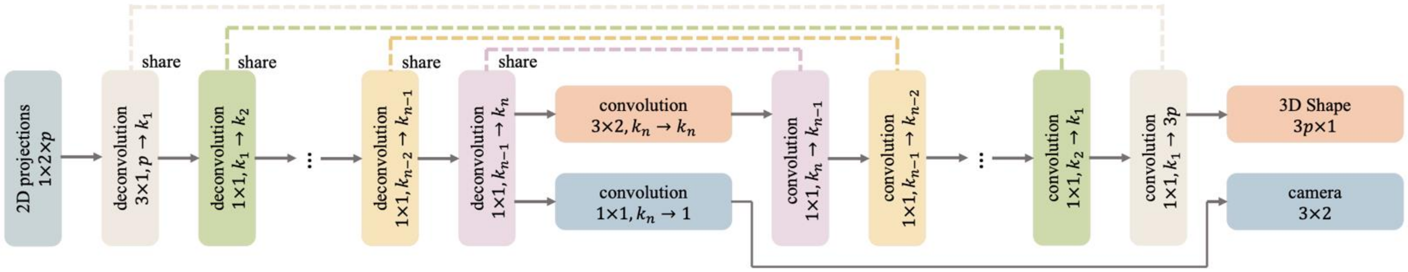
$$\mathbf{W} = \begin{pmatrix} W_1 \\ W_2 \\ W_3 \\ \cdots \\ W_m \end{pmatrix} = \begin{pmatrix} l_{11}R_1 & & l_{1k}R_1 \\ & \ddots & \\ l_{m1}R_m & & l_{mk}R_m \end{pmatrix} \begin{pmatrix} B_1 \\ B_2 \\ \cdots \\ B_k \end{pmatrix} = \mathbf{RS}$$

$2m\ X\ n$ $\qquad$ $2m\ X\ 3k$ $\qquad$ $3k\ X\ n$

Major problem: how to choose k ?

WHAT IF WE USED DEEP NETWORKS?

Deep NRSfM [Kong and Lucey, ICCV 2019]

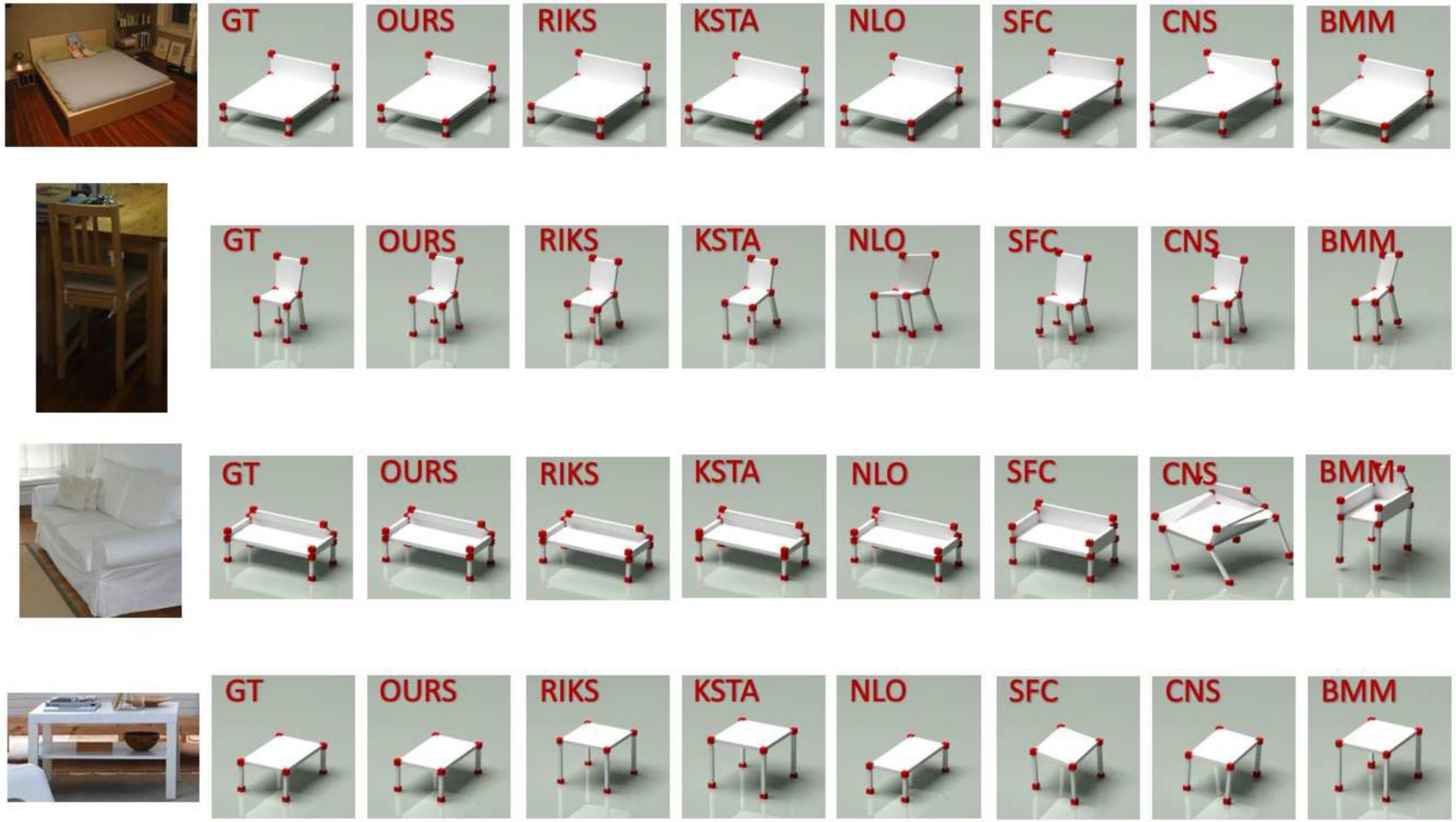


An encoder-decoder network with shared hierarchical dictionaries

$\mathbf{W} = \mathbf{RS}$ modelled with dictionaries $\mathbf{D}$

$$
\begin{aligned}
\mathbf{s} &= \mathbf{D}_1\boldsymbol{\psi}_1, &\quad \|\boldsymbol{\psi}_1\|_0 < \lambda_1, \boldsymbol{\psi}_1 \geq 0,\\
\boldsymbol{\psi}_1 &= \mathbf{D}_2\boldsymbol{\psi}_2, &\quad \|\boldsymbol{\psi}_2\|_0 < \lambda_2, \boldsymbol{\psi}_2 \geq 0,\\
\vdots \quad &, &\quad \vdots\\
\boldsymbol{\psi}_{n-1} &= \mathbf{D}_n\boldsymbol{\psi}_n, &\quad \|\boldsymbol{\psi}_n\|_0 < \lambda_n, \boldsymbol{\psi}_n \geq 0,
\end{aligned}
$$

WHAT IF WE USED DEEP NETWORKS?



3D RECONSTRUCTION OF DEFORMABLE OBJECTS USING STATISTICS

Traditional (statistical modeling)

- Low rank shape-basis [Bregler et al., 2000]

+ non-linear refinement [Del Bue et al., 2004]
+ trace minimisation and refinement [Dai et al., 2012]
+ Discrete cosine transformation [Gotardo et al., 2012]

- Low rank trajectory-basis [Akhter et al., 2009]

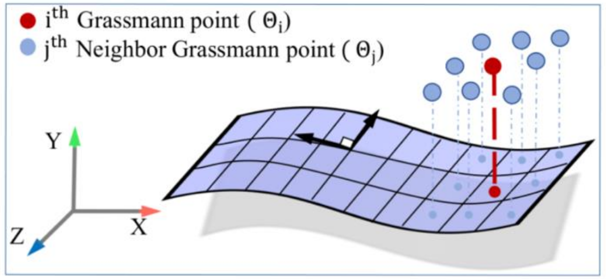
Neural network (statistical modeling)

- Hierarchical dictionary absed shape-basis [Kong and Lucey, ICCV 2019]
- Auto-encoder to align 3D shapes to common reference[Wang and Lucey, CVPR 2021]

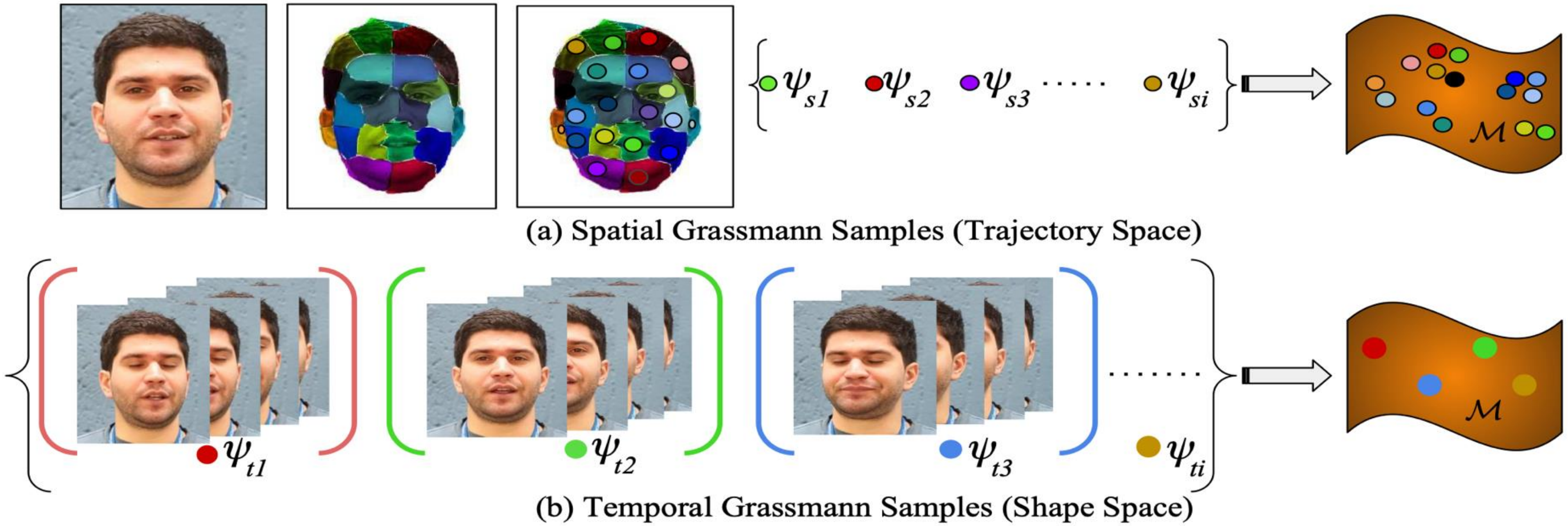
Only good for simple or sparse objects

STATISTICS-BASED DENSE RECONSTRUCTION

Dense NRSfM [Kumar et al, ICCV 2019]



Grassmannian Modeling

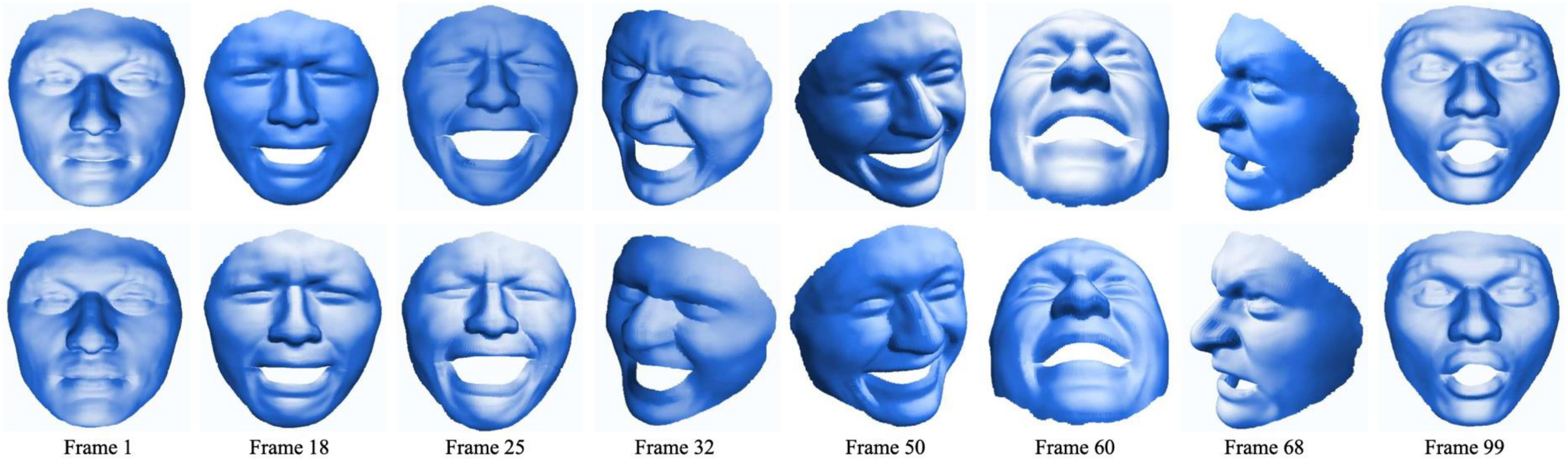


Consider local linear spaces for both shape and trajectory

Use grassmannian modeling to parametrise local linear subspaces to vector format

STATISTICS-BASED DENSE RECONSTRUCTION

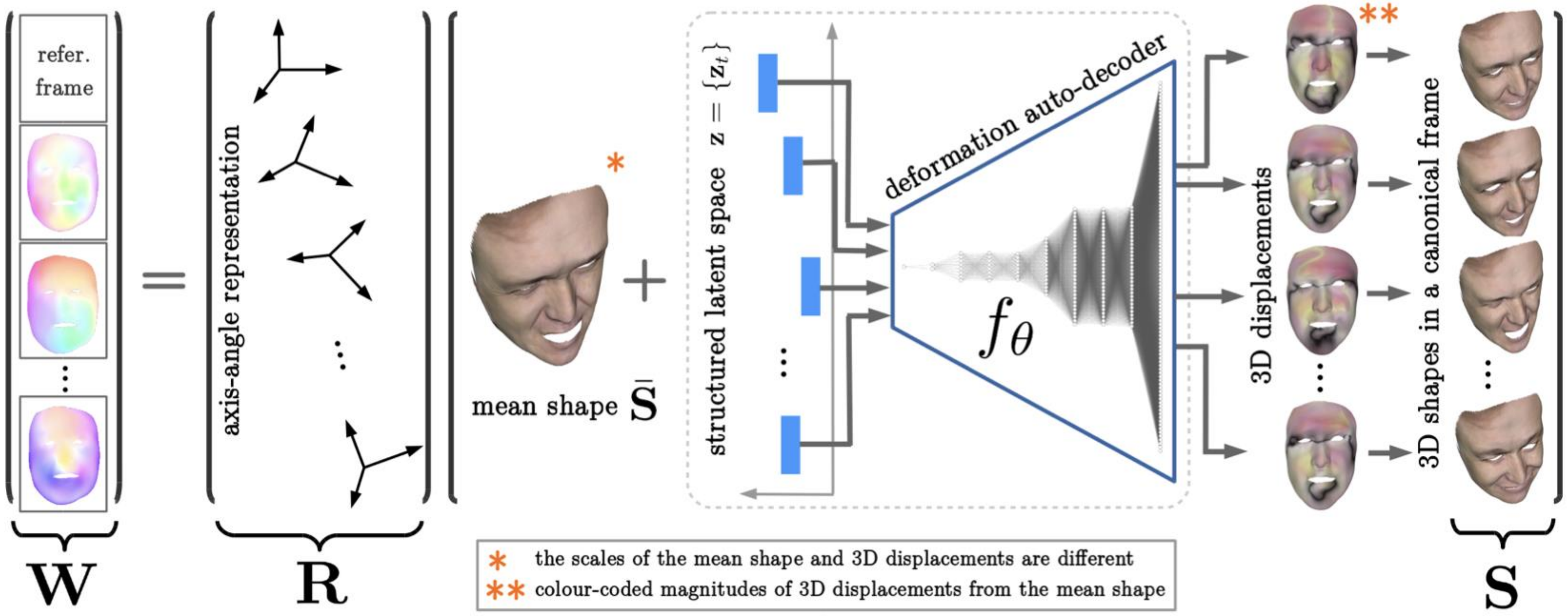
Dense NRSfM [Kumar et al, ICCV 2019]



Looks good!

STATISTICS-BASED DENSE RECONSTRUCTION

Neural NRSfM [Sidhu et al, ECCV 2020]



End-to-end learning with differentiable losses
latent space representation of deformation autoencoder

STATISTICS-BASED DENSE RECONSTRUCTION

Neural NRSfM [Sidhu et al, ECCV 2020]

Similar performance as other dense methods, visually appealing results

STATISTICS-BASED DEFORMABLE 3D RECONSTRUCTION

Sparse

Dense

Traditional (statistical modeling)

- Low rank shape-basis [Bregler et al., 2000]

+ non-linear refinement [Del Bue et al., 2004]
+ trace minimisation and refinement [Dai et al., 2012]
+ Discrete cosine transformation [Gotardo et al., 2012]

- Low rank trajectory-basis [Akhter et al., 2009]

- Grassmannain simplification of local linear modelling [Kumar et al, ICCV 2019]

Neural network (statistical modeling)

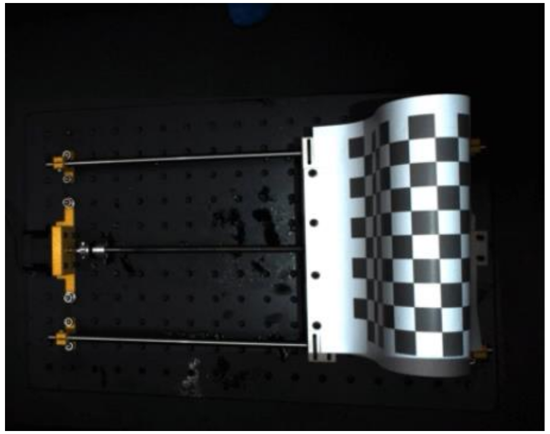
- Hierarchical dictionary absed shape-basis [Kong and Lucey, ICCV 2019]
- Auto-encoder to align 3D shapes to common reference[Wang and Lucey, CVPR 2021]

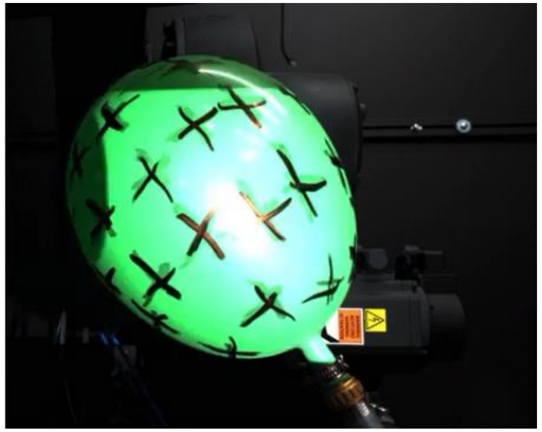
- End-to-end learning of deformations with latent space constraints [Sidhu et al, ECCV 2020]

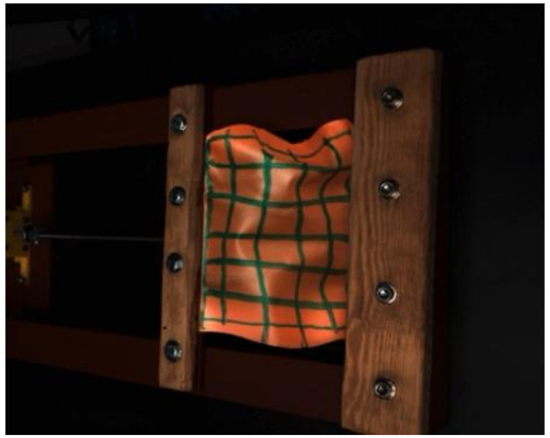
DIVERSE RANGE OF DEFORMABLE OBJECTS

Isometry (geodesic-preserving): e.g. paper or cloth [Most common]
Conformality (angle-preserving): e.g. balloon
Elasticity: e.g. rubber

Isometry = Conformality + Equiareality

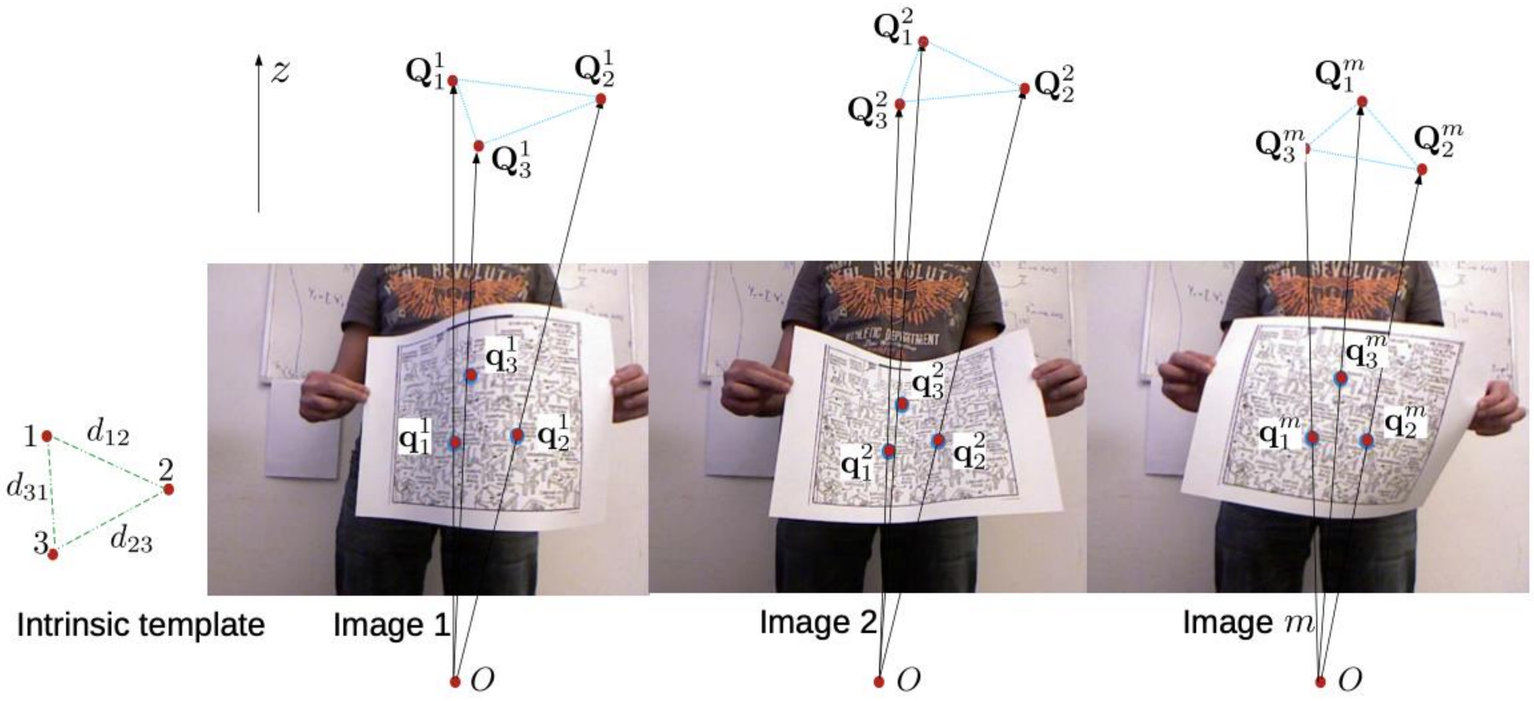






ISOMETRIC DEFORMATIONS

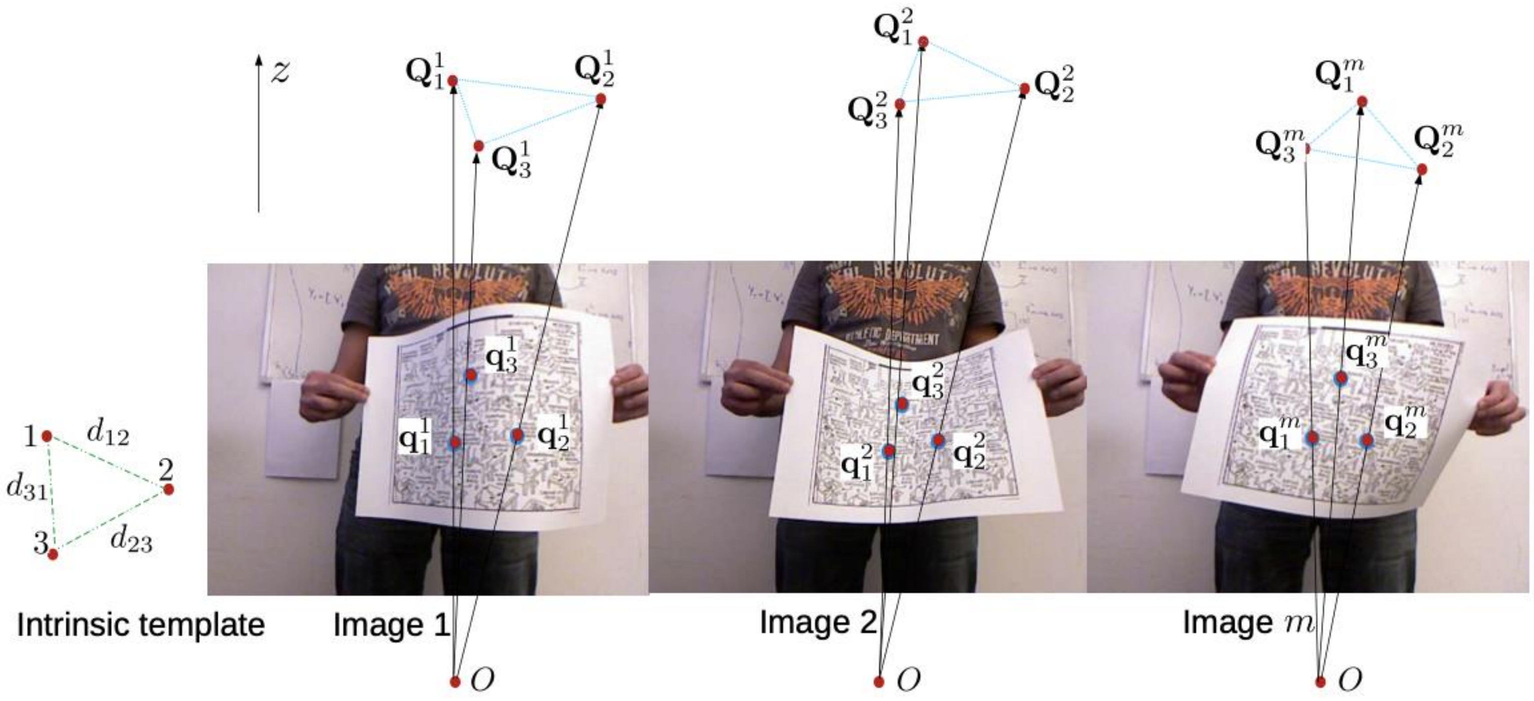
Inextensible NRSfM [Chhatkuli et al, CVPR 2016]



Maximise depth to ensure triangles to be congruent to intrinsice template
Better results than statistics-based methods

ISOMETRIC DEFORMATIONS

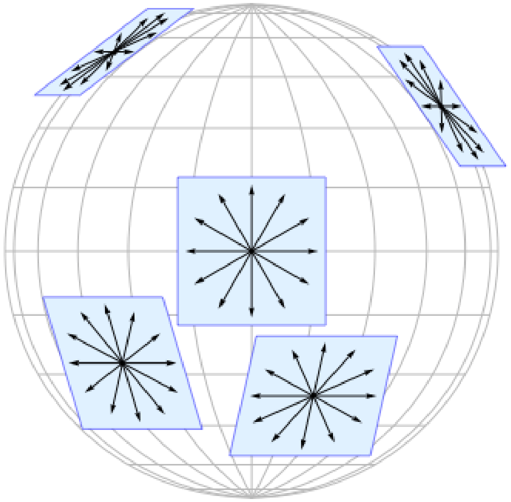
Inextensible NRSfM [Chhatkuli et al, CVPR 2016]



Problems: A computationally expensive approach

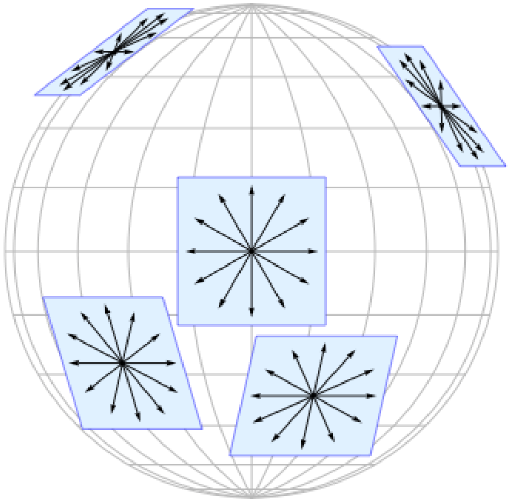
Euclidean approximations of the geodesics; marred with perspective projection

ISOMETRY: HOW TO PRESERVE GEODESICS?



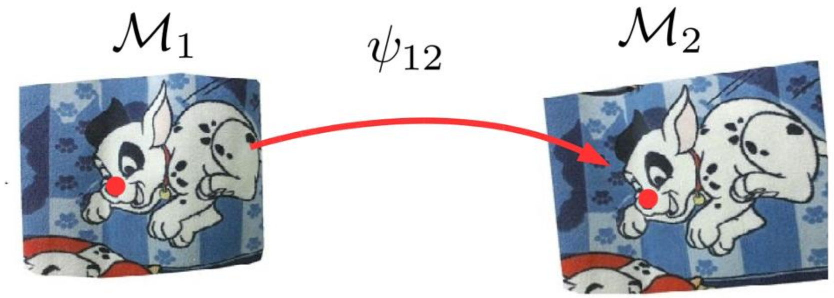
Surfaces are infinitesimally planar. Therefore, deformations are locally linear.

ISOMETRY: HOW TO PRESERVE GEODESICS?



Surfaces are infinitesimally planar. Therefore, deformations are locally linear.

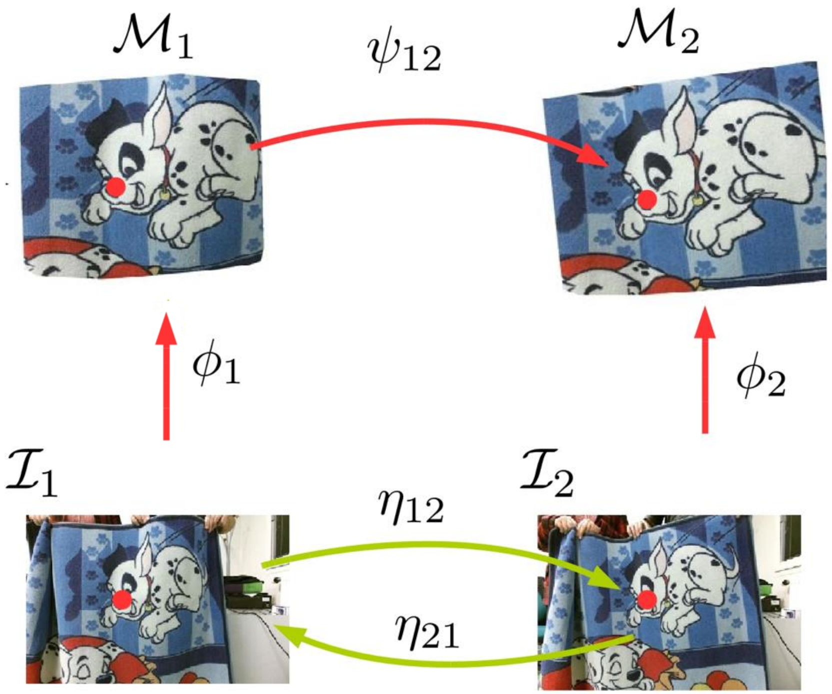
In order to preserve distances, consider rigid motion of tangent plane



Locally, $P_2 = R_{12}P_1 + T_{12}$

$J^T_{\psi_{12}} J_{\psi_{12}} = R_{12}^T R_{12} = I$

ISOMETRY: HOW TO PRESERVE GEODESICS?



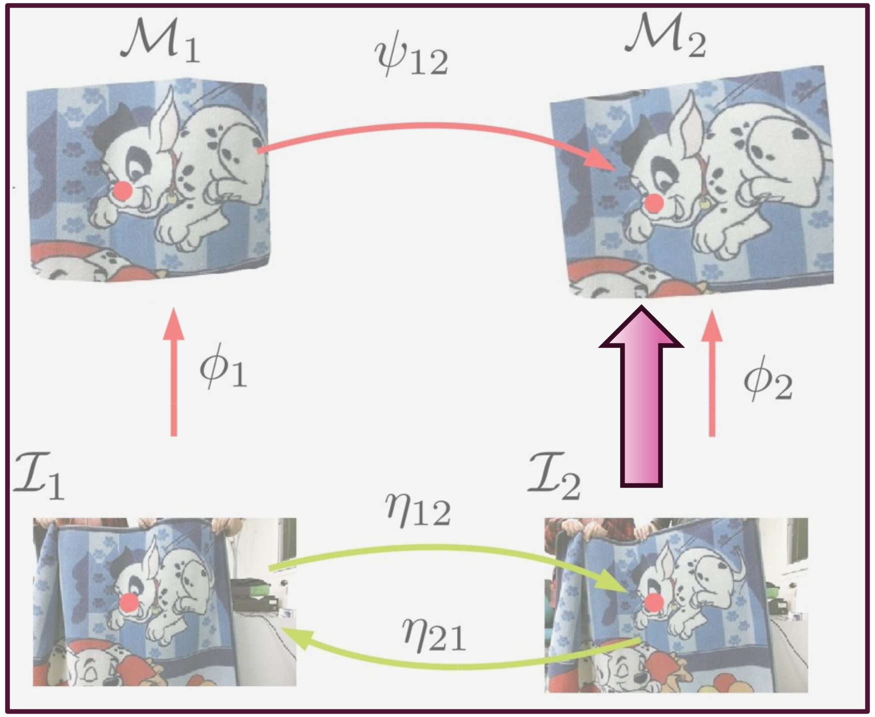
Registration : computed using optical flow or SIFT

Surface parametrization ϕ : perspective camera

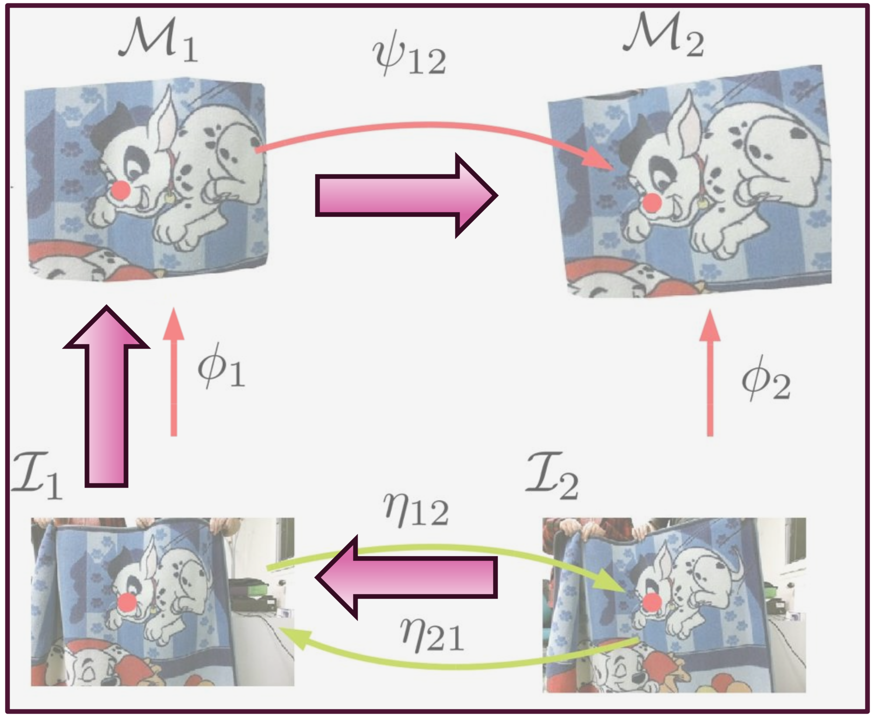
$$\phi = 1/z \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$

z : inverse of depth (unknown)

ISOMETRY: HOW TO PRESERVE GEODESICS?

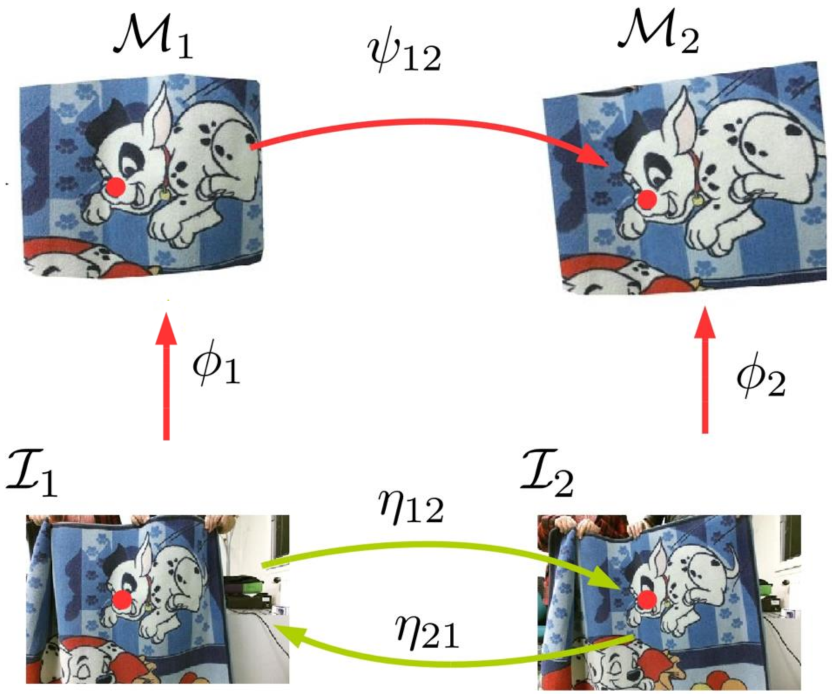


=



Local constraints at $(\mathcal{M}_2, \mathcal{I}_2)$ computed using ϕ_2 and $(\psi_{12}, \phi_1, \eta_{21})$ must be equal

ISOMETRY: HOW TO PRESERVE GEODESICS?



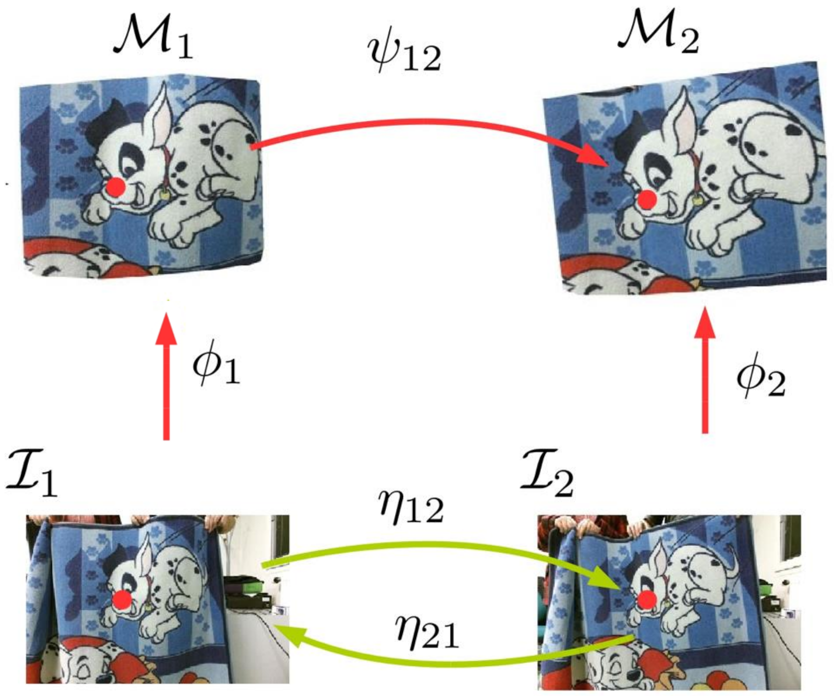
Registration η_{21} : computed using optical flow or SIFT
Surface parametrization ϕ : perspective camera

$$\phi = 1/z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

z : inverse of depth (unknown)

$$\phi_2 = \psi_{12} \circ \phi_1 \circ \eta_{21}$$

ISOMETRY: HOW TO PRESERVE GEODESICS?



Registration η_{21} : computed using optical flow or SIFT
Surface parametrization ϕ : perspective camera

$$\phi = 1/z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

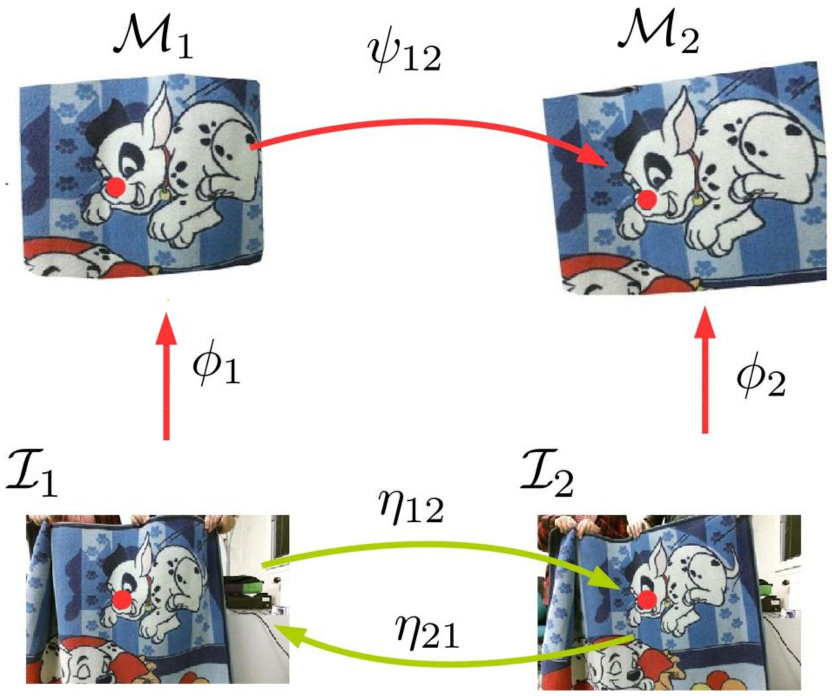
z : inverse of depth (unknown)

$$\phi_2 = \psi_{12} \circ \phi_1 \circ \eta_{21}$$

$$\mathbf{J}_{\phi_2} = \mathbf{J}_{\psi_{12}} \mathbf{J}_{\phi_1} \mathbf{J}_{\eta_{12}}$$

$$\mathbf{J}_{\phi_2}^{\top} \mathbf{J}_{\phi_2} = \mathbf{J}_{\eta_{12}}^{\top} \mathbf{J}_{\phi_1}^{\top} \mathbf{J}_{\phi_1} \mathbf{J}_{\eta_{12}}$$

ISOMETRY: HOW TO PRESERVE GEODESICS?



Registration η_{21} : computed using optical flow or SIFT
Surface parametrization ϕ : perspective camera

$$\phi = 1/z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

z : inverse of depth (unknown)

$$\phi_2 = \psi_{12} \circ \phi_1 \circ \eta_{21}$$

$$\mathbf{J}_{\phi_2} = \mathbf{J}_{\psi_{12}} \mathbf{J}_{\phi_1} \mathbf{J}_{\eta_{12}}$$

$$\mathbf{J}_{\phi_2}^\top \mathbf{J}_{\phi_2} = \mathbf{J}_{\eta_{12}}^\top \mathbf{J}_{\phi_1}^\top \mathbf{J}_{\phi_1} \mathbf{J}_{\eta_{12}}$$

5 variables, 3 equations: we need more

HOW TO SOLVE ?

What we did so far:

In order to preserve distances, consider rigid motion of tangent plane

Tangent plane : an infinitesimally close neighbor

HOW TO SOLVE ?

What we did so far:

In order to preserve distances, consider rigid motion of tangent plane

Tangent plane : an infinitesimally close neighbor

What if we look at the infinitesimally close neighbor of infinitesimally close neighbor??

DEFORMATION MODELING WITH LOCAL STRUCTURES

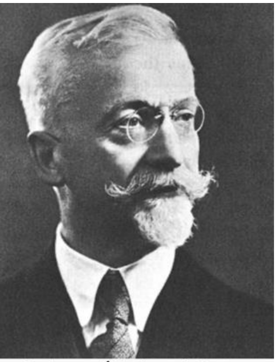
Preservation of local structures (tangent plane orientations and smoothness)

- Use Cartan's connections to restrict infinitesimally close tangent planes [Parashar et al, TPAMI 2019]

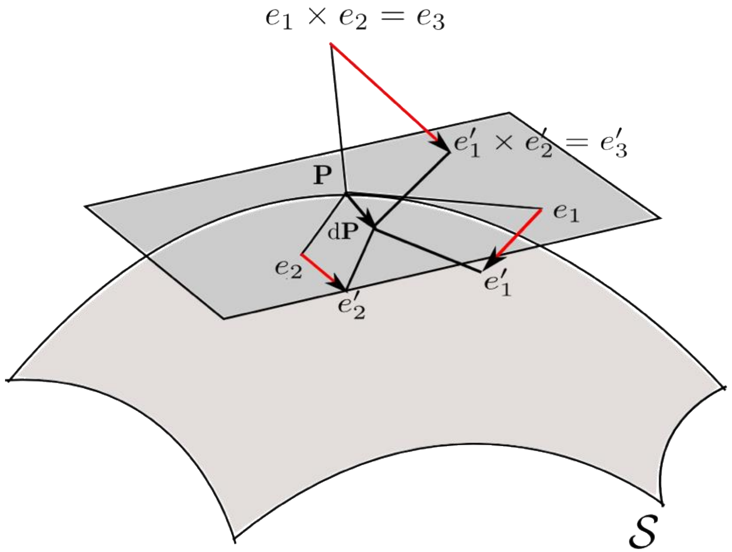
Infinitesimally close neighbor of infinitesimally close neighbor

=

Local changes in tangent planes



Professor Élie Joseph Cartan



dP is linear in terms of (e_1, e_2, e_3), so is de_i

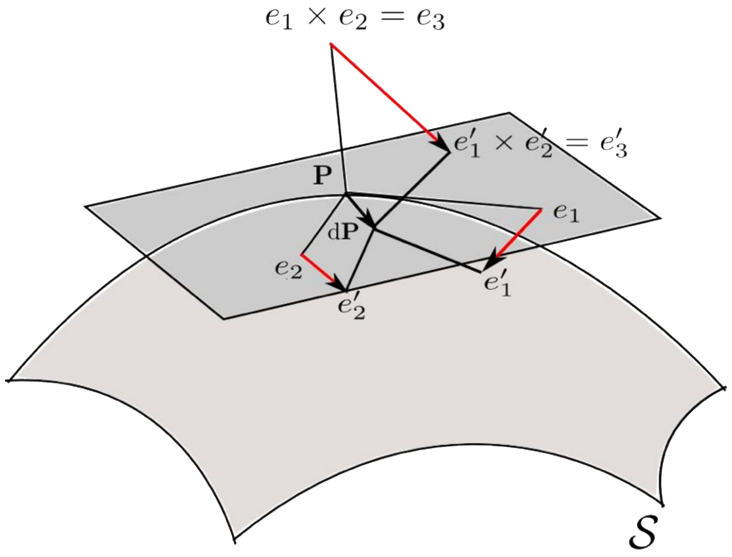
Connections are the combination weights

DEFORMATION MODELING WITH LOCAL STRUCTURES

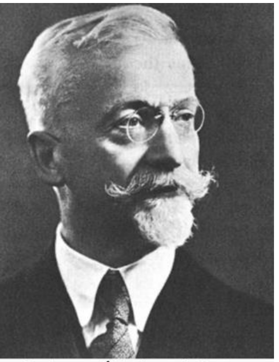
Preservation of local structures (tangent plane orientations and smoothness)

- Use Cartan's connections to restrict infinitesimally close tangent planes [Parashar et al, TPAMI 2019]

Local changes in tangent planes = connections

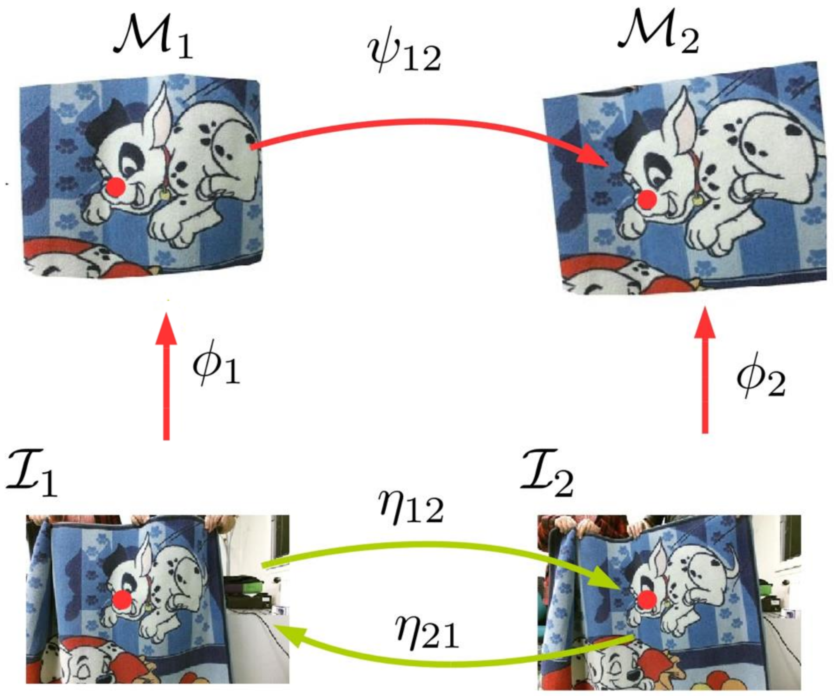


dP is linear in terms of (e_1, e_2, e_3), so is de_i
Connections are the combination weights



Professor Élie Joseph Cartan

ISOMETRY: HOW TO PRESERVE GEODESICS?

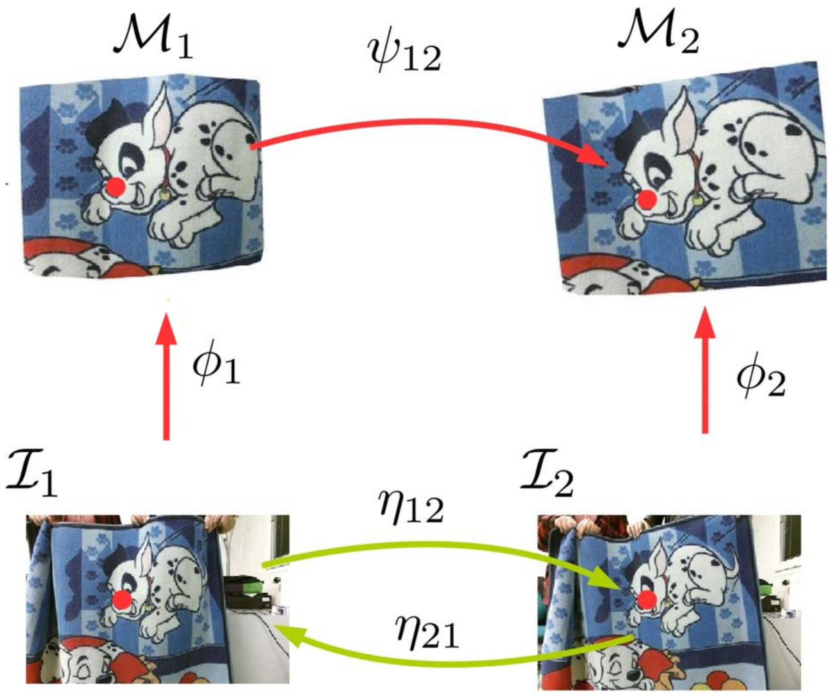


Metric tensor: $\mathbf{g} = \mathbf{J}_\phi^\top \mathbf{J}_\phi$ Connections: $\mathbf{\Gamma} = \frac{\partial \mathbf{g}}{\mathbf{g}}$

Connections at $(\mathcal{M}_2, \mathcal{I}_2)$ computed using ϕ_2 and $(\psi_{12}, \phi_1, \eta_{21})$ must be equal

This gives 2 additional equations

ISOMETRY: HOW TO PRESERVE GEODESICS?



Metric tensor: $\mathbf{g} = \mathbf{J}_\phi^\top \mathbf{J}_\phi$ Connections: $\mathbf{\Gamma} = \frac{\partial \mathbf{g}}{\mathbf{g}}$

Connections at $(\mathcal{M}_2, \mathcal{I}_2)$ computed using ϕ_2 and $(\psi_{12}, \phi_1, \eta_{21})$ must be equal

$$[\mathbf{n}]_\times^\top \mathbf{S} [\mathbf{n}]_\times = 0$$

$$\mathbf{S} = \mathbf{H}^\top \mathbf{H} - \mathbf{I}$$

$$\mathbf{H}^\top = \begin{pmatrix} \mathbf{I}_{2\times 2} & 0 \\ -\bar{\mathbf{x}}^\top & 1 \end{pmatrix} \begin{pmatrix} \mathbf{J}_\eta^\top & \mathbf{m} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{I}_{2\times 2} & 0 \\ \mathbf{x}^\top & 1 \end{pmatrix}$$

$\mathbf{n}$: normal at ϕ_2

$\mathbf{m}$: second order derivatives at η_{21}

3D PLANAR STRUCTURE AND HOMOGRAPHY

Homography Decomposition

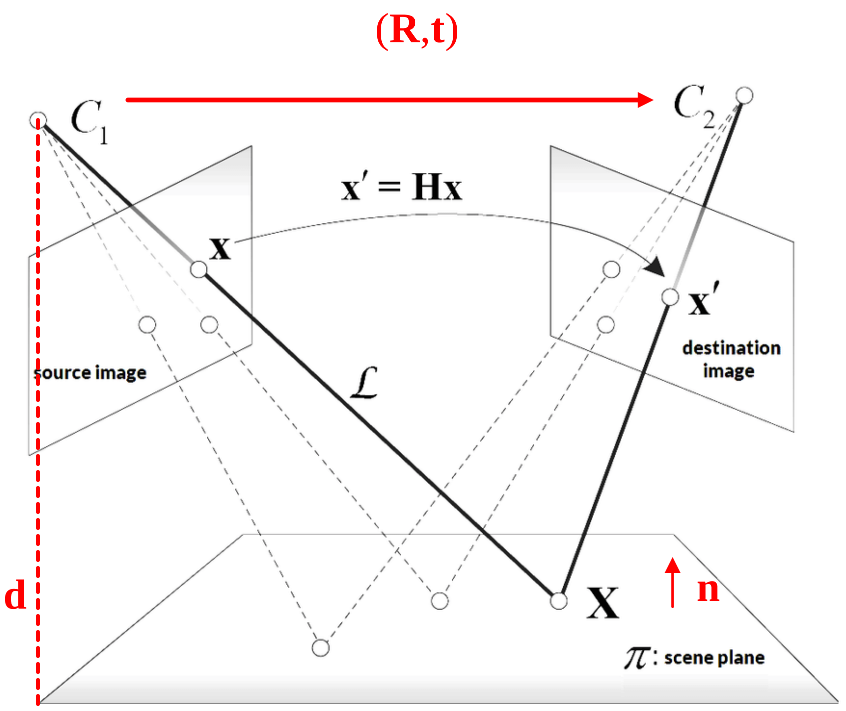
Given $\mathbf{H}$ computed with DLT, one can obtain closed-form solution to normal

$$\mathbf{S} = \mathbf{H}^\top \mathbf{H} - \mathbf{I} = \begin{bmatrix} s_{11} & s_{12} & s_{13} \\ s_{12} & s_{22} & s_{23} \\ s_{13} & s_{23} & s_{33} \end{bmatrix} \qquad M_{\mathbf{S}_{11}} = -\begin{vmatrix} s_{22} & s_{23} \\ s_{23} & s_{33} \end{vmatrix} = s_{23}^2 - s_{22}s_{33} \geq 0$$

$$\mathbf{n}'_a(s_{11}) = \begin{bmatrix} s_{11} \\ s_{12} + \sqrt{M_{\mathbf{S}_{33}}} \\ s_{13} + \epsilon_{23}\sqrt{M_{\mathbf{S}_{22}}} \end{bmatrix}; \qquad \mathbf{n}'_b(s_{11}) = \begin{bmatrix} s_{11} \\ s_{12} - \sqrt{M_{\mathbf{S}_{33}}} \\ s_{13} - \epsilon_{23}\sqrt{M_{\mathbf{S}_{22}}} \end{bmatrix}$$

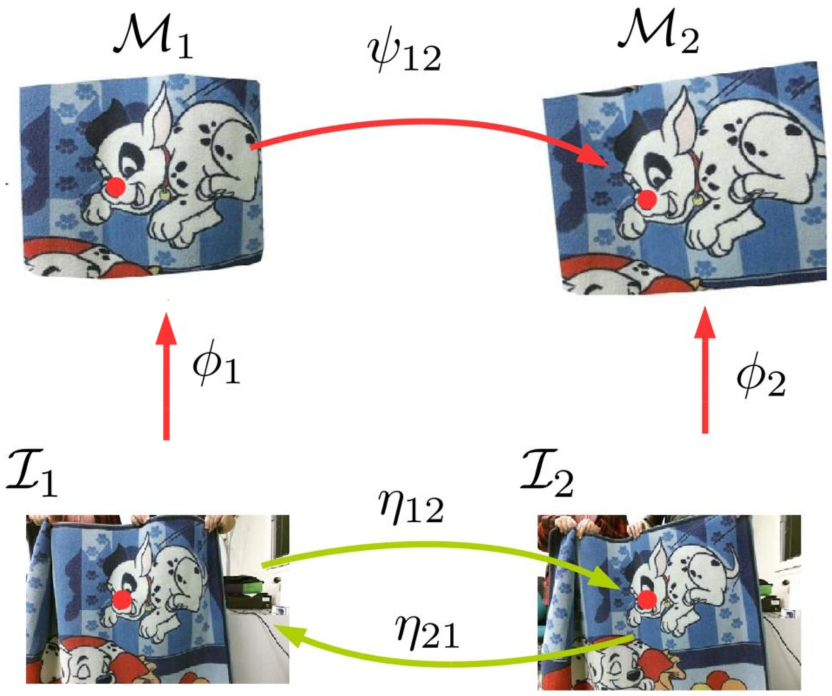
$$\mathbf{n}'_a(s_{22}) = \begin{bmatrix} s_{12} + \sqrt{M_{\mathbf{S}_{33}}} \\ s_{22} \\ s_{23} - \epsilon_{13}\sqrt{M_{\mathbf{S}_{11}}} \end{bmatrix}; \qquad \mathbf{n}'_b(s_{22}) = \begin{bmatrix} s_{12} - \sqrt{M_{\mathbf{S}_{33}}} \\ s_{22} \\ s_{23} + \epsilon_{13}\sqrt{M_{\mathbf{S}_{11}}} \end{bmatrix}$$

$$\mathbf{n}'_a(s_{33}) = \begin{bmatrix} s_{13} + \epsilon_{12}\sqrt{M_{\mathbf{S}_{22}}} \\ s_{23} + \sqrt{M_{\mathbf{S}_{11}}} \\ s_{33} \end{bmatrix}; \qquad \mathbf{n}'_b(s_{33}) = \begin{bmatrix} s_{13} - \epsilon_{12}\sqrt{M_{\mathbf{S}_{22}}} \\ s_{23} - \sqrt{M_{\mathbf{S}_{11}}} \\ s_{33} \end{bmatrix}$$



[Malis et Vargas, 2007].Deeper understanding of the homography decomposition for vision-based control .

ISOMETRIC NRSFM = HOMOGRAPHY DECOMPOSITION OF INFINITESIMALLY SMALL PLANES



Registration η_{21} : computed using optical flow or SIFT
Surface parametrization ϕ : perspective camera

$$\phi = 1/z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

z : inverse of depth (unknown)

$$\mathbf{J}_{\phi_2}^\top \mathbf{J}_{\phi_2} = \mathbf{J}_{\eta_{12}}^\top \mathbf{J}_{\phi_1}^\top \mathbf{J}_{\phi_1} \mathbf{J}_{\eta_{12}}$$

5 variables, 3 equations

$$[\mathbf{n}]_\times^\top \mathbf{S} [\mathbf{n}]_\times = 0$$

2 variables, 2 equations

2 possible solutions, disambiguation with neighbors

NRSFM WITH DIFFERENTIAL GEOMETRY

3DVfX [Parashar et al, Eurographics 2019]

ARE RIGID AND DEFORMABLE OBJECTS REALLY DIFFERENT?

- Rigid objects: Cameras related by $\mathbf{R},\mathbf{T}$

Homography computed using DLT

- Deformable objects: Motion in terms of $\mathbf{R},\mathbf{T}$ and scale [Parashar et al, TPAMI 2024]

Homography: local projective transformation

$$\mathbf{H}^\top = \begin{pmatrix} \mathbf{I}_{2\times 2} & 0 \\ -\bar{\mathbf{x}}^\top & 1 \end{pmatrix} \begin{pmatrix} \mathbf{J}_\eta^\top & \mathbf{m} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{I}_{2\times 2} & 0 \\ \mathbf{x}^\top & 1 \end{pmatrix}$$

WHAT DID WE LEARN?

- 3D reconstruction of rigid objects whether by traditional (COLMAP/GLOMAP) or non-traditional methods (Nerfs/Gaussian splatting/Mast3r) is successful due to strong foundation of true geometric awareness from images.
- 3D reconstruction of deformable objects traditionally is studied with statistical approximations. It has been successful but strengthening the true geometric awareness is important to bridge the gap between their performance with the rigid counterparts.