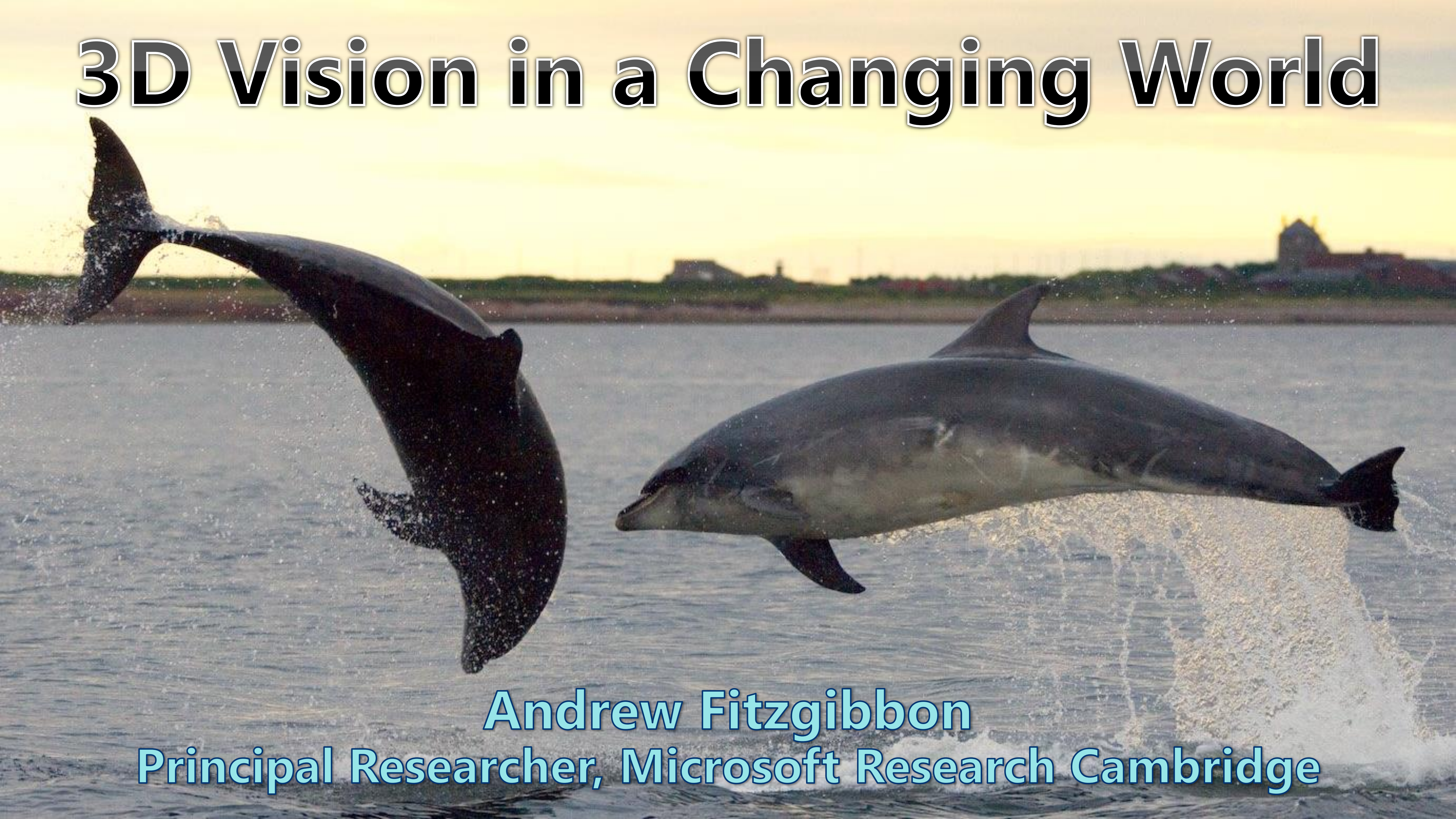
3D Vision in a Changing World



Andrew Fitzgibbon
Principal Researcher, Microsoft Research Cambridge

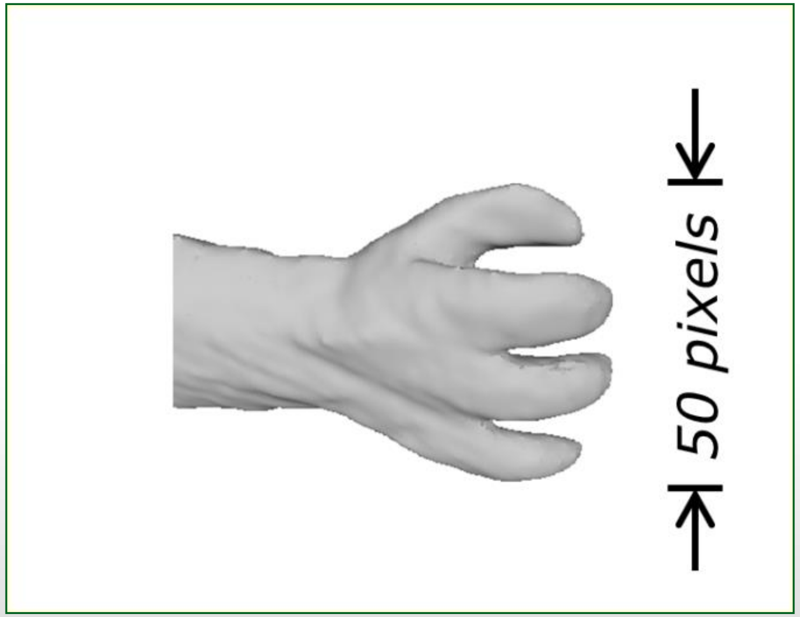
Collaborators

Mukta Prasad
ETH Zurich

Tom Cashman
TranscenData Europe

Pushmeet Kohli
Microsoft Research

Alex Rav-Acha
SightEra Technologies



- 1998: we computed a decent 3D reconstruction of a 36-frame sequence
- Giving 3D super-resolution
- And set ourselves the goal of solving a 1500-frame sequence
- Leading to...

[FCZ98] Fitzgibbon, Cross & Zisserman, SMILE 1998

Early Work

3D from Monocular RGB video

Input: Standard video

Processing:

1. Detect high-contrast points
2. Track from frame to frame
3. Compute most likely 3D structure

Usage: augmented reality

EARLY WORK

Microsoft

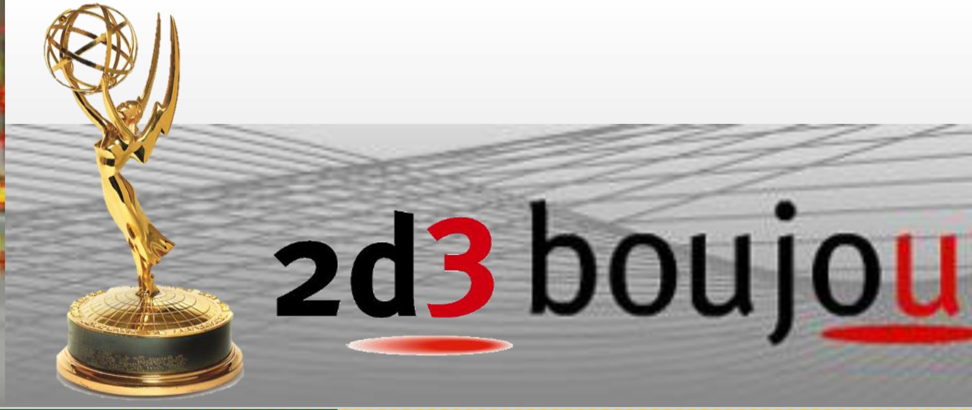
3D from Monocular RGB video

Input: Standard video

Processing:

1. Detect high-contrast points
2. Track from frame to frame
3. Compute most likely 3D structure

Usage: augmented reality

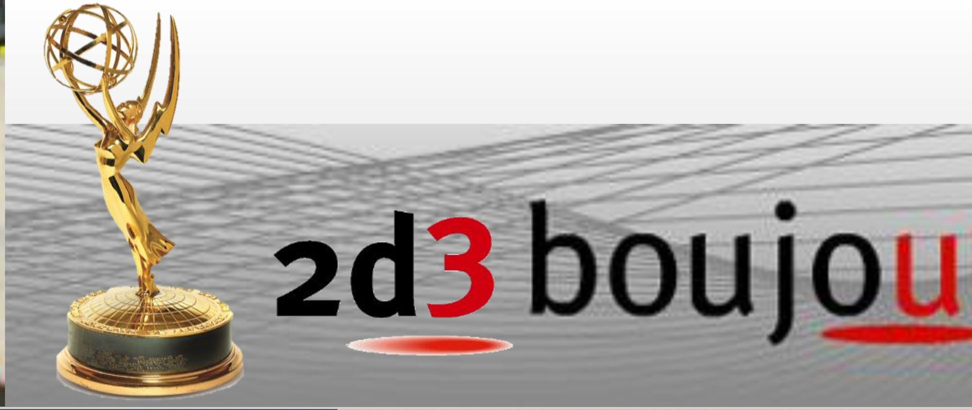


3D from Monocular RGB video

Input: Standard video

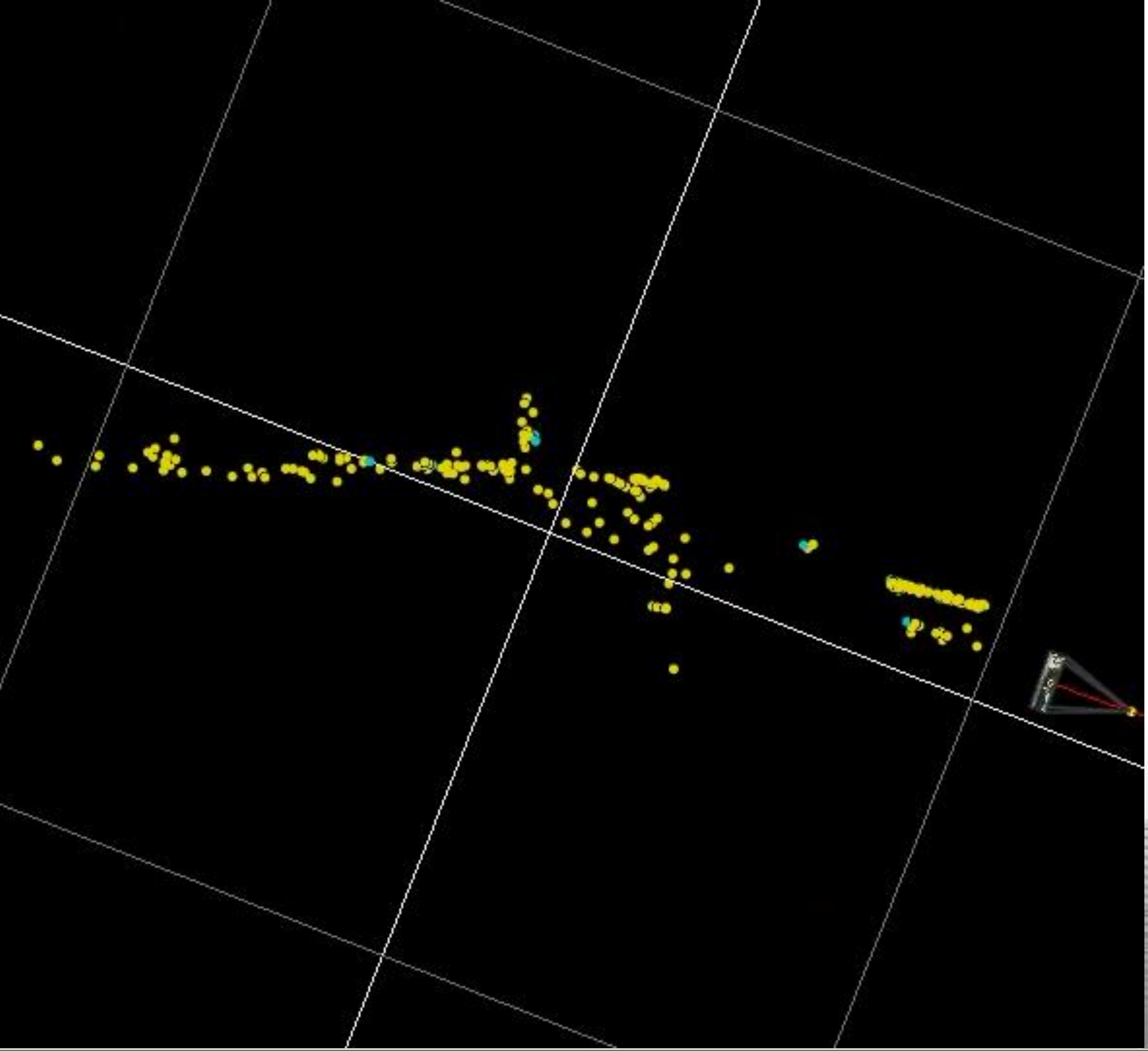
Processing:
1. Detect high-contrast points
2. Track from frame to frame
3. Compute most likely 3D structure

Usage: augmented reality



EARLY WORK

Microsoft



3D from Monocular RGB video

Input: Standard video

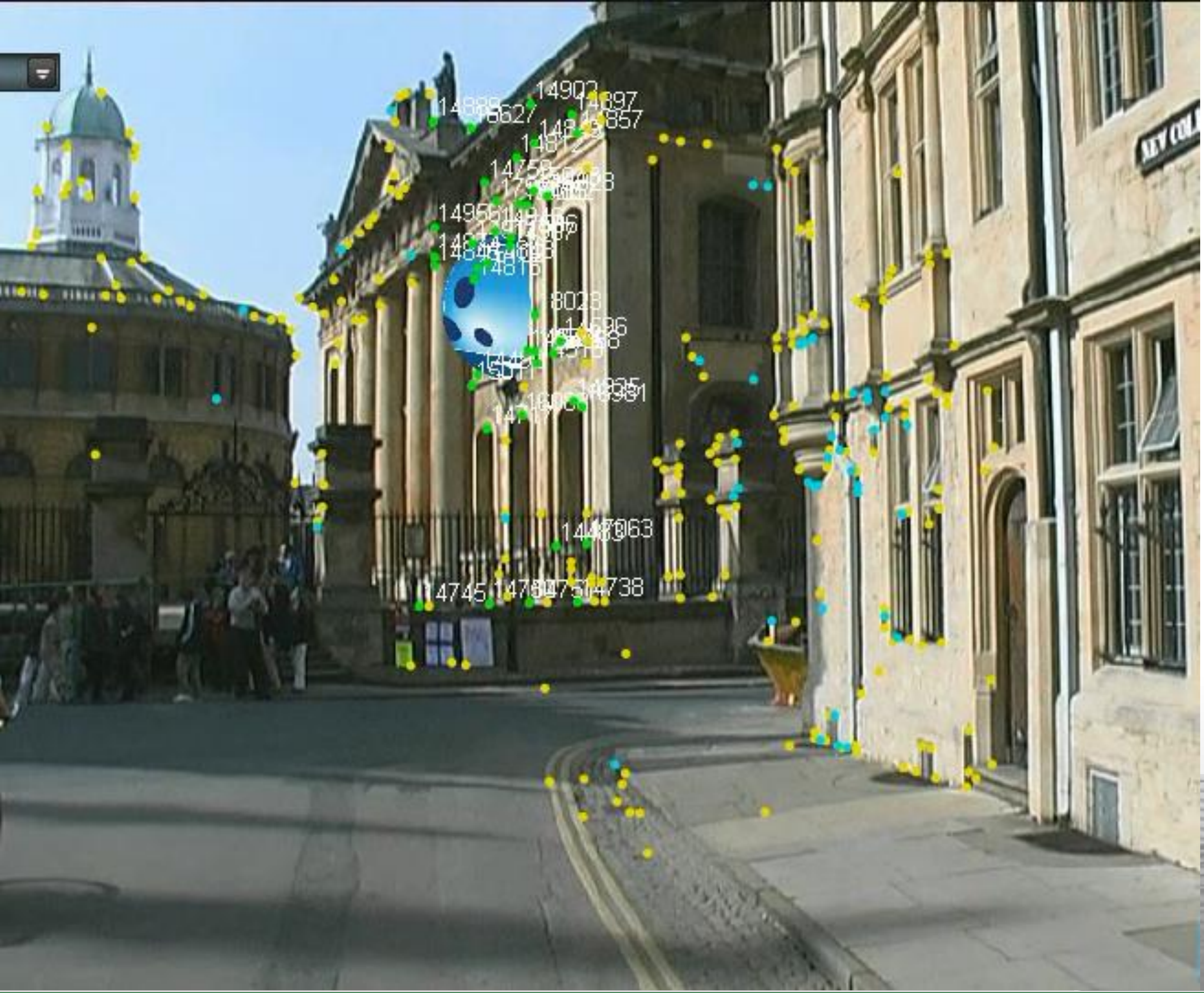
Processing:

1. Detect high-contrast points
2. Track from frame to frame
3. Compute most likely 3D structure

Usage: augmented reality

Early Work

Microsoft



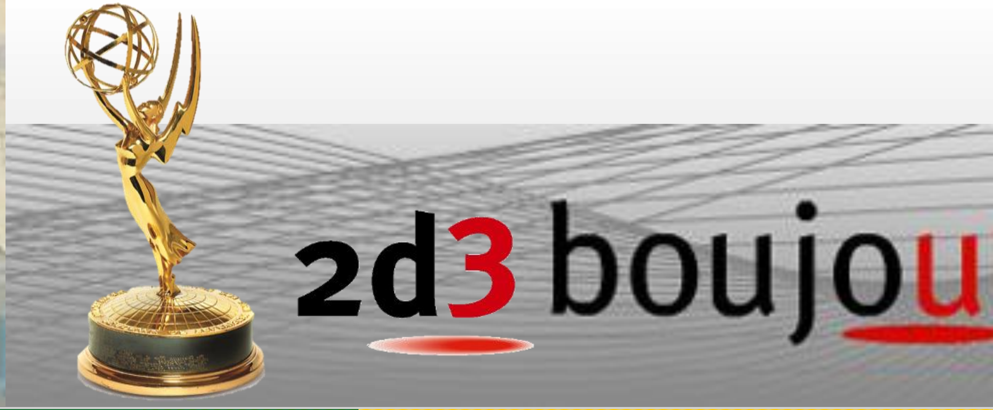
3D from Monocular RGB video

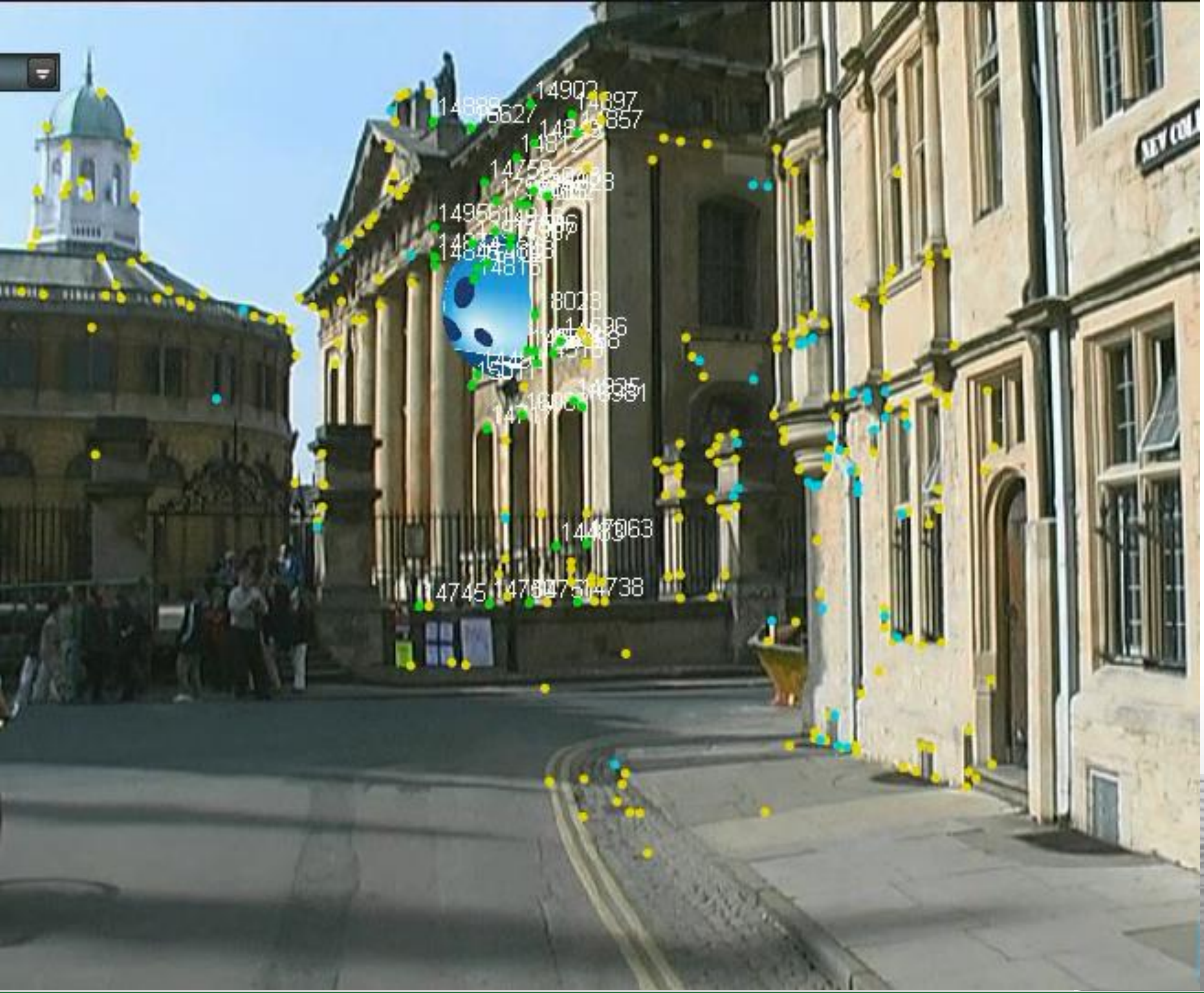
Input: Standard video

Processing:

1. Detect high-contrast points
2. Track from frame to frame
3. Compute most likely 3D structure

Usage: augmented reality





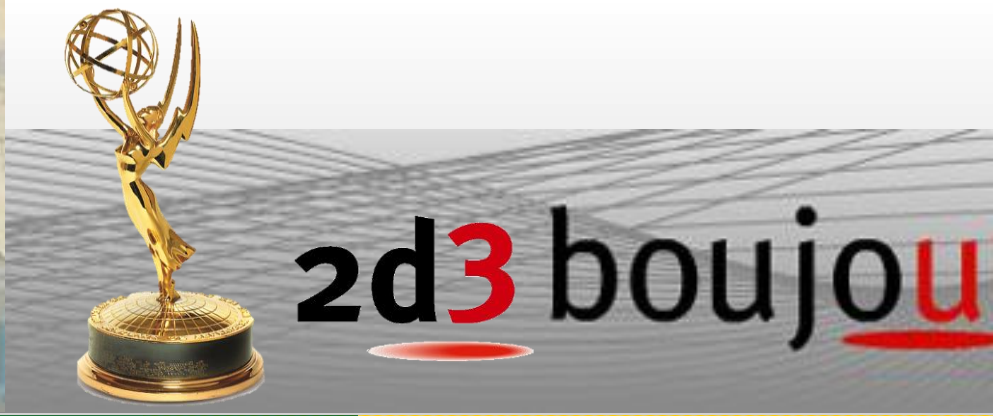
3D from Monocular RGB video

Input: Standard video

Processing:

1. Detect high-contrast points
2. Track from frame to frame
3. Compute most likely 3D structure

Usage: augmented reality



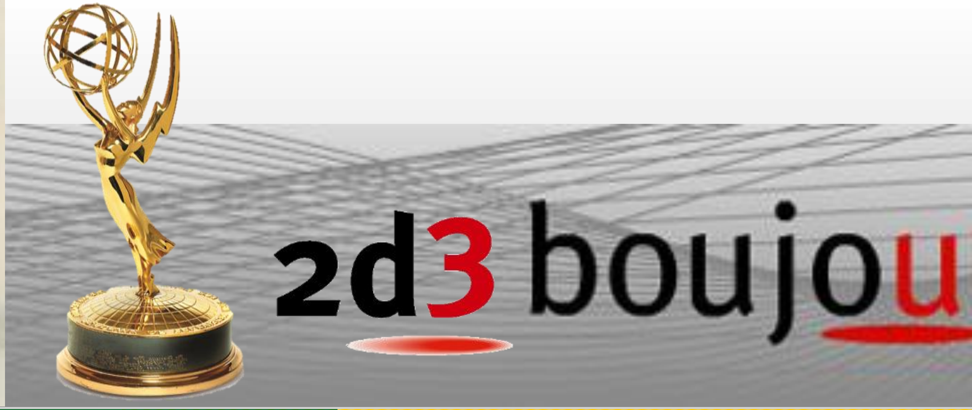
Early Work

3D from Monocular RGB video

Input: Standard video

Processing:
1. Detect high-contrast points
2. Track from frame to frame
3. Compute most likely 3D structure

Usage: augmented reality



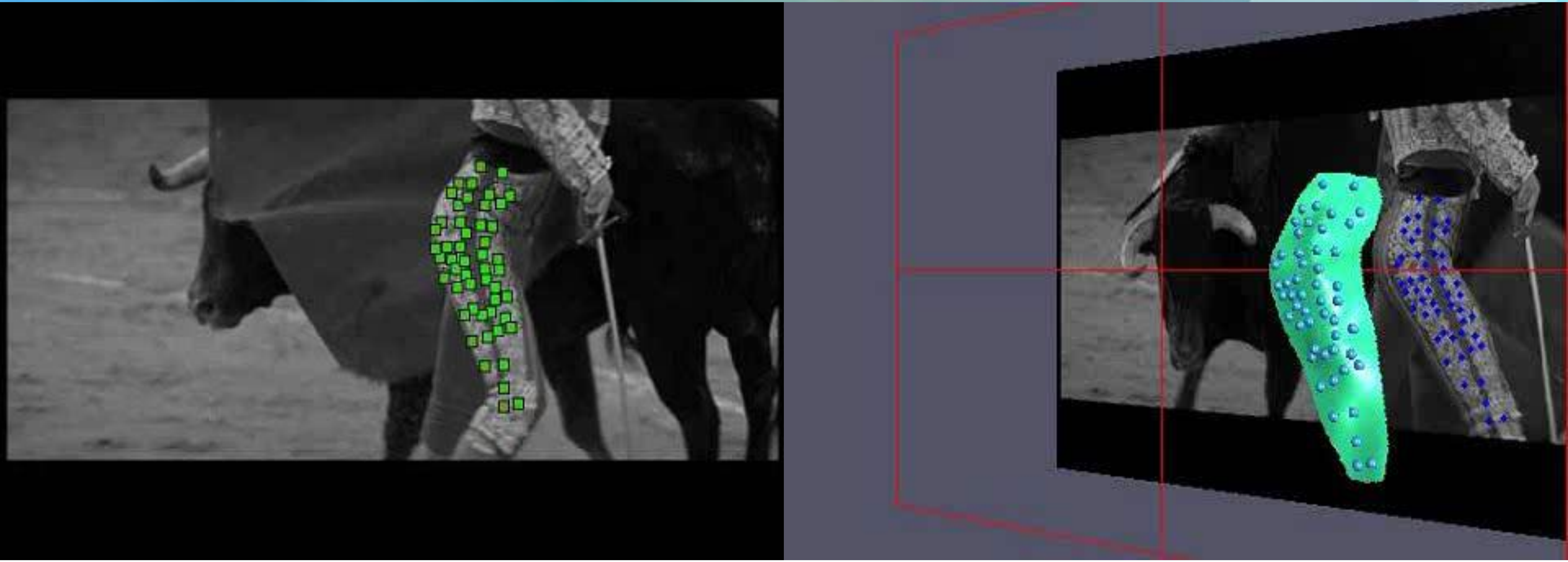
EARLY WORK

But… so flat, so dull…

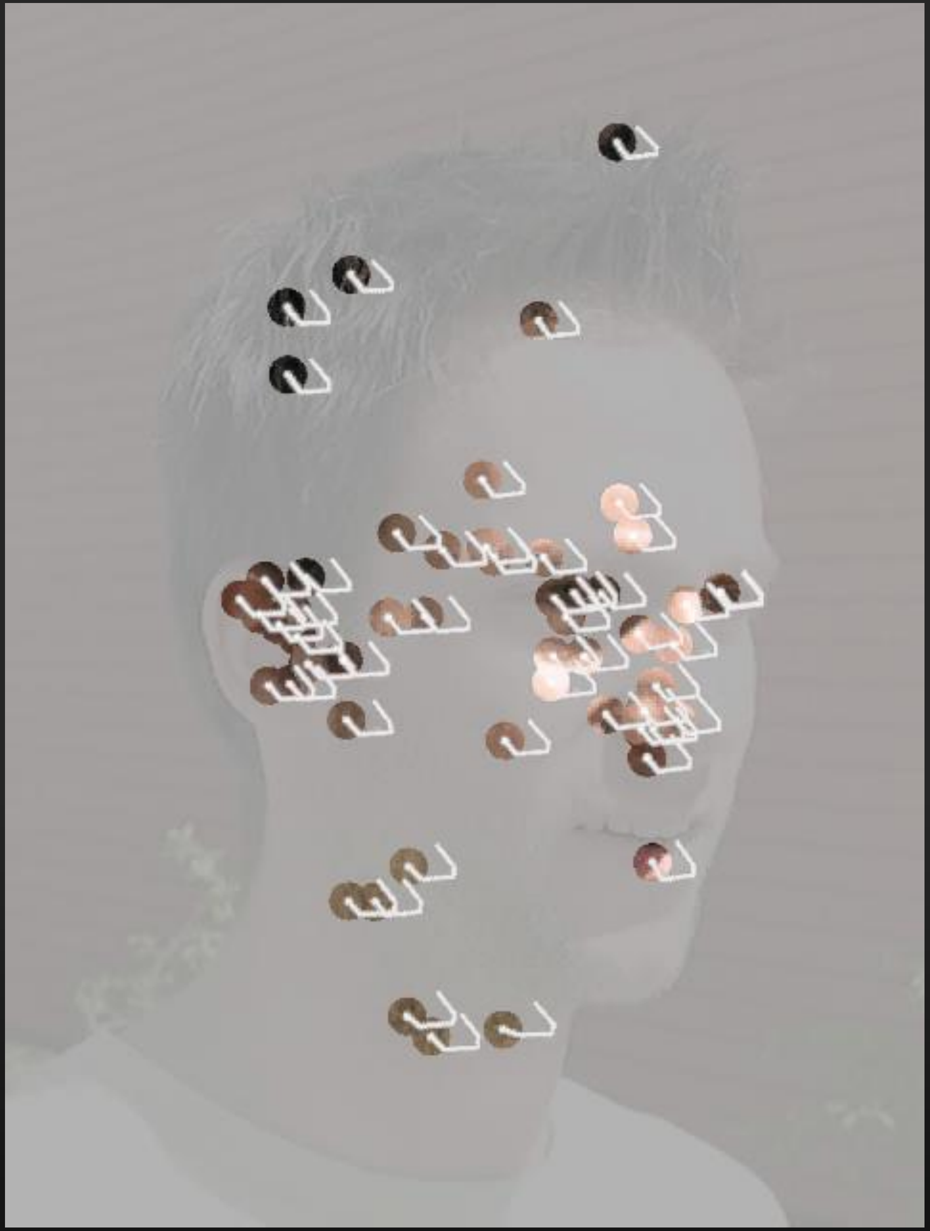


UNWRAPMOSAICS

How do I do it?

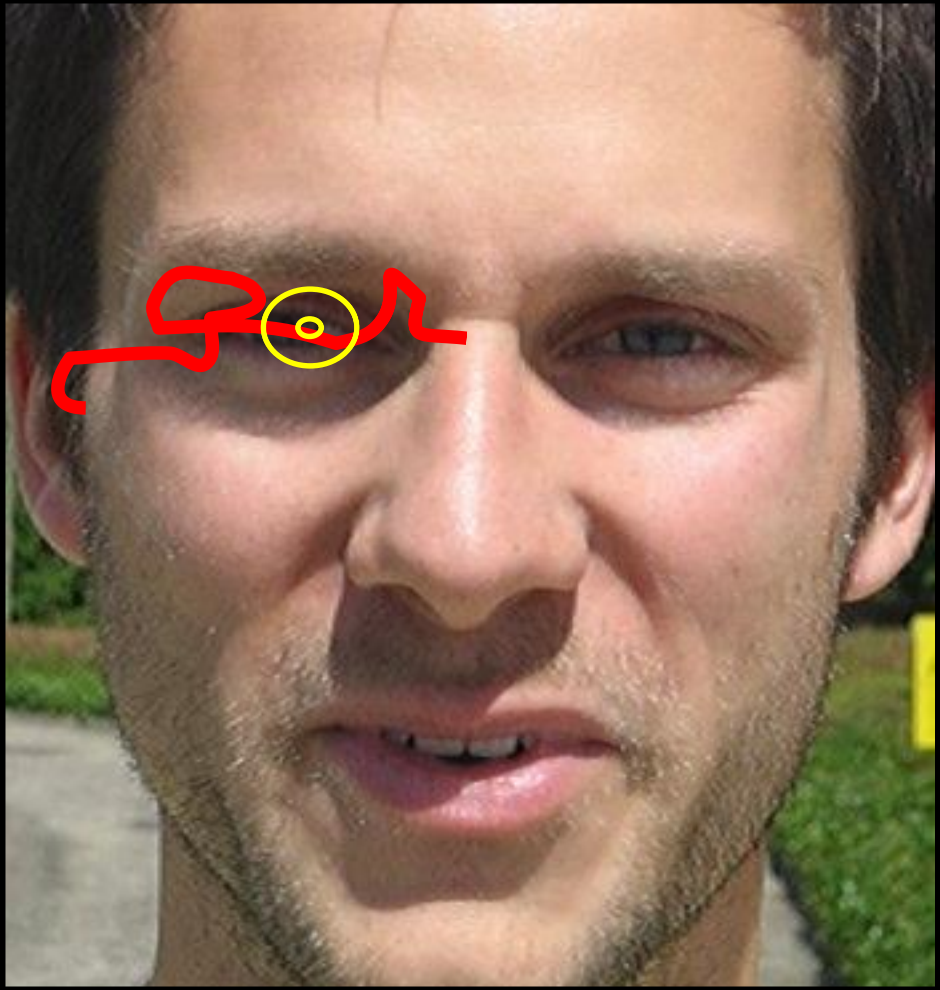


Non-Rigid Structure from Motion
C Bregler, L Torresani, A Hertzmann, H Biermann
CVPR 2000 – PAMI 2008



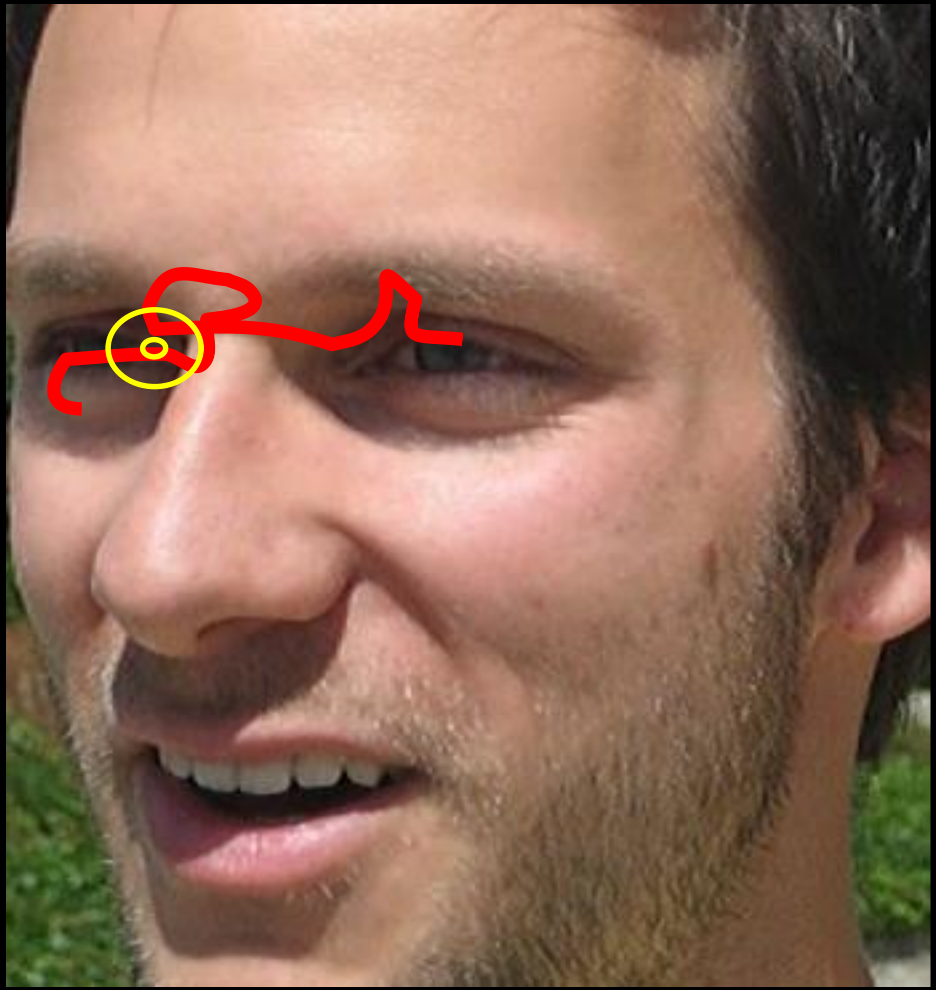
(311, 308)

$$\begin{bmatrix} 311 \\ 308 \\ \\ \\ \\ \\ \\ \end{bmatrix}$$



(204, 285)

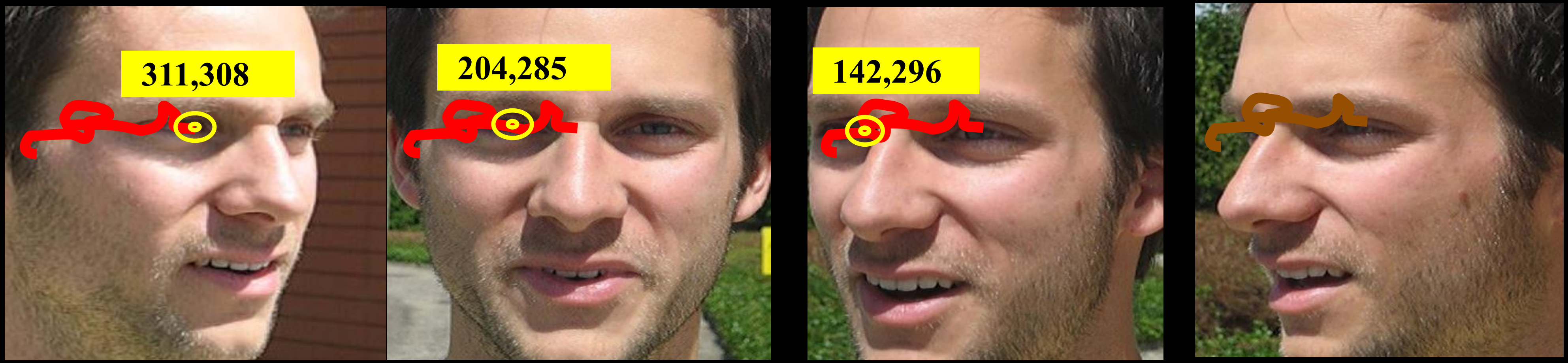
311
308
204
285



(142, 296)

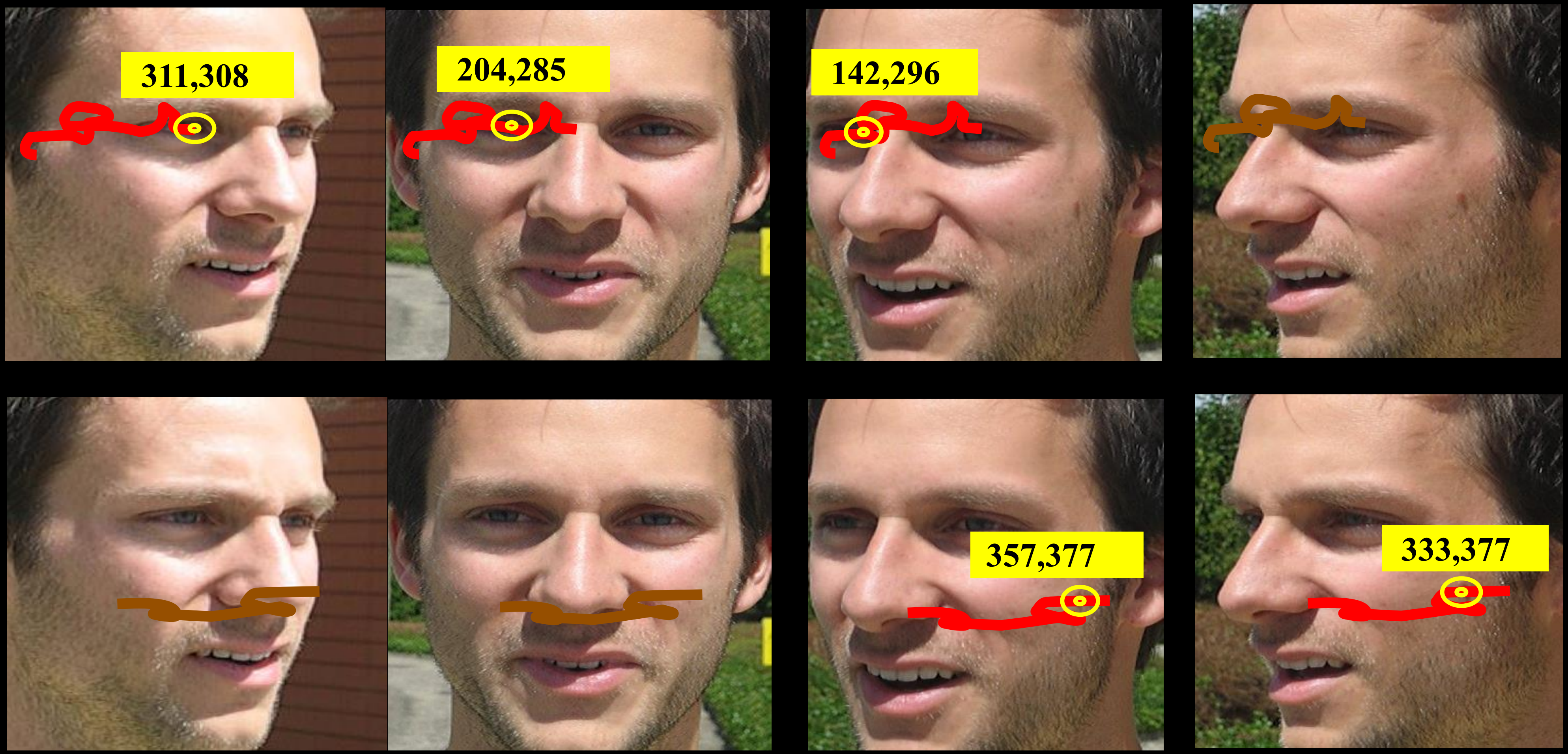
311
308
204
285
142
296

[311, 308, 204, 285, 142, 296, *, *]

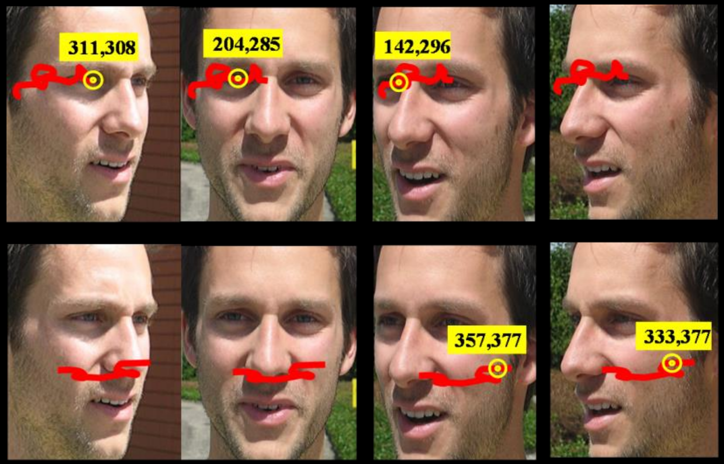


2T

$$\begin{bmatrix} 311 \\ 308 \\ 204 \\ 285 \\ 142 \\ 296 \\ * \\ * \end{bmatrix}$$



311	*
308	*
204	*
285	*
142	357
296	377
*	333
*	377

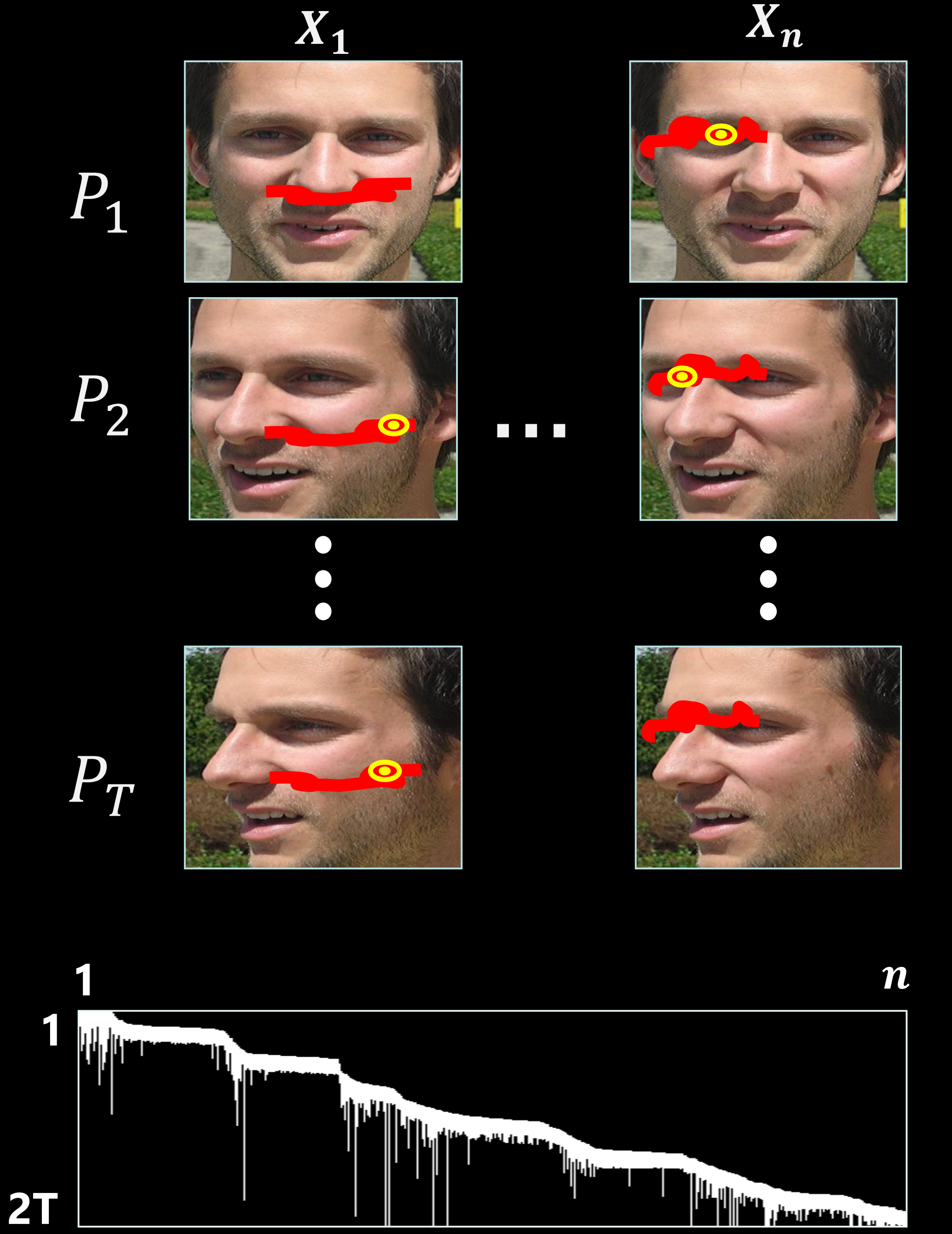


⋮

311	*			
308	*			
204	*			
285	*	⋮	⋮	⋮
142	357			
296	377			
*	333			
*	377			

(For this example: ntracks = 1135, T = 227)

Measurement Matrix: M



(For this example: ntracks = 1135, T = 227)

Derive $M = P\,X$, and factorize

Embedding

$$M_{:,i} = \pi(X_i) \qquad \pi: \mathbb{R}^r \mapsto \mathbb{R}^{2T}$$

Orthographic: linear (in X) embedding in $\mathbb{R}^4$

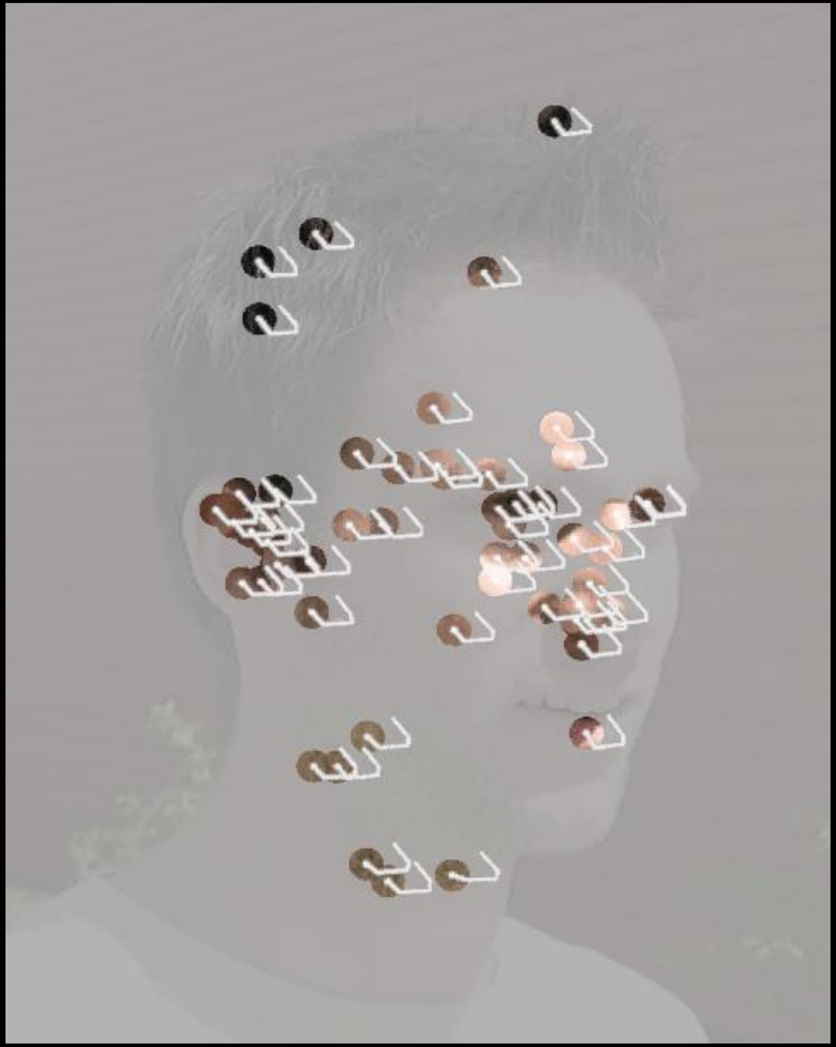
Perspective: (slightly) nonlinear embedding in $\mathbb{R}^3$

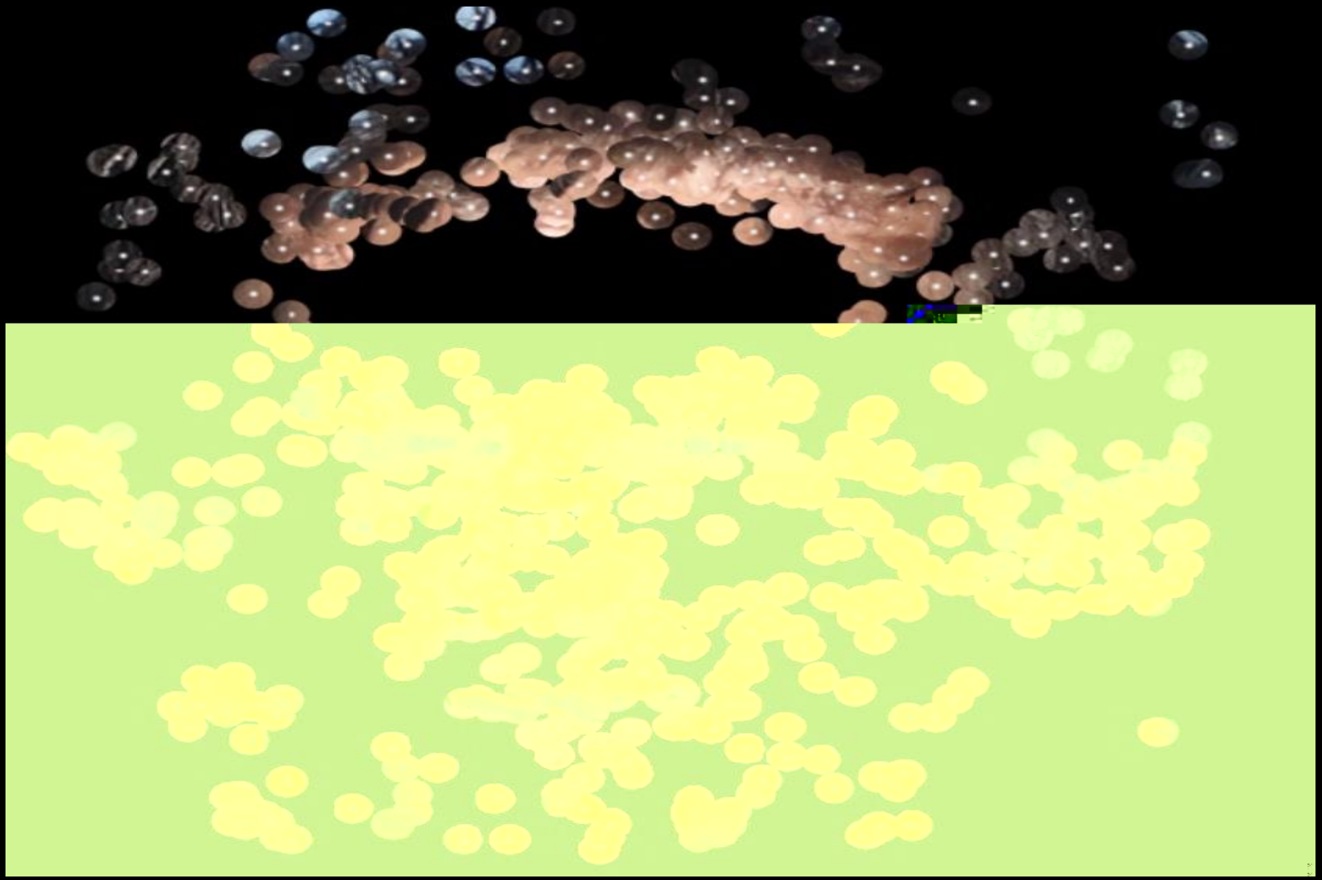
Previous work on nonrigid case: embed into $\mathbb{R}^{3K}$

Our big idea: surfaces are mappings $\mathbb{R}^2 \mapsto \mathbb{R}^3$

So embed (nonlinearly) into $\mathbb{R}^2$

Nonlinear embedding into $\mathbb{R}^2$



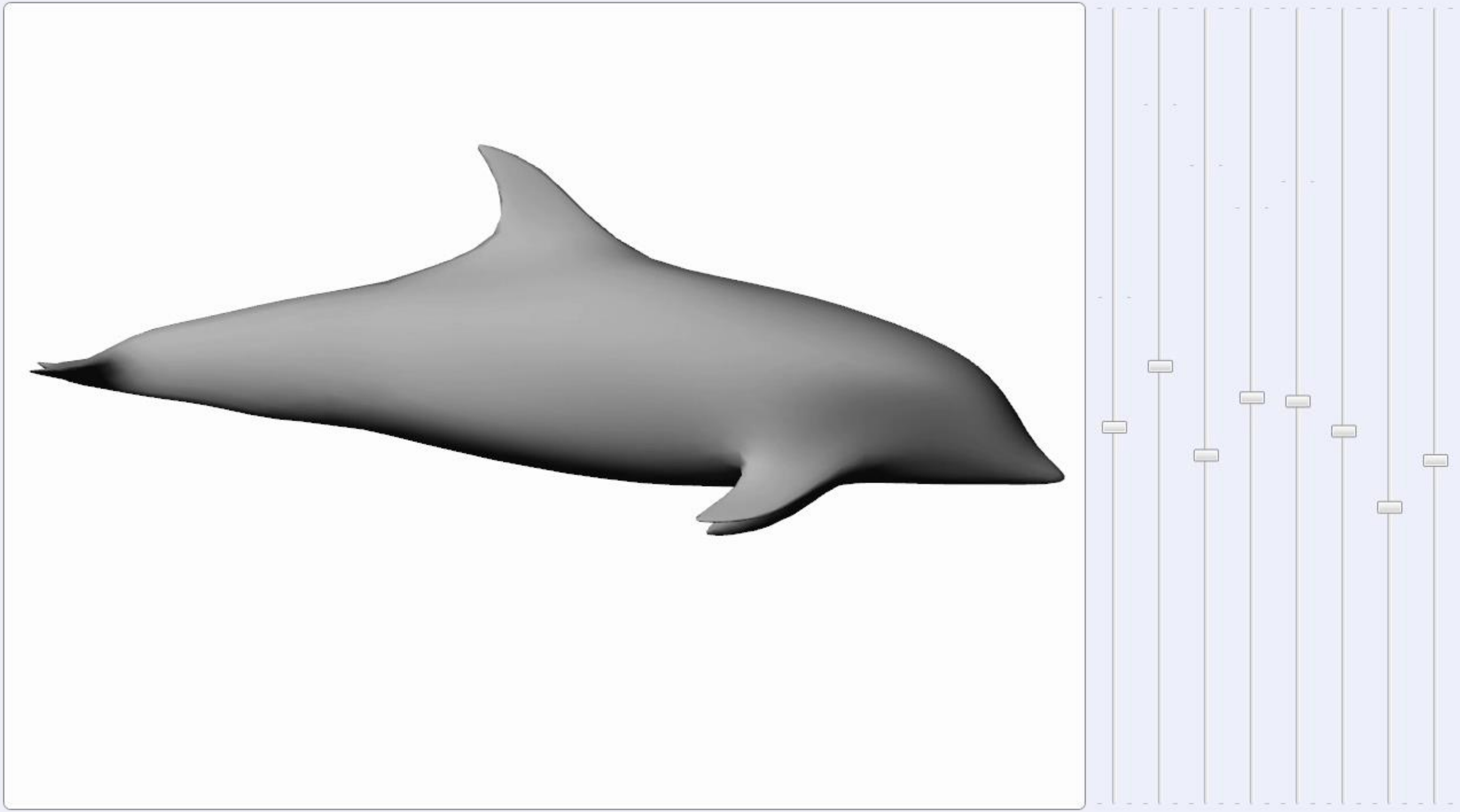


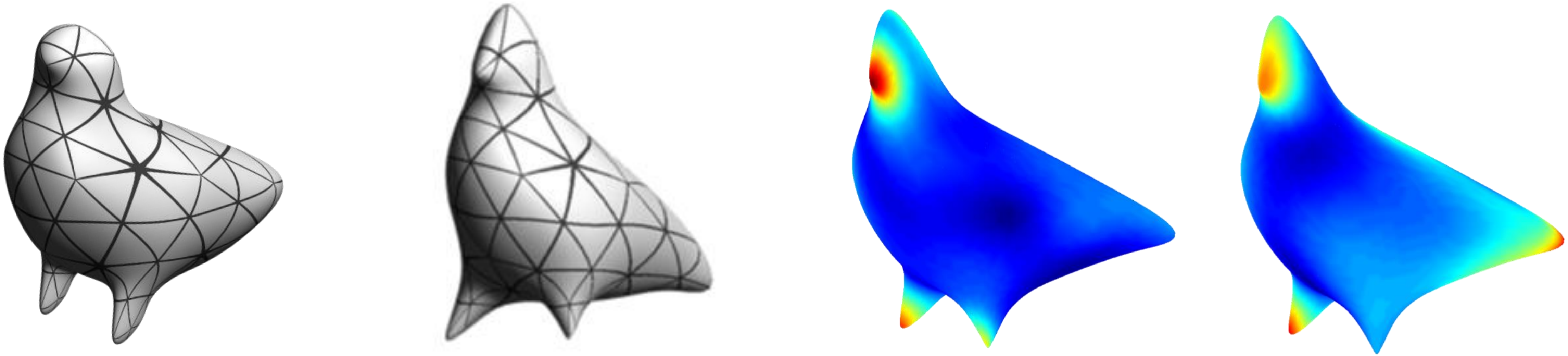

Microsoft
Research
Cambridge

dolphins



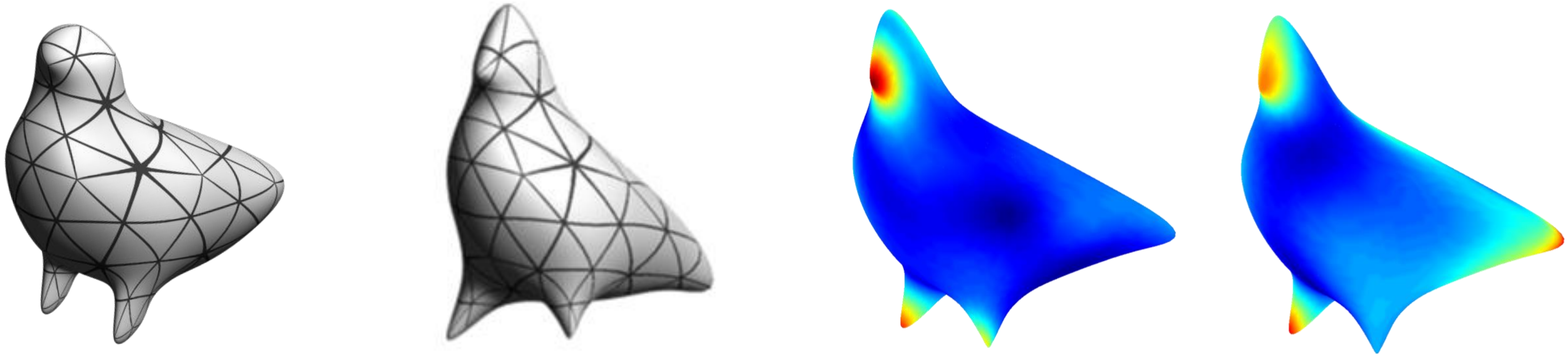




$$\mathcal{X}_n = \alpha_{n0}\,\mathcal{B}_0 + \alpha_{n1}\,\mathcal{B}_1 + \alpha_{n2}\,\mathcal{B}_2$$

$$\mathcal{X}_n = \sum_{k=0}^{K} \alpha_{nk}\mathcal{B}_k$$

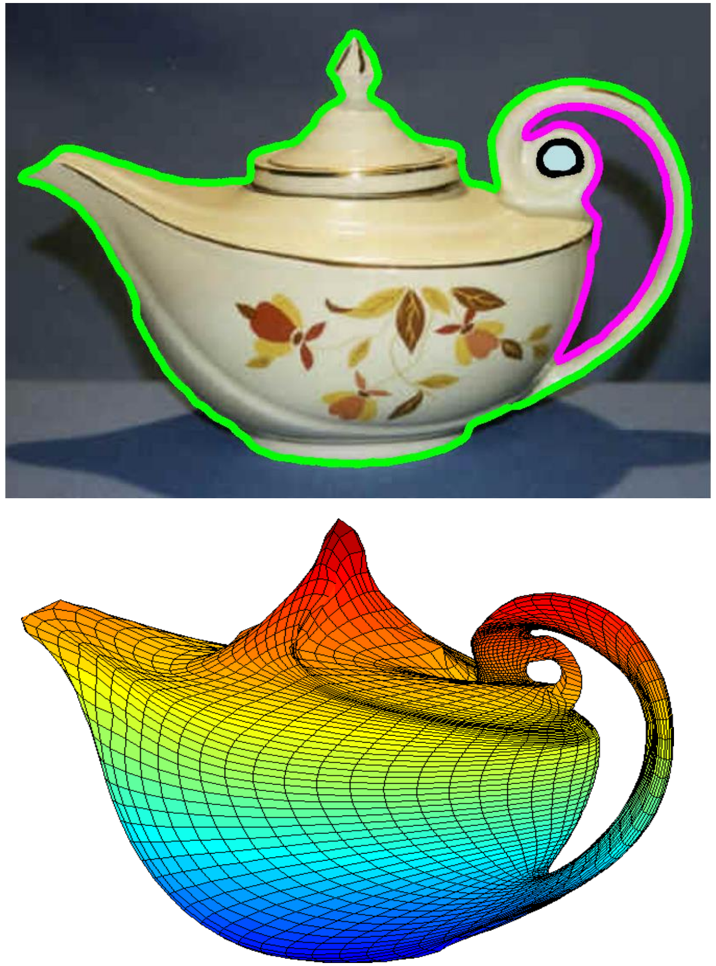


$$\mathcal{X}_n = \mathcal{B}_0 + \alpha_{n1}\mathcal{B}_1 + \alpha_{n2}\mathcal{B}_2$$

$$\mathcal{X}_n = \sum_{k=0}^{K} \alpha_{nk}\mathcal{B}_k$$

So I want a morphable model. What can I do?

[Prasad, Fitzgibbon, Zisserman]



3D from Single Images

- Automatic approaches not [yet] robust for curved surfaces
- Manual approaches require detailed annotation of many images
- And still need work for inter-model registration

3D Class Models from Images

1. Wireframe models

2. Subdivision surface models

Wireframe “Armature” Models

Calder, Alexander - *"Cow"* - (1929)

- Model class defined by 3D wireframe curves:
  - Sharp silhouettes
  - Internal edges

Wireframe “Armature” Models

[Prasad, Fitzgibbon, Zisserman, CVPR 2010]

Training images



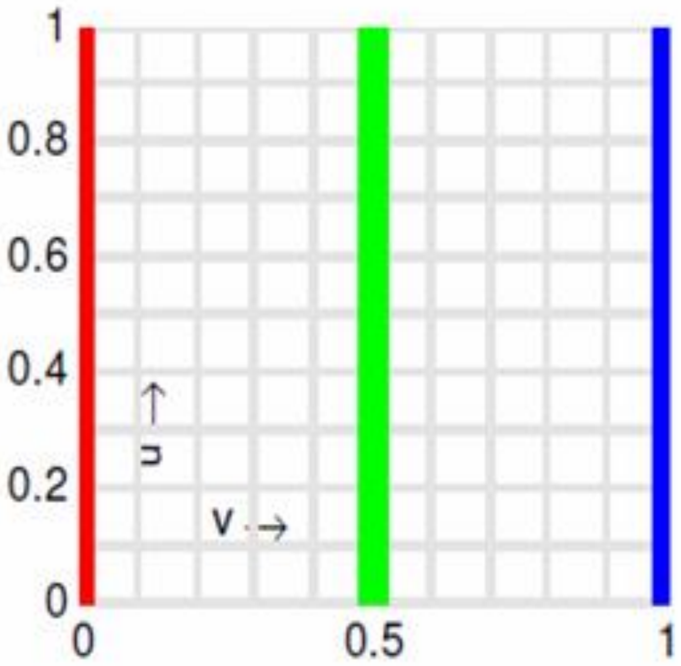


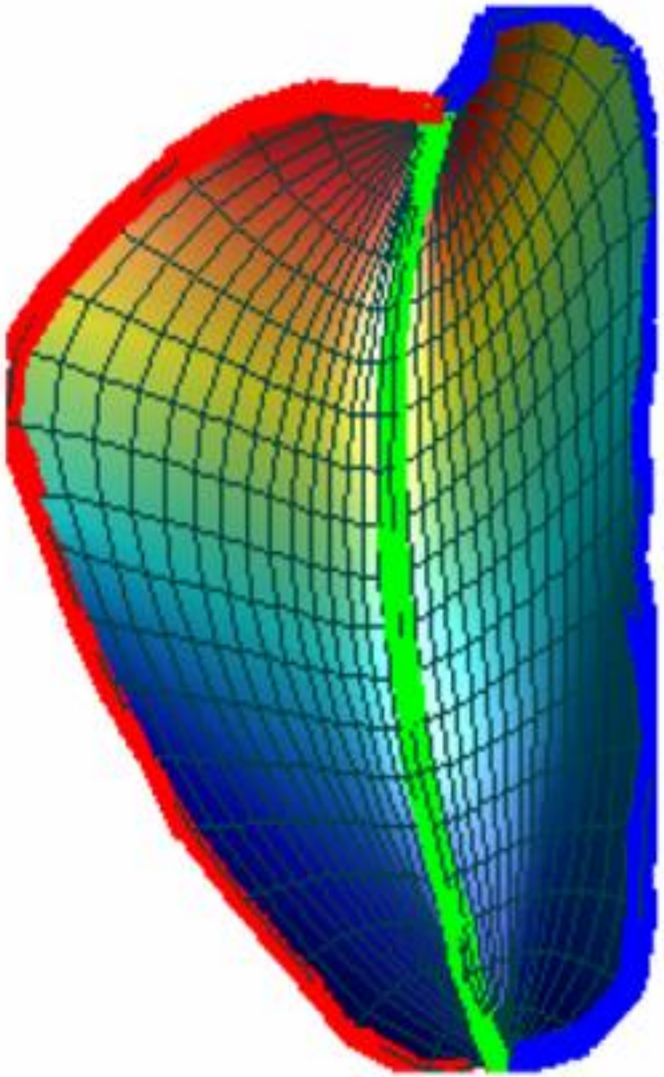

3D Representation

3D Model:

$\mathcal{X} = \; U \times V \times 3$ array,

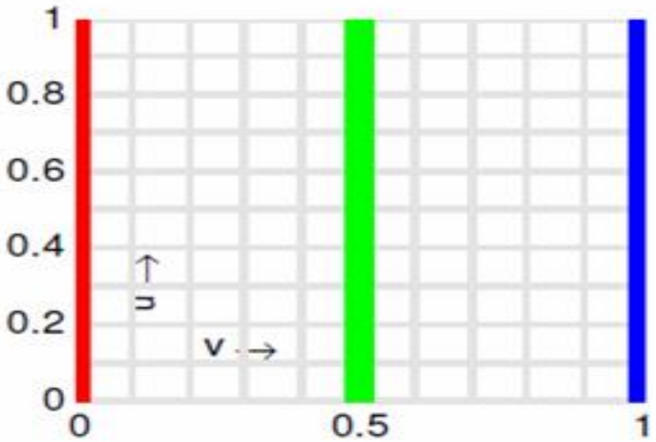
elements $\boldsymbol{X}_{uv} \in \mathbb{R}^3$

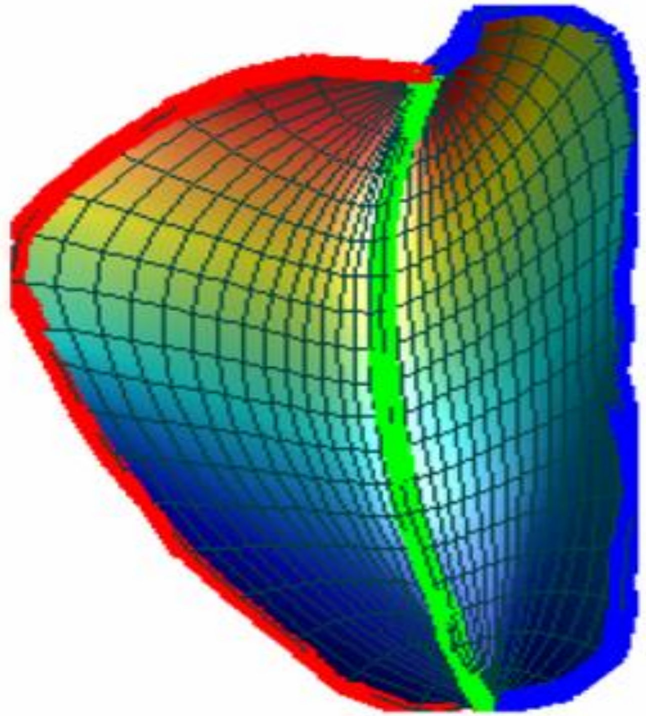




If we knew correspondences $\tilde{\boldsymbol{w}}_{nuv}$, we would solve missing data problem

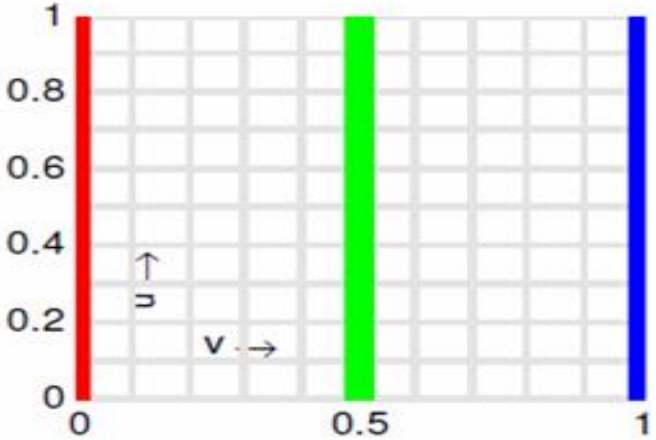
$$\min_{\substack{\alpha_{1..n} \\ B_{1..K} \\ P_{1..N}}} \sum_{\boldsymbol{n}} \sum_{\boldsymbol{u}} \sum_{\boldsymbol{v}} \phi_{nuv} \left\| \tilde{\boldsymbol{w}}_{nuv} - \pi(P_n, \sum_k \alpha_{nk} \boldsymbol{B}_{kuv}) \right\|$$

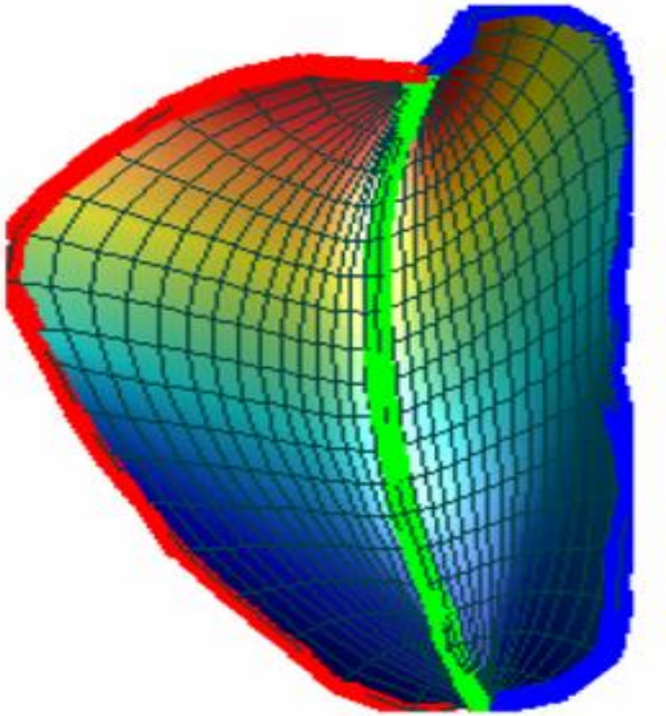




If we knew correspondences $\tilde{\boldsymbol{w}}_{nuv}$, we would solve missing data problem

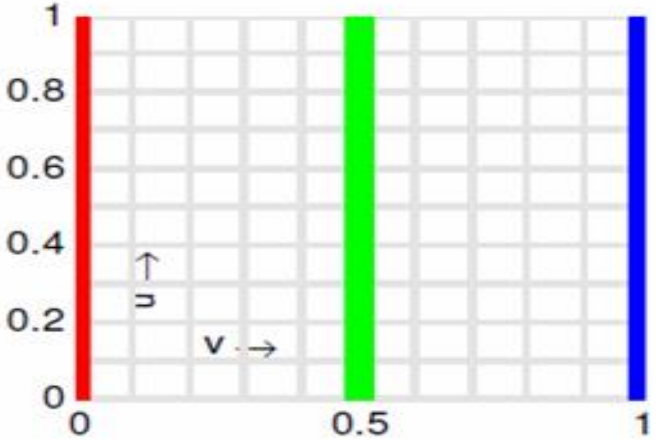
$$\min_{\theta} \sum_{\boldsymbol{n}} \sum_{\boldsymbol{u}} \sum_{\boldsymbol{v}} \phi_{nuv} \left\| \tilde{\boldsymbol{w}}_{nuv} - \boldsymbol{w}_{nuv}(\theta) \right\|$$

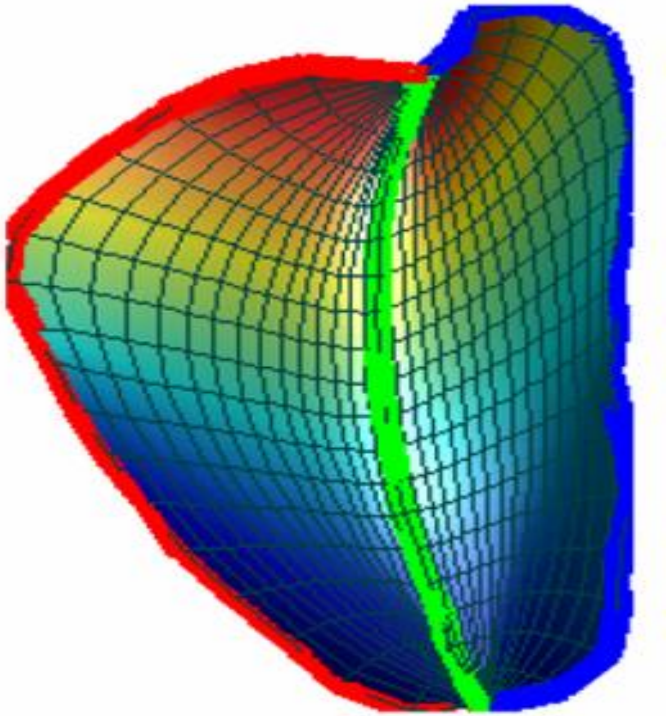




Without correspondences, image curve is $\tilde{\boldsymbol{w}}_{nu}(t)$, so solve

$$\min_{\theta} \sum_{\boldsymbol{n}} \sum_{\boldsymbol{u}} \sum_{\boldsymbol{v}} \phi_{nuv} \min_{t} \|\tilde{\boldsymbol{w}}_{nu}(t) - \boldsymbol{w}_{nuv}(\theta)\|$$





To solve this problem:

$$\min_{\theta} \sum_{\boldsymbol{n}} \sum_{\boldsymbol{u}} \sum_{\boldsymbol{v}} \phi_{nuv} \min_{t} \|\widetilde{\boldsymbol{w}}_{nu}(t) - \boldsymbol{w}_{nuv}(\theta)\|$$

Do this:

$$\min_{\substack{\theta \\ t_{1..NUV}}} \sum_{\boldsymbol{n}} \sum_{\boldsymbol{u}} \sum_{\boldsymbol{v}} \phi_{nuv} \|\widetilde{\boldsymbol{w}}_{nu}(t_{nuv}) - \boldsymbol{w}_{nuv}(\theta)\|$$

[Berthilsson & Kahl 01]

More simply

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} \min_{t} f_n(t, \theta)$$

More simply

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} \min_{t} f_n(t, \theta)$$

$$= \underset{\theta}{\operatorname{argmin}} \sum_{n} \min_{t_n} f_n(t_n, \theta)$$

More simply

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} \min_{t} f_n(t, \theta)$$

$$= \underset{\theta}{\operatorname{argmin}} \sum_{n} \min_{t_n} f_n(t_n, \theta)$$

$$= \underset{\theta}{\operatorname{argmin}} \min_{t_{1..N}} \sum_{n} f_n(t_n, \theta)$$

[Recall that: $\min_{x} f(x) + \min_{y} g(y) = \min_{x,y} f(x) + g(y)$]

More simply

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} \min_{t} f_n(t, \theta)$$

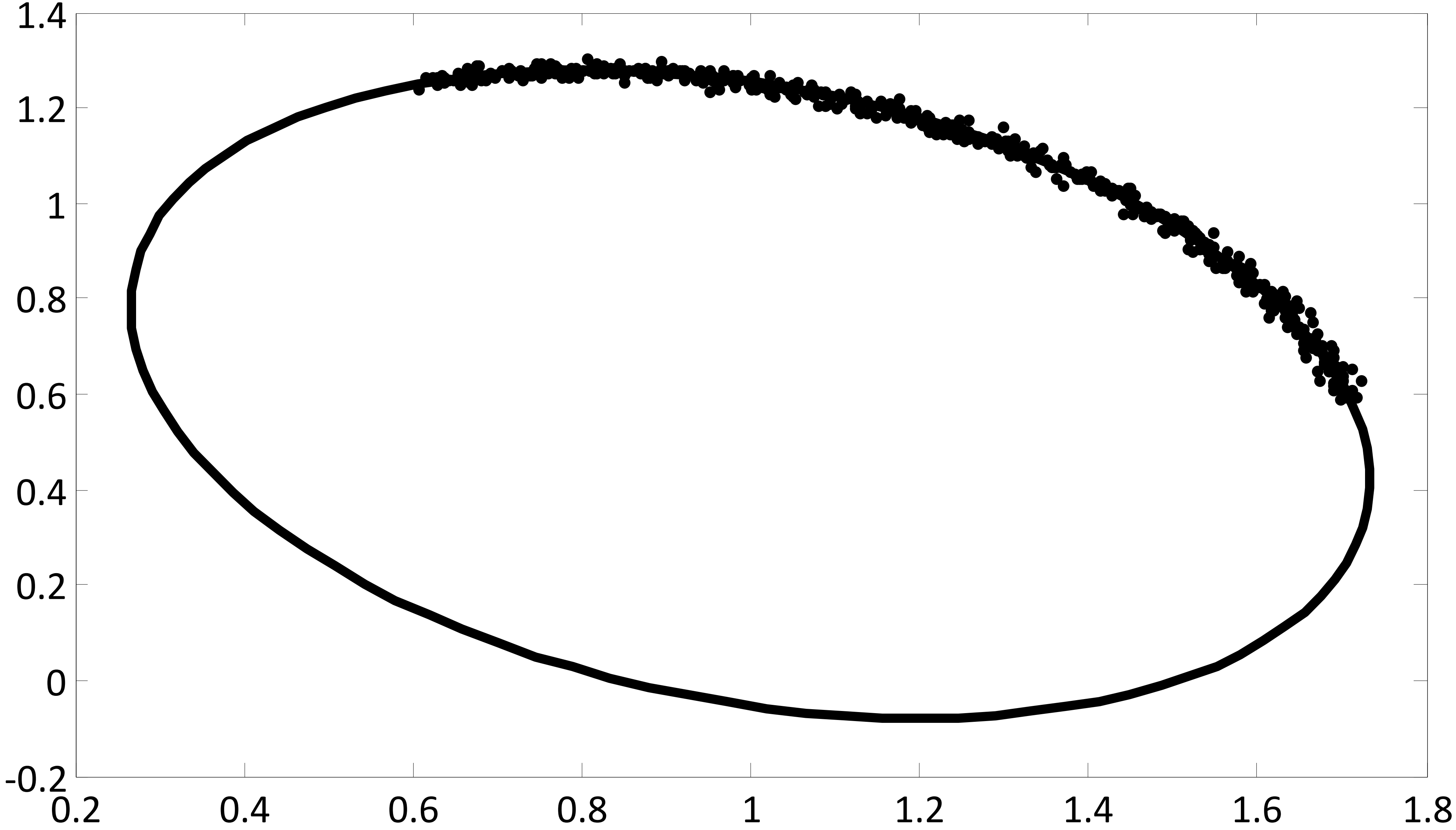
$$= \underset{\theta}{\operatorname{argmin}} \sum_{n} \min_{t_n} f_n(t_n, \theta)$$

So solve

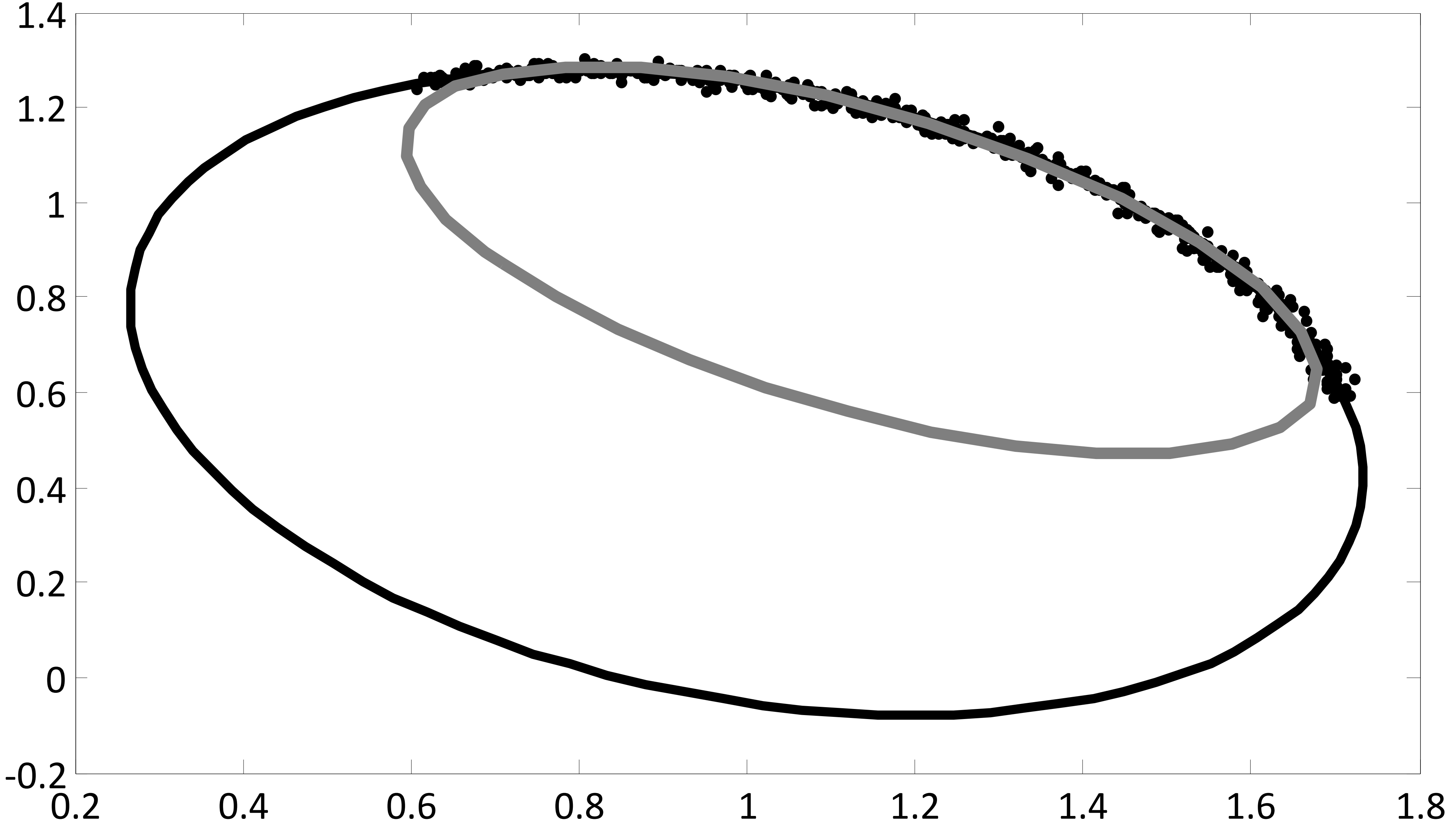
$$\min_{\theta, t_1, \ldots, t_N} \sum_{n=1}^{N} f_n(t_n, \theta)$$

And throw away the t's

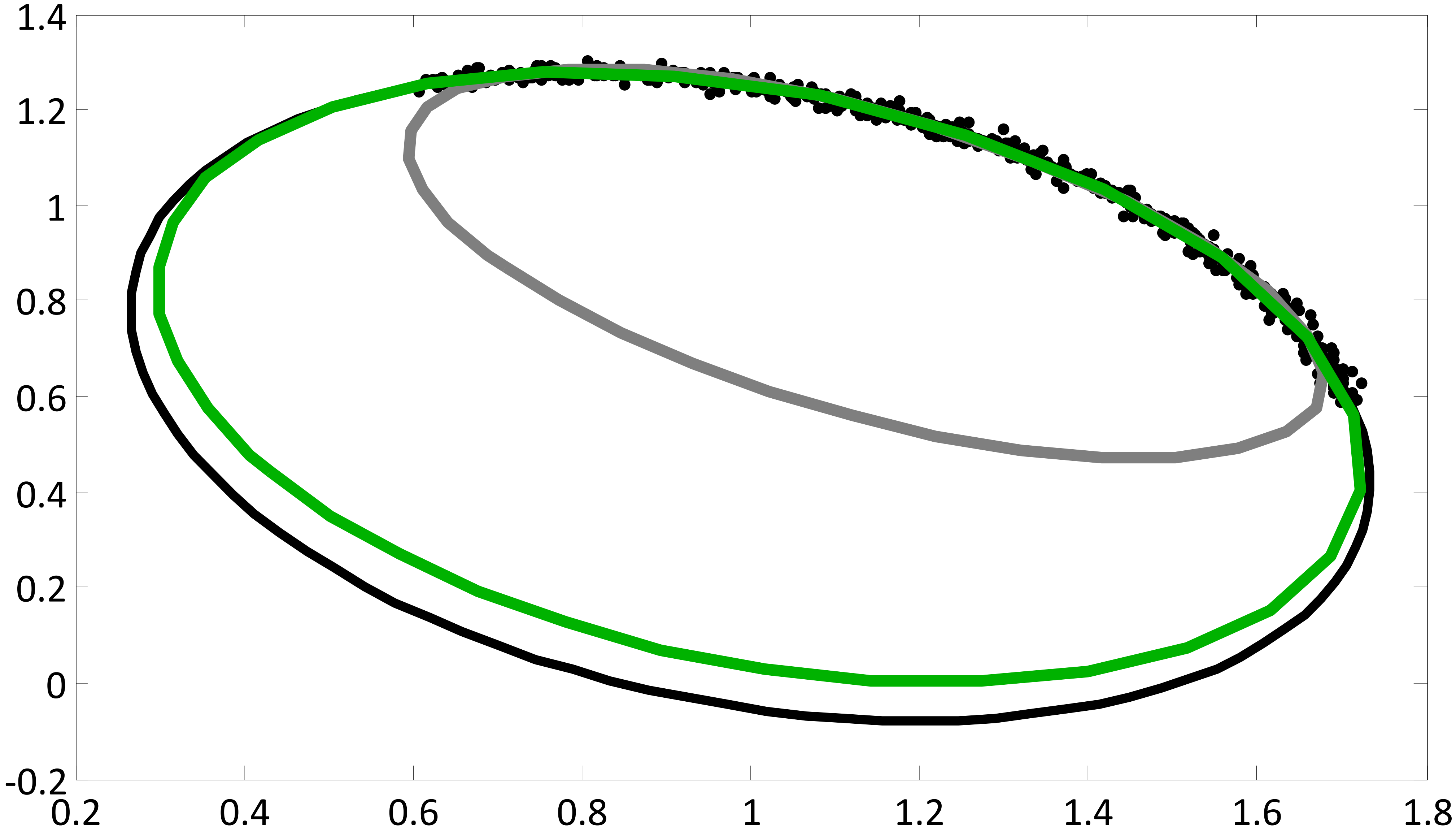
An old favourite



“Closed form” solution…

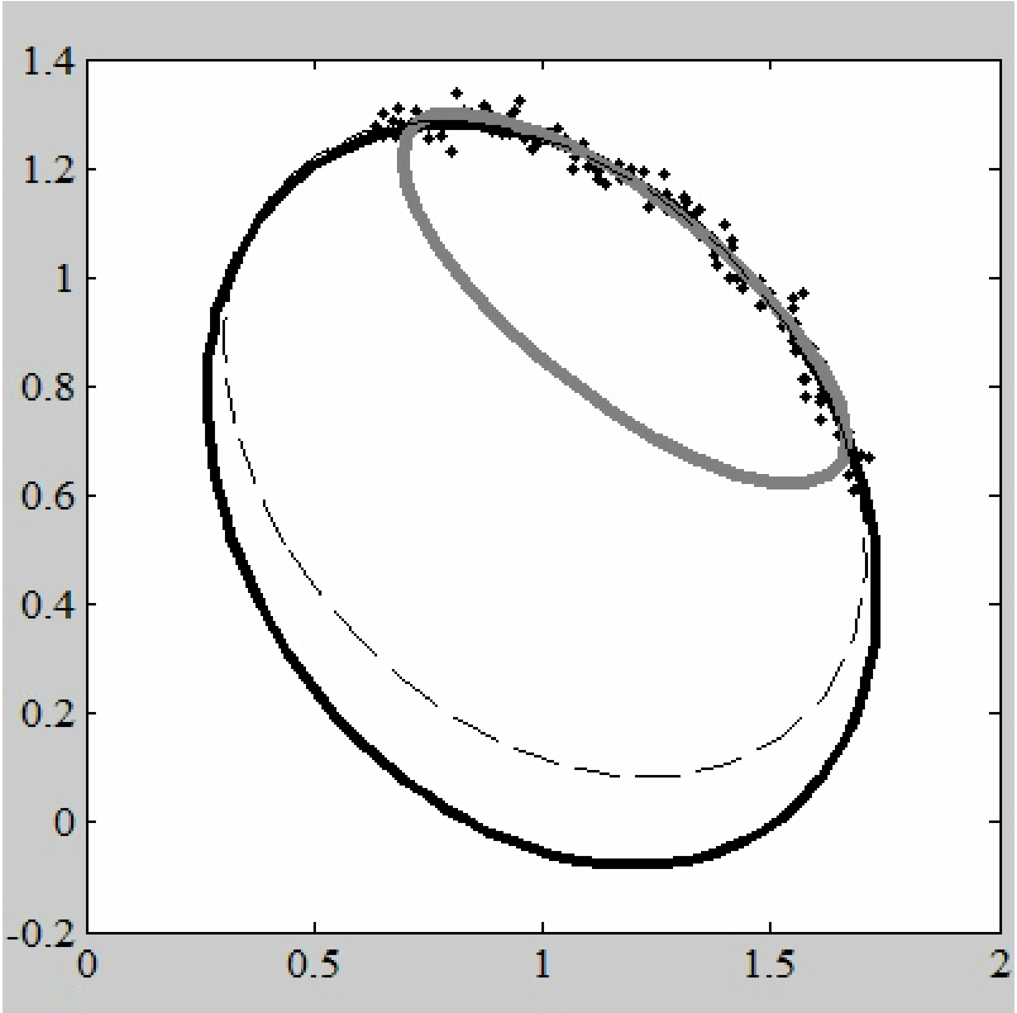


“Gold standard” solution…



[Gander, Golub, Strebel, BIT 34(1994)]

Attempt 1: alternate t and θ

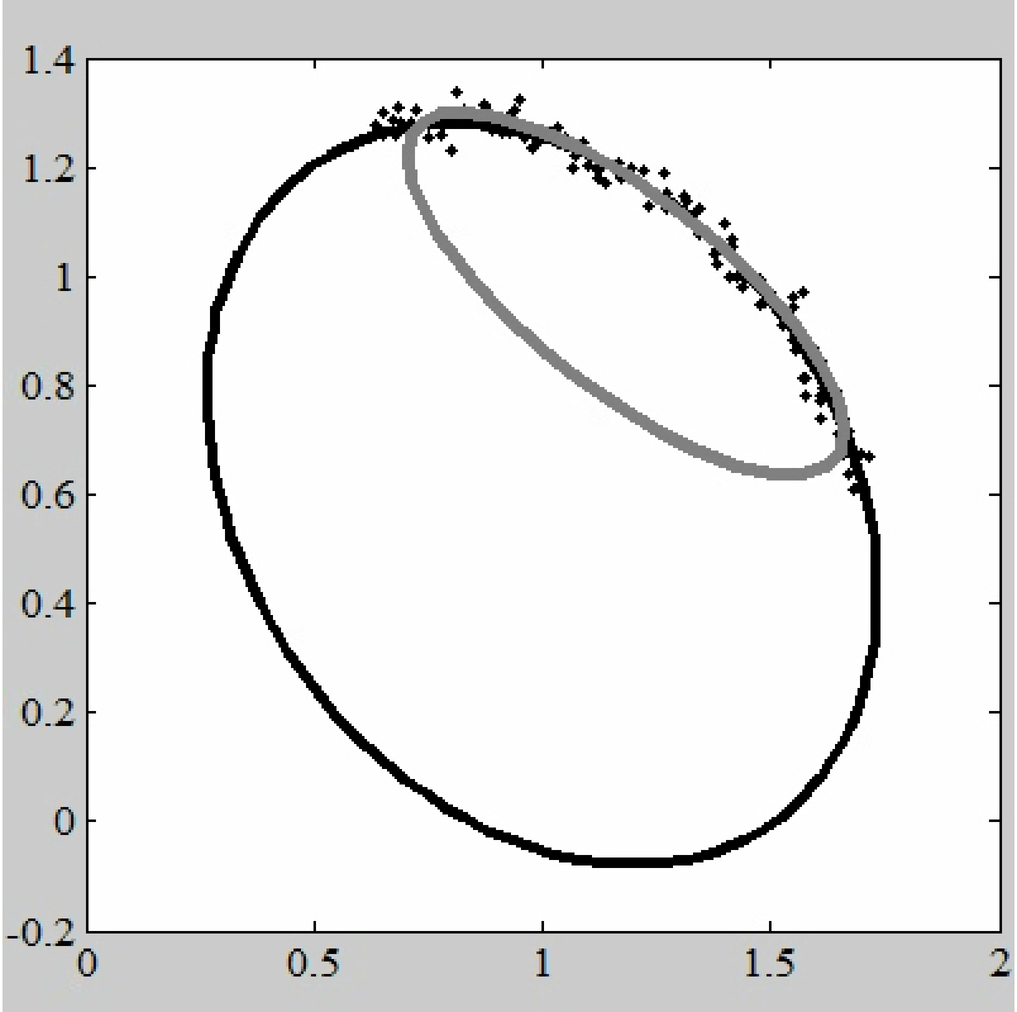


$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} \min_{t} f_n(t, \theta)$$

$$= \underset{\theta}{\operatorname{argmin}} \sum_{n} \min_{t_n} f_n(t_n, \theta)$$

1. Fix θ, find all t_n
2. Fix t_n, find θ

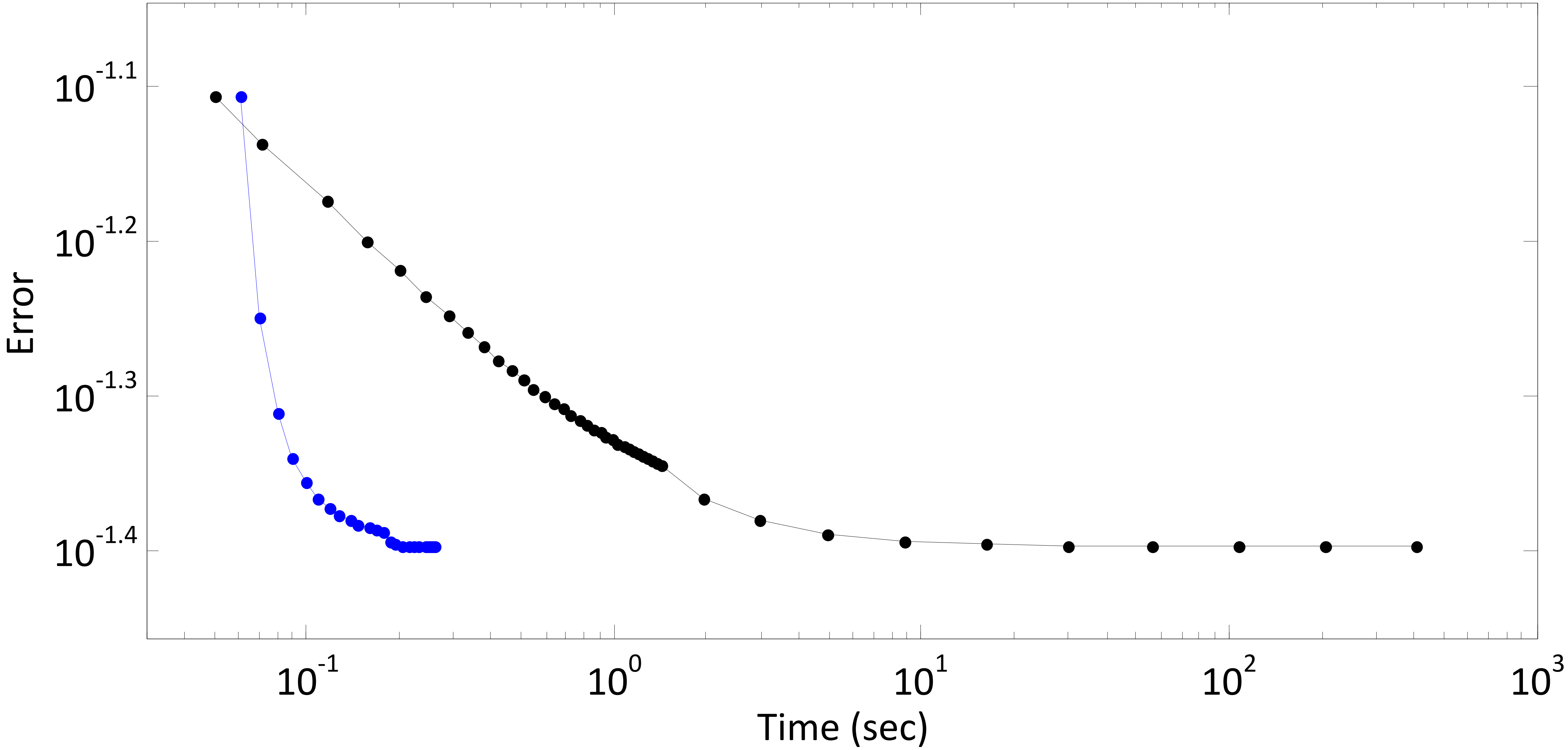
Attempt 2: All at once



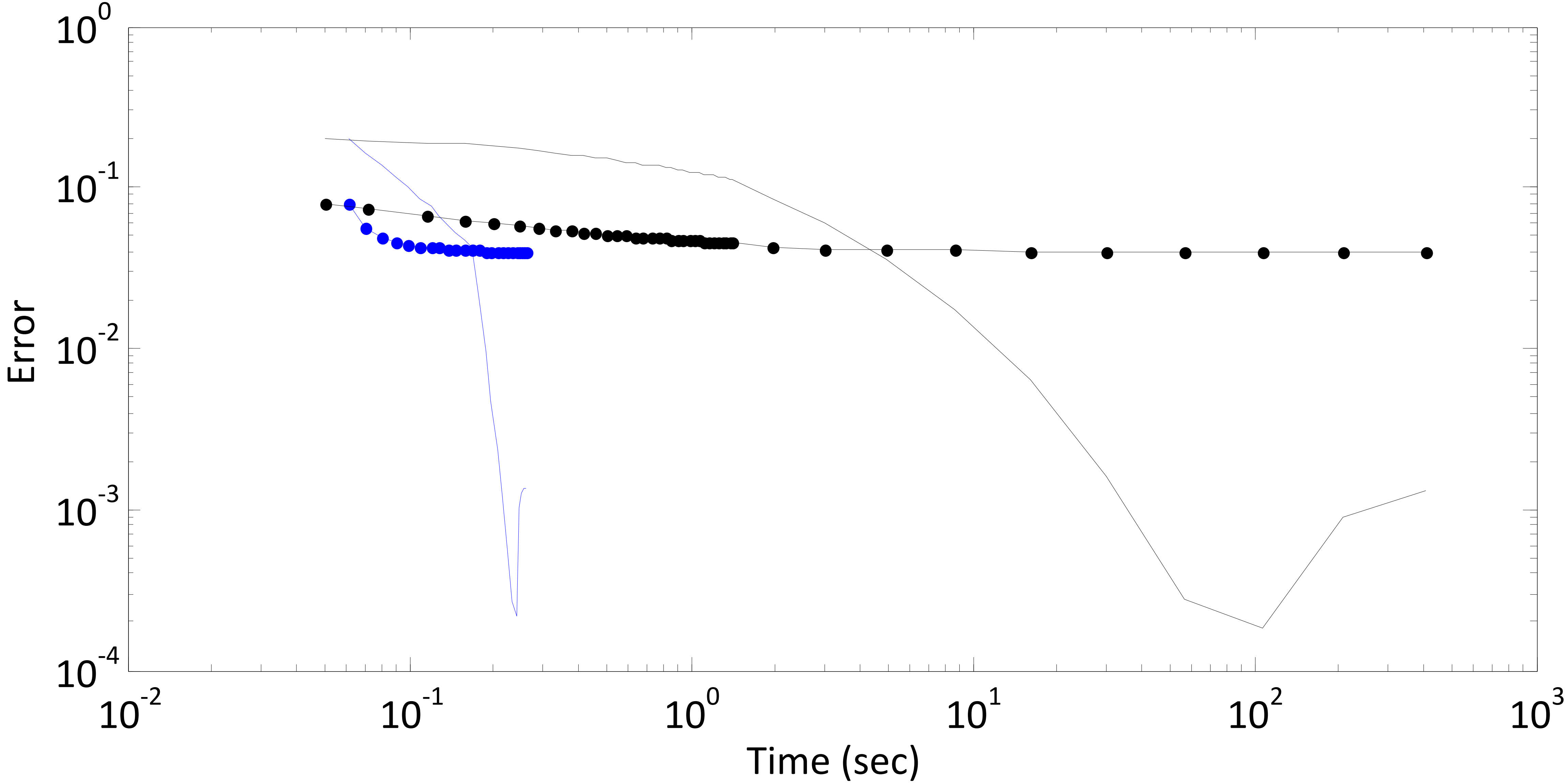
$$(\hat{\theta}, \sim) = \underset{\theta, t_1, \ldots, t_N}{\operatorname{argmin}} \sum_{n=1}^{N} f_n(t_n, \theta)$$

1. Call `lsqnonlin`
2. Throw away ts

Convergence curves, one instance



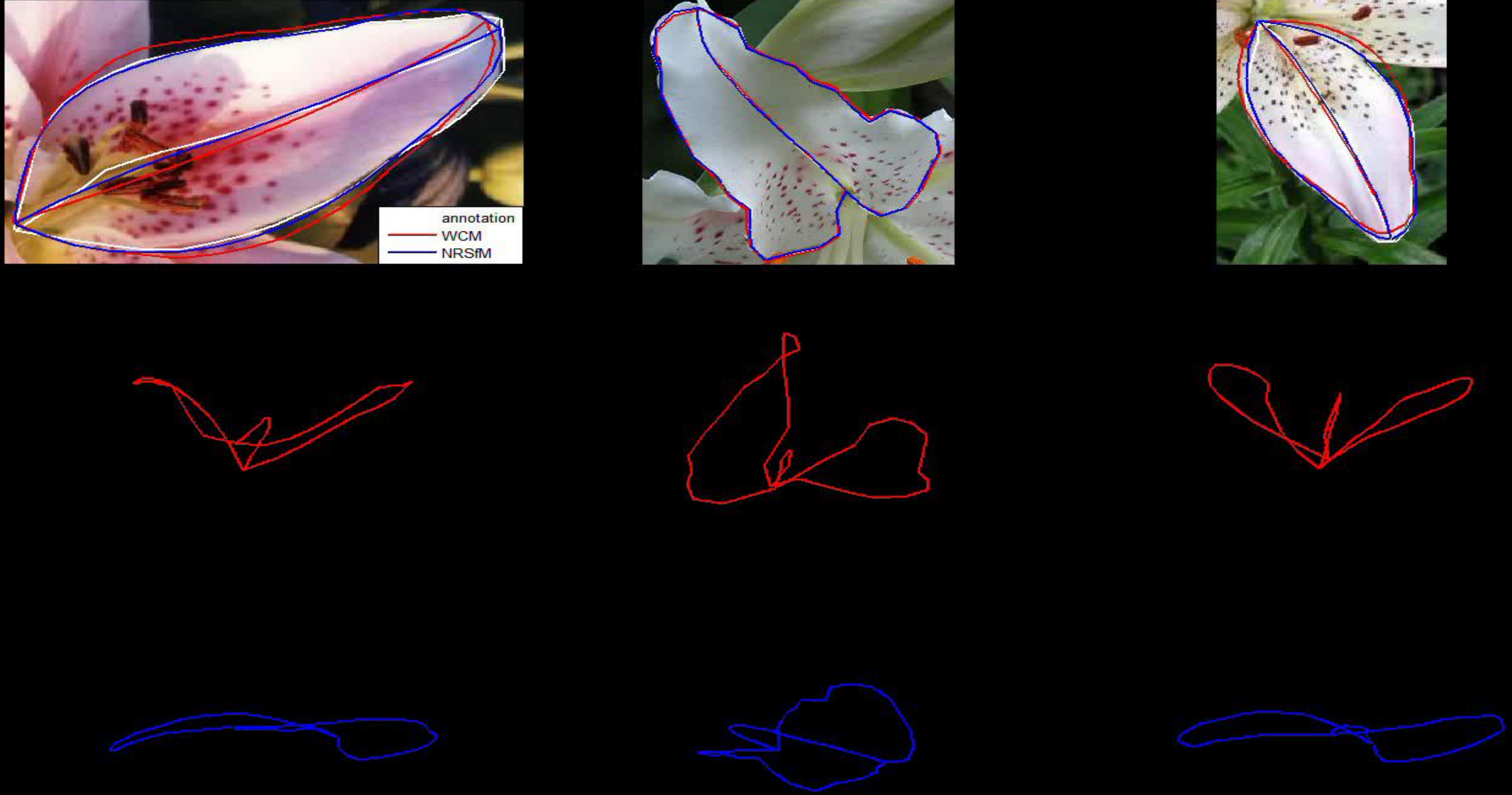
Convergence curves, one instance



Training images

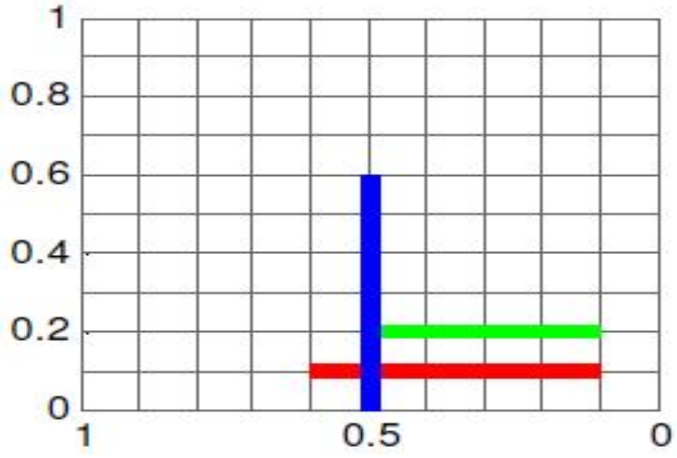


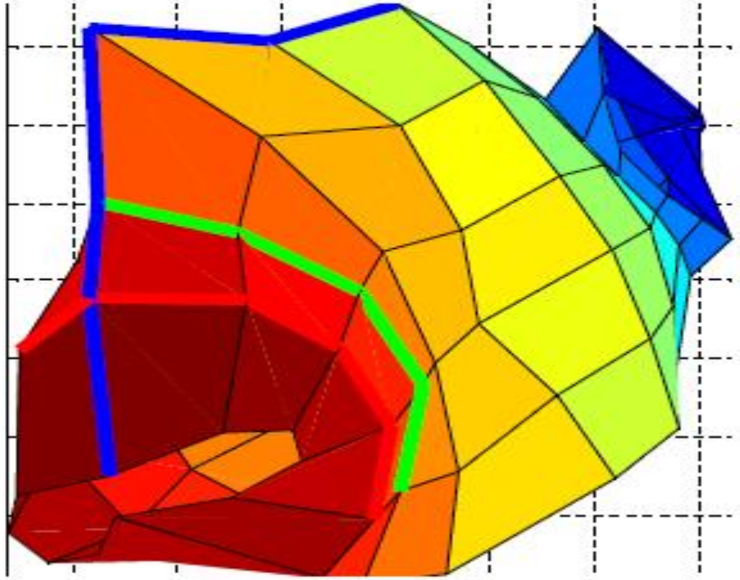


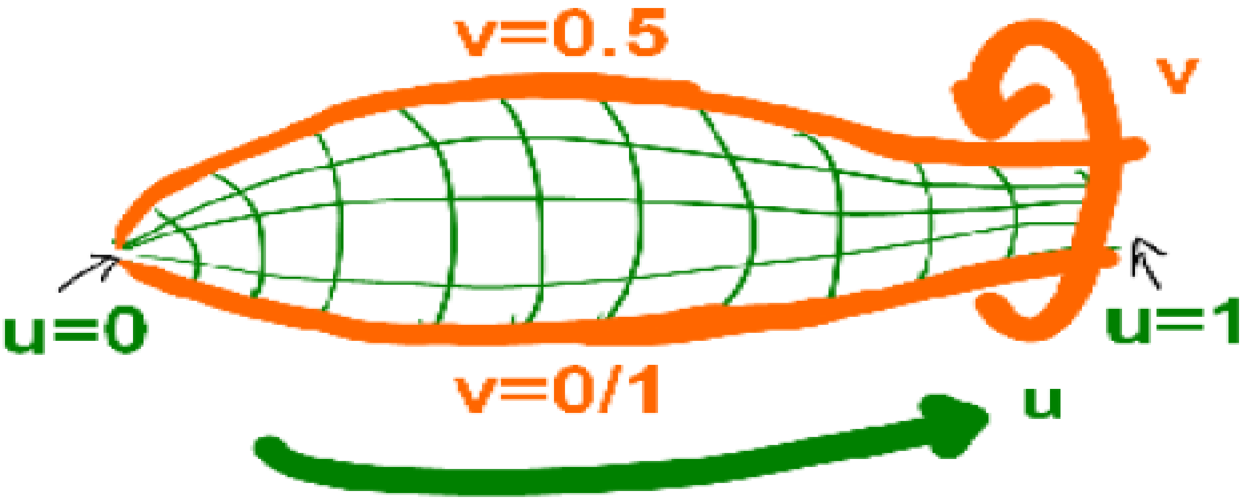


annotation
WCM
NRSfM

Training images

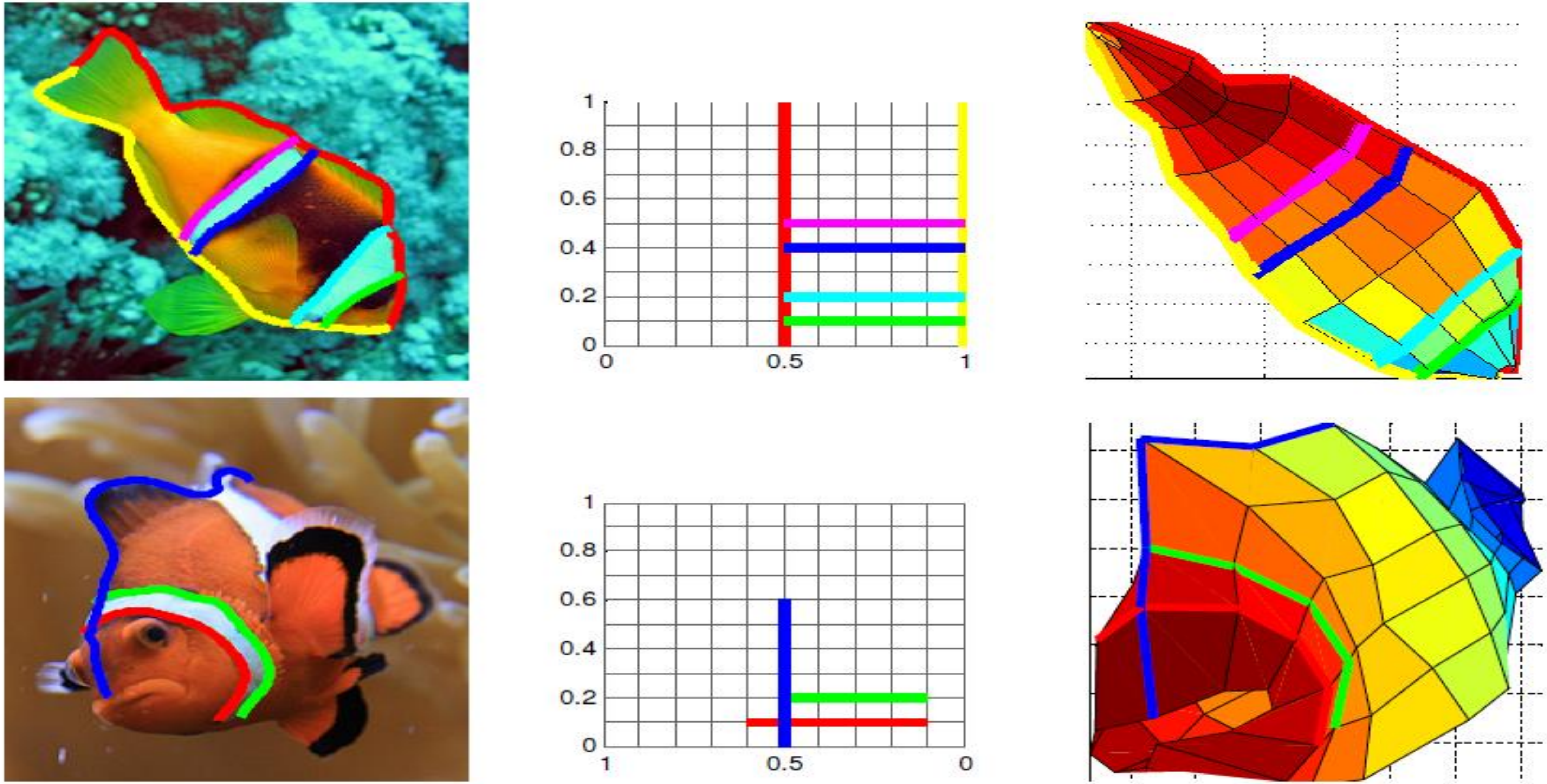
Partial occlusion

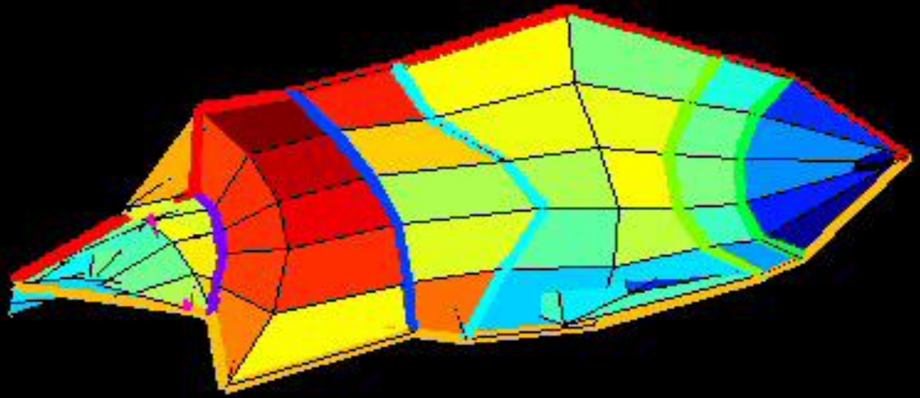


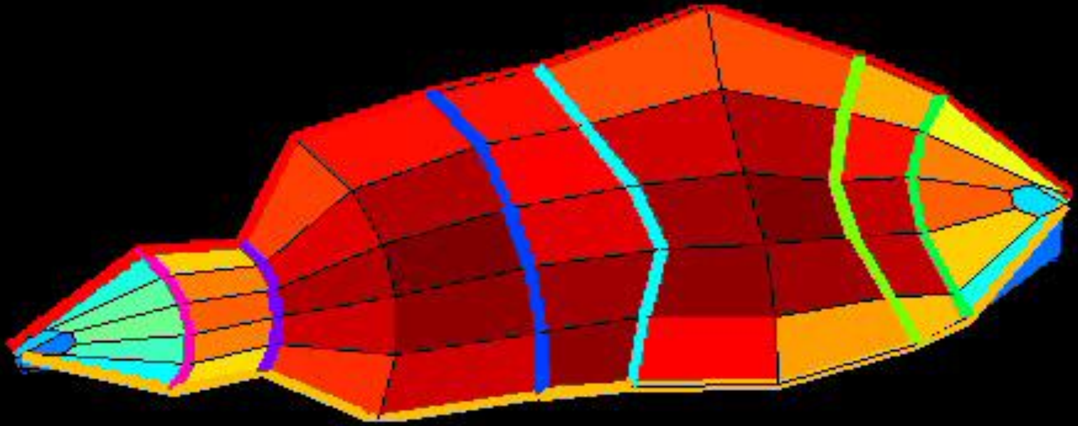




Partial occlusion

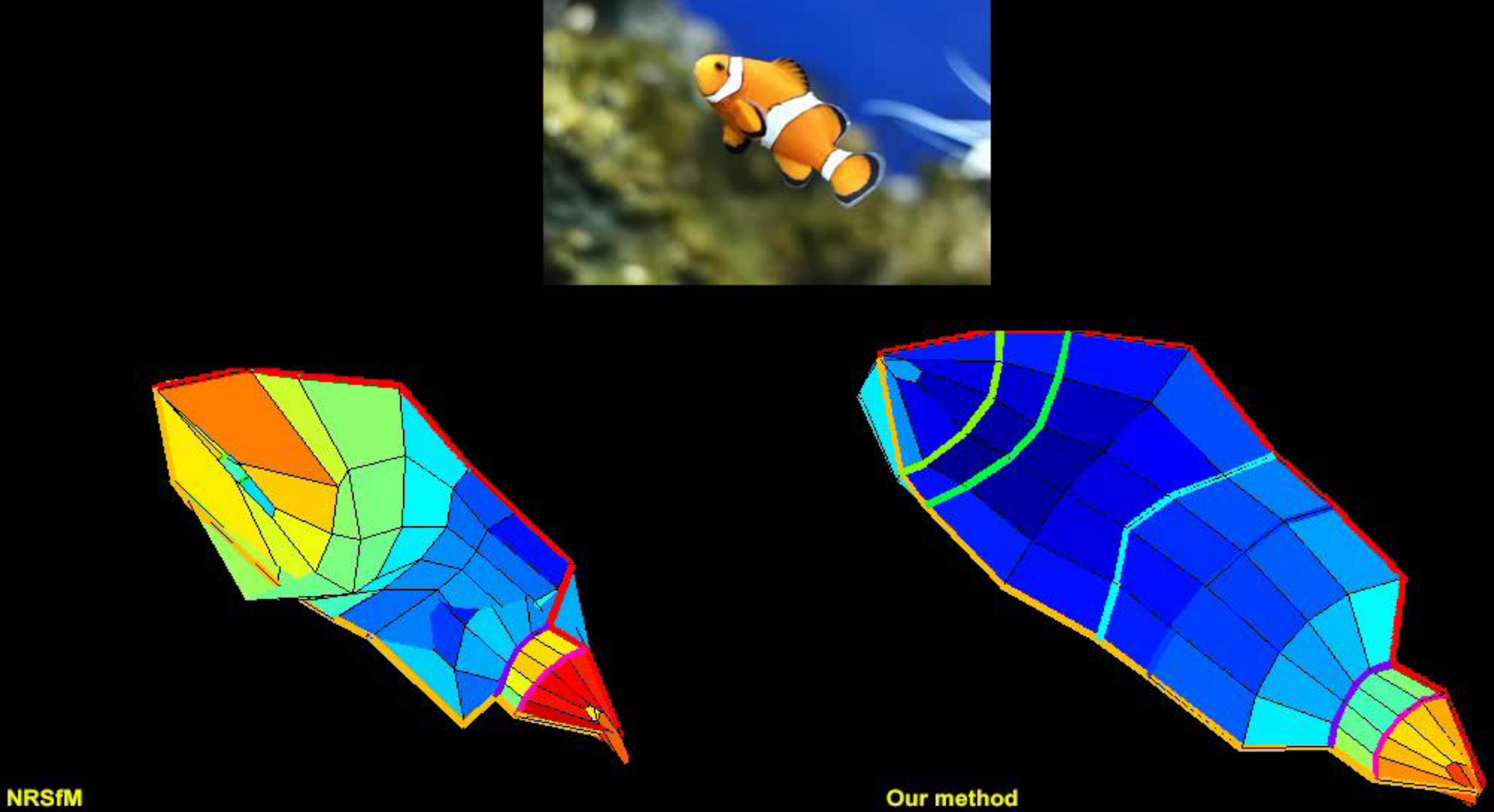


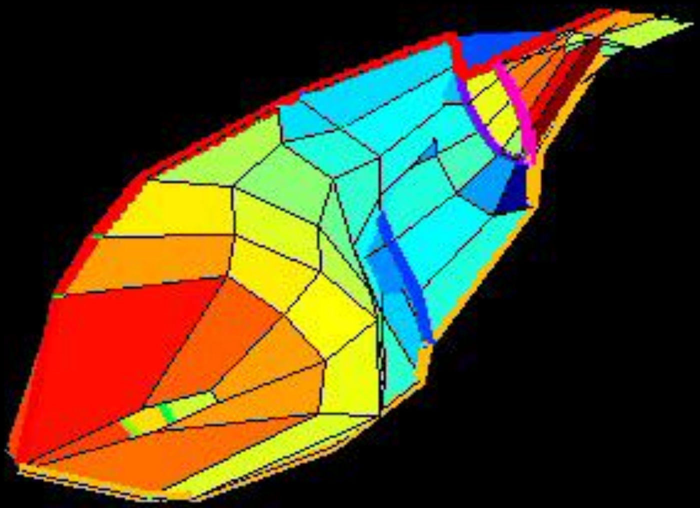




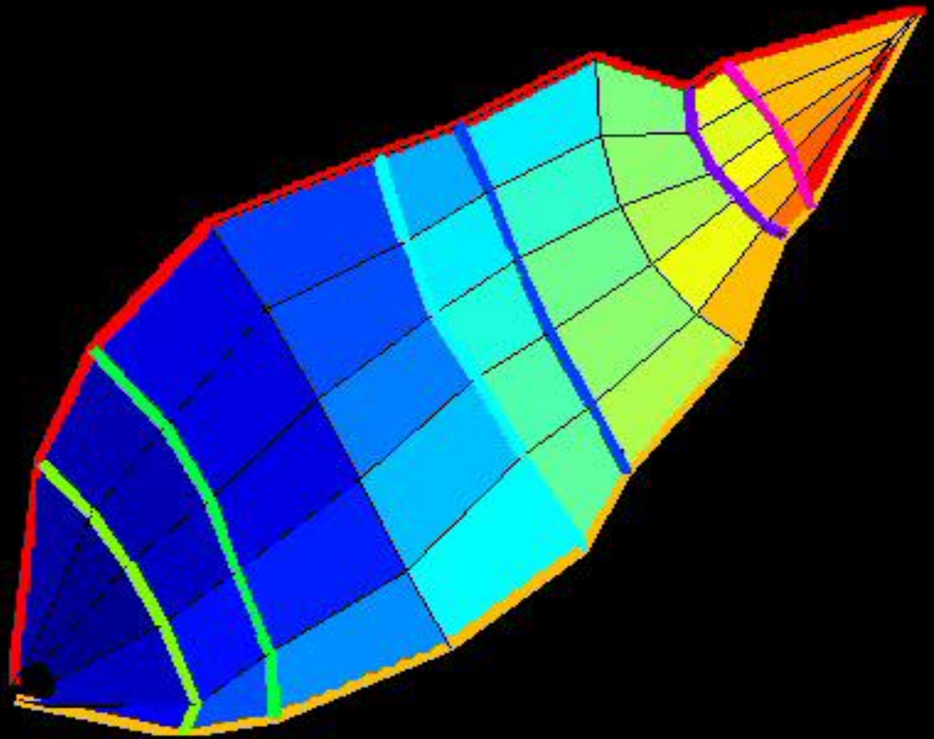
NRSfM

Our method

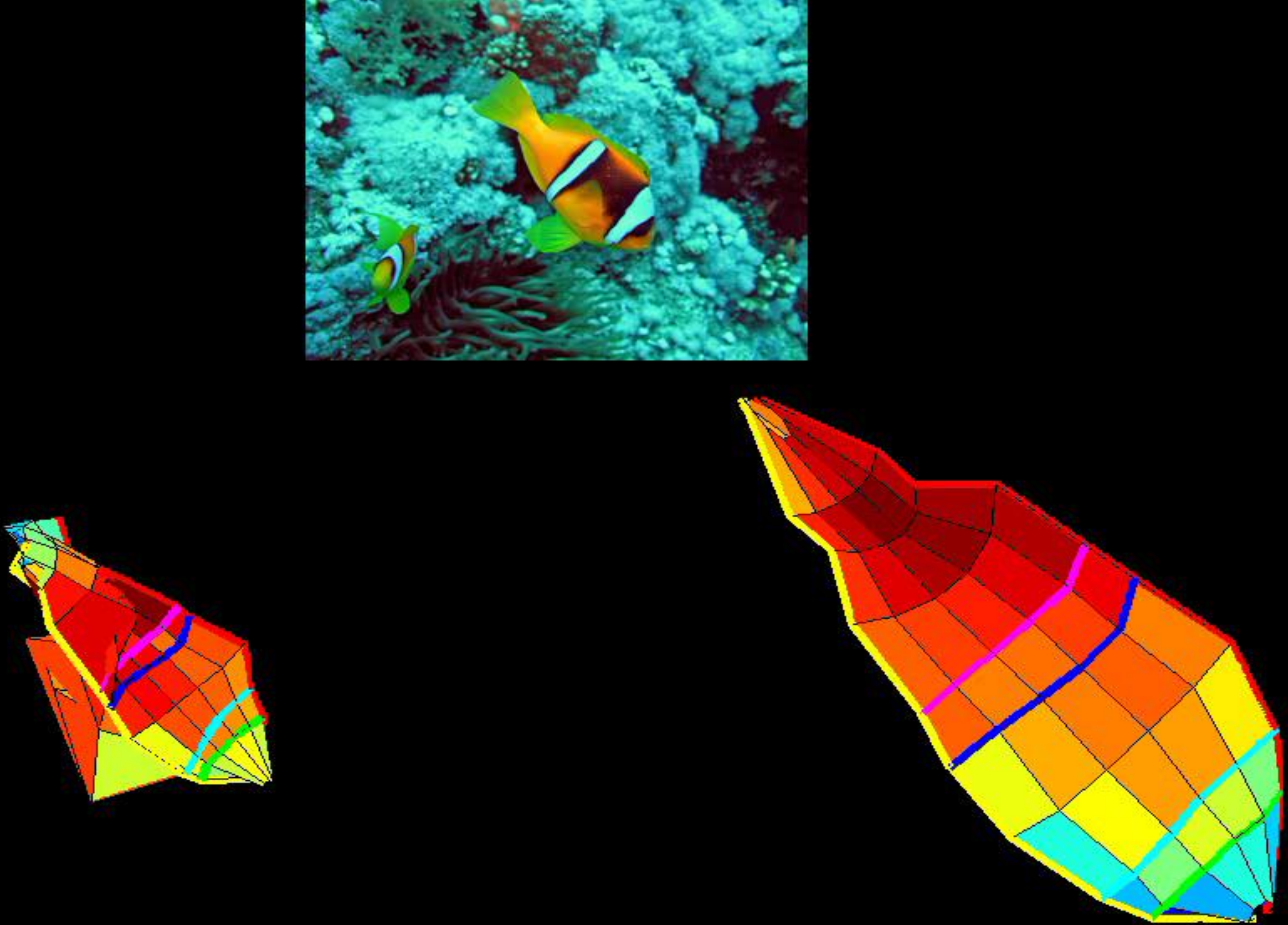

NRSfM
Our method



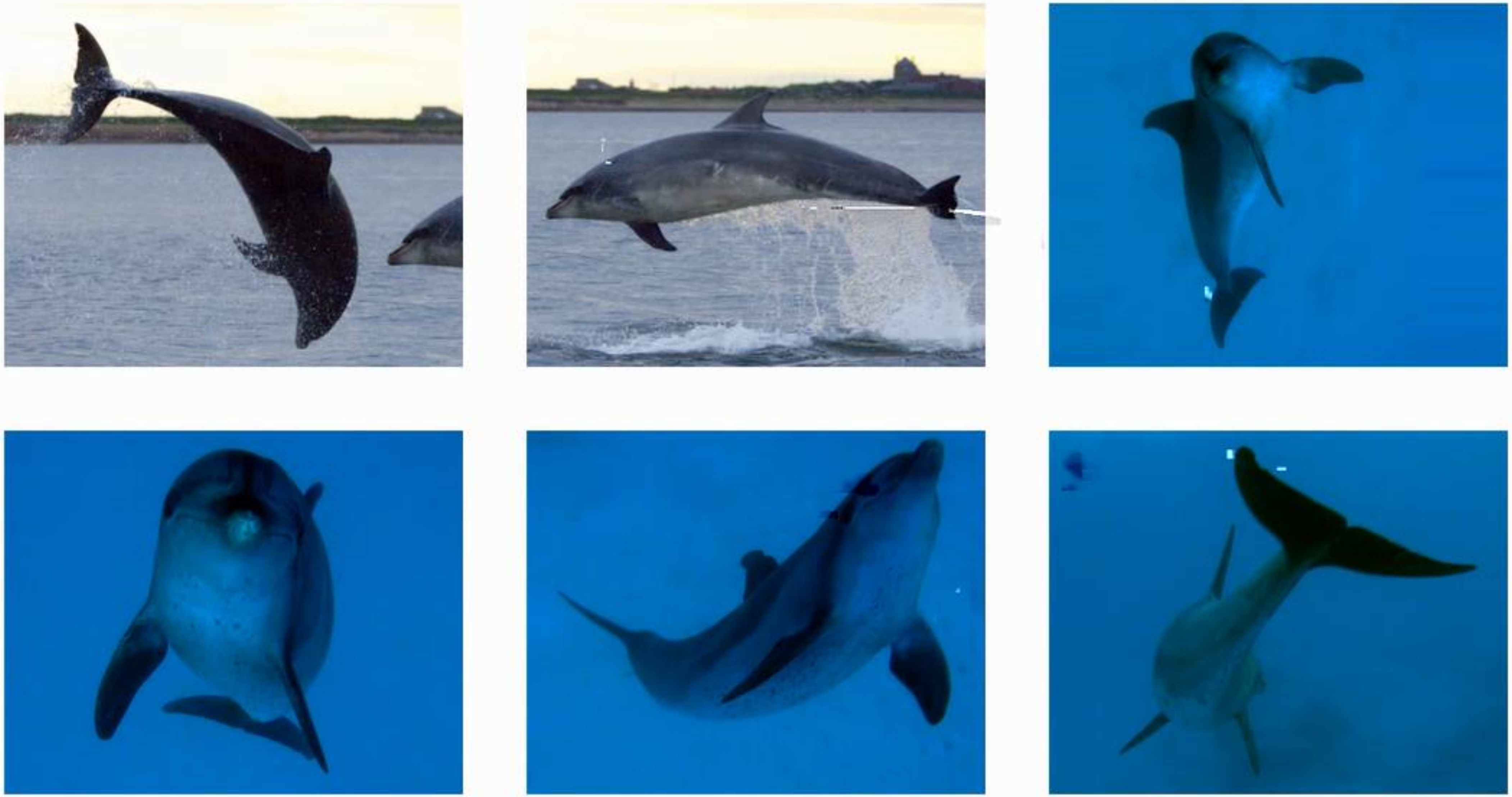
NRSfM



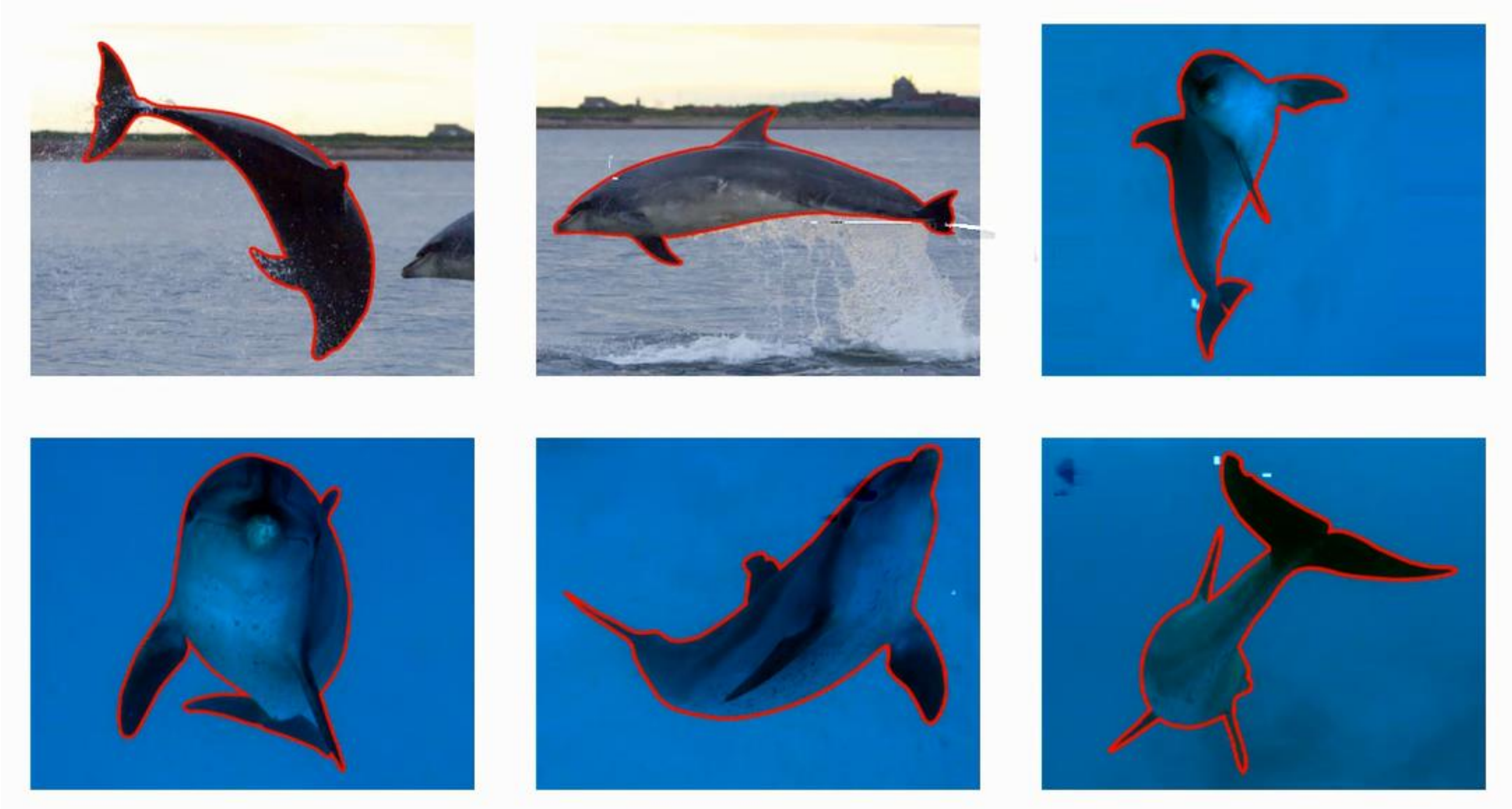
Our method



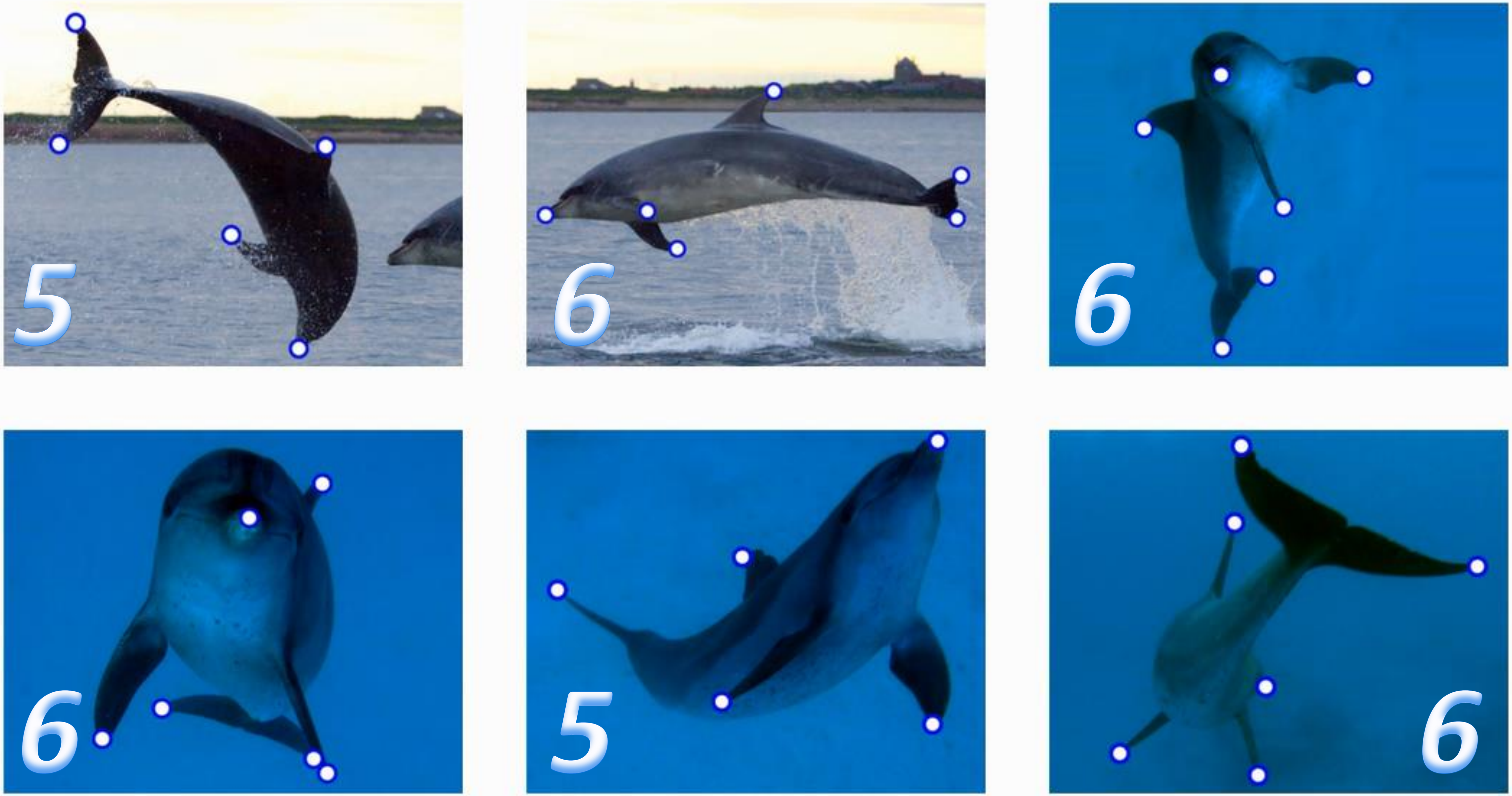
Back to dolphins: Input images



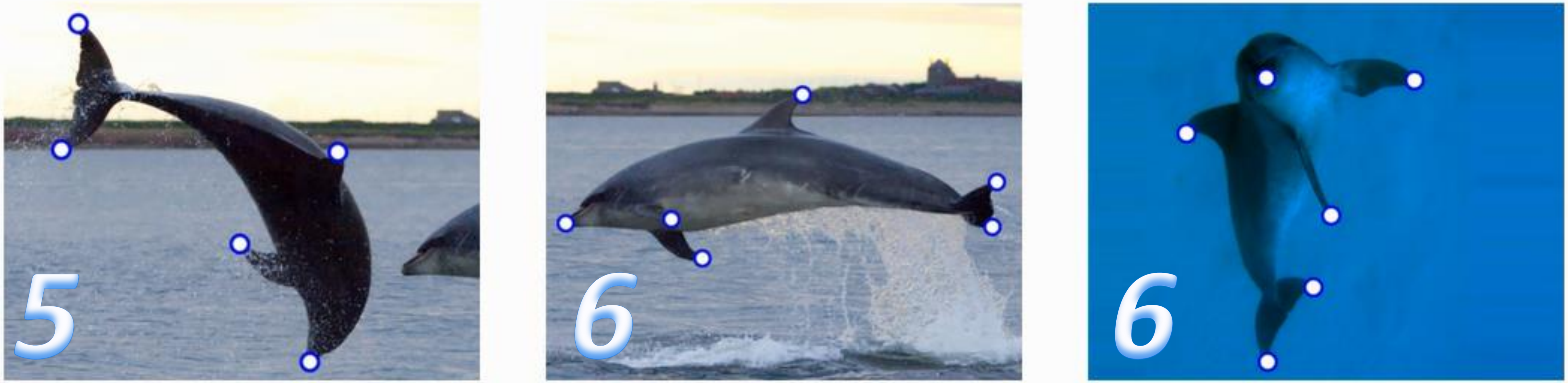
Input 1: Segmentation



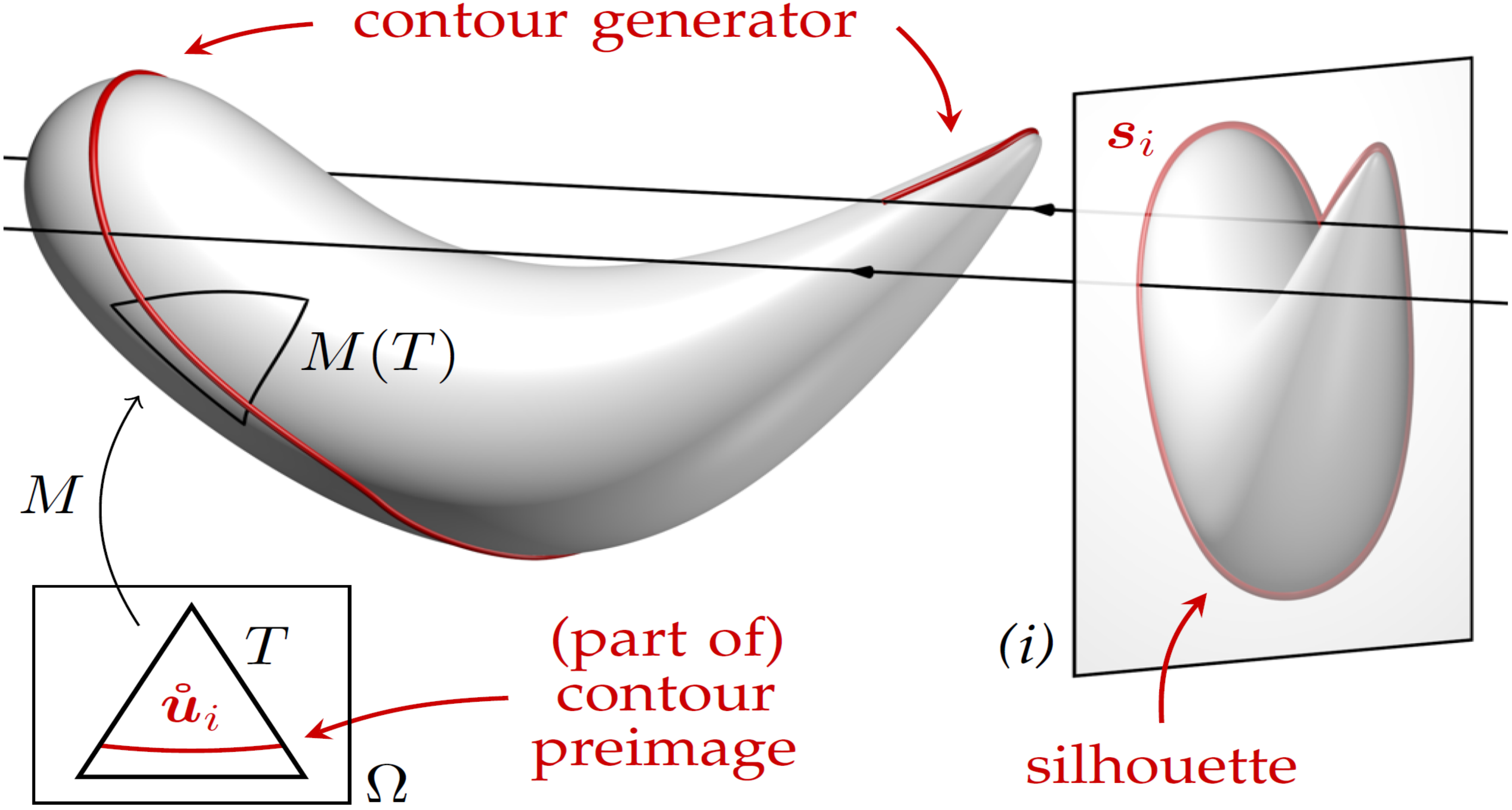
Input 2: Keypoints (if available)



Input 2: Keypoints (if available)



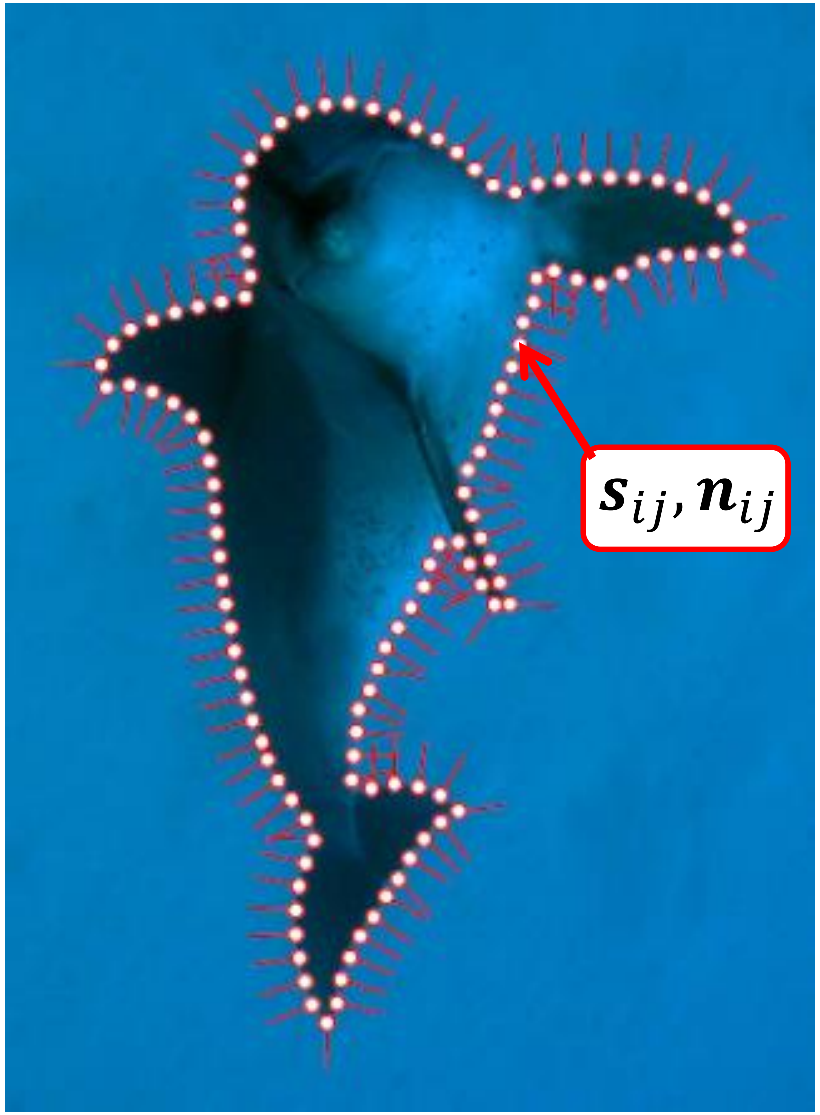
- Far too few points for nonrigid SfM
- Not all points selected in each image
- Could in principle be learned



contour generator
$\boldsymbol{s}_i$
$M(T)$
M
T
$\mathring{\boldsymbol{u}}_i$
Ω
(part of)
contour
preimage
(i)
silhouette

Data terms

Image i

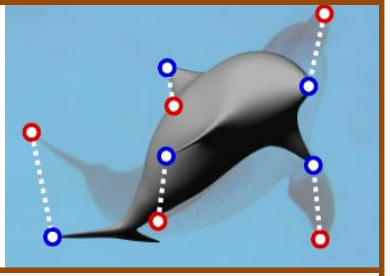
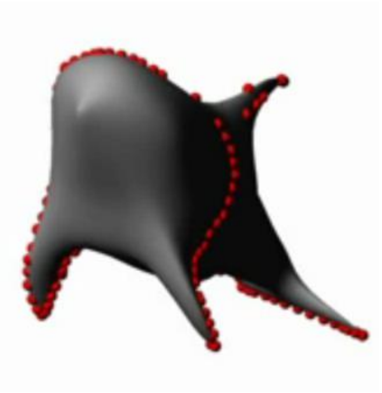


Silhouette:

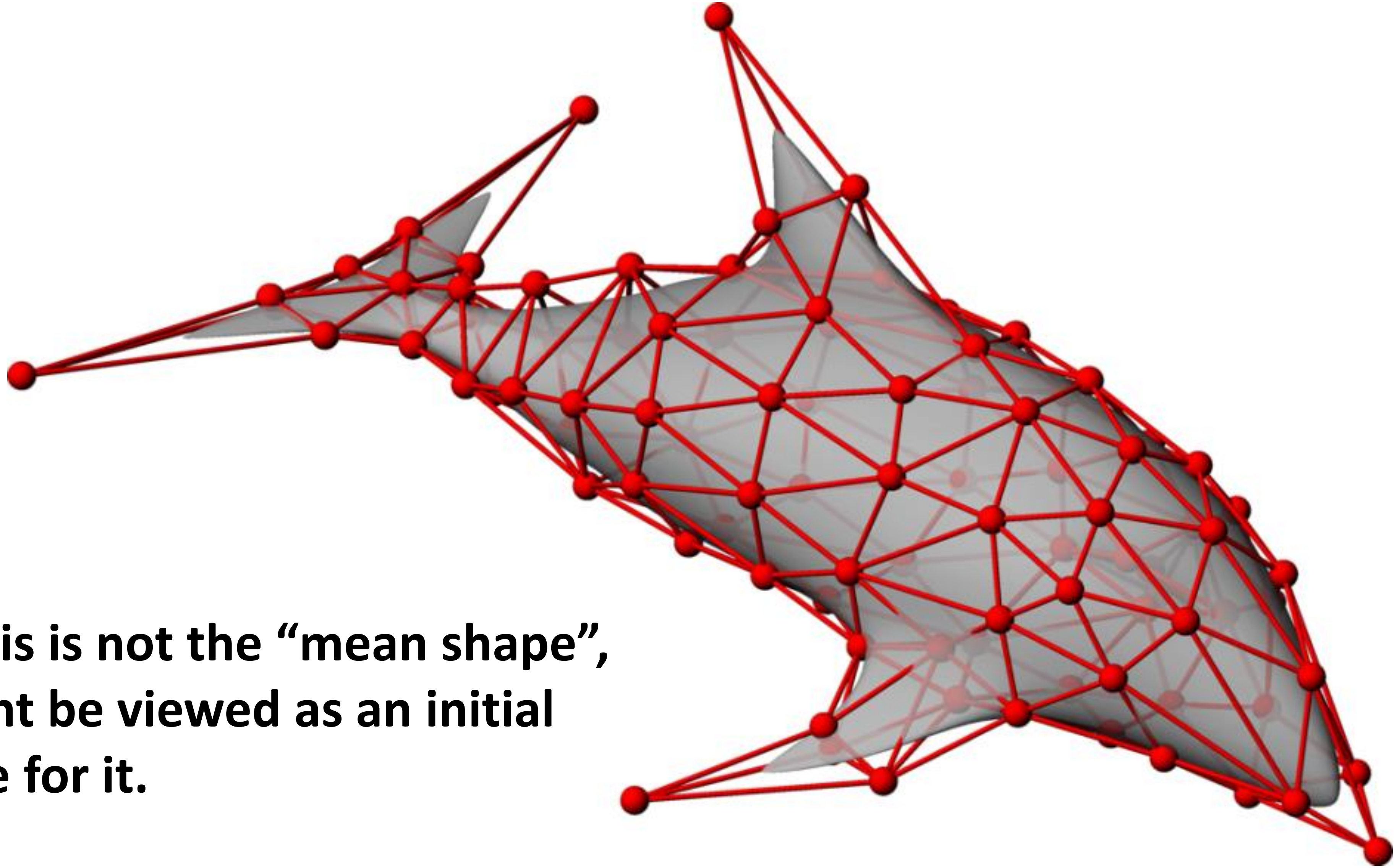
$$E_i^{\text{sil}} = \frac{1}{2}\sigma_{\text{sil}}^{-2} \sum_{j=1}^{S_i} \left\| s_{ij} - \pi_i \left(M(\mathring{u}_{ij} | \boldsymbol{X}_i) \right) \right\|^2$$

Normal:

$$E_i^{\text{norm}} = \frac{1}{2}\sigma_{\text{norm}}^{-2} \sum_{j=1}^{S_i} \left\| \begin{bmatrix} n_{ij} \\ 0 \end{bmatrix} - \nu \left(\boldsymbol{R}_i N(\mathring{u}_{ij} | \boldsymbol{X}_i) \right) \right\|^2$$

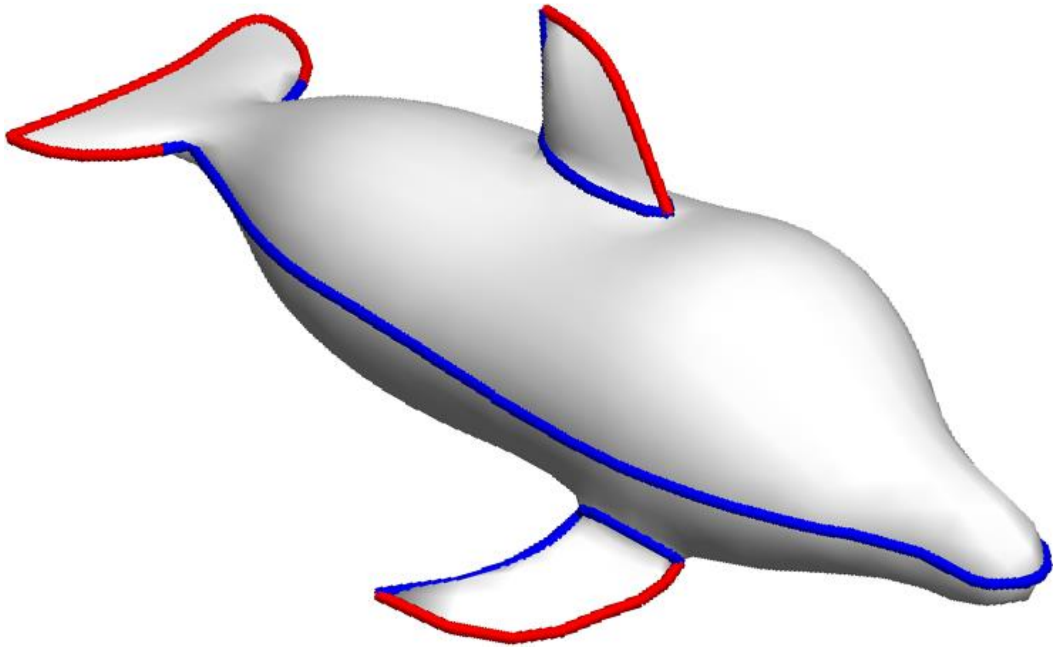
Data fidelity terms	$E_i^{\text{sil}} = \frac{1}{2}\sigma_{\text{sil}}^{-2} \sum_{j=1}^{S_i} \left\| \boldsymbol{s}_{ij} - \pi_i\left(M(\mathring{u}_{ij} \mid \boldsymbol{X}_i)\right)\right\|^2$ $E_i^{\text{norm}} = \frac{1}{2}\sigma_{\text{norm}}^{-2} \sum_{j=1}^{S_i} \left\| \begin{bmatrix} \boldsymbol{n}_{ij} \\ 0 \end{bmatrix} - \nu\left(\boldsymbol{R}_i N(\mathring{u}_{ij} \mid \boldsymbol{X}_i)\right)\right\|^2$ 
	$E_i^{\text{con}} = \frac{1}{2}\sigma_{\text{con}}^{-2} \sum_{k=1}^{K_i} \left\| \boldsymbol{c}_{ik} - \pi_i\left(M(\mathring{\mu}_{ik} \mid \boldsymbol{X}_i)\right)\right\|^2$ 
Smoothing terms	$E_m^{\text{tp}} = \frac{\bar{\lambda}^2}{2} \int_\Omega \left\| M_{xx}(\mathring{u} \mid \boldsymbol{B}_m)\right\|^2 + 2\left\| M_{xy}(\mathring{u} \mid \boldsymbol{B}_m)\right\|^2 + \left\| M_{yy}(\mathring{u} \mid \boldsymbol{B}_m)\right\|^2 \, \mathrm{d}\mathring{u}$
"Technical" terms	$E_i^{\text{reg}} = \beta \sum_{m=1}^{D} \alpha_{im}^2$ $\quad \boldsymbol{X}_i = \sum_{m=0}^{D} \alpha_{im} \boldsymbol{B}_m$ $E_i^{\text{cg}} = \gamma \sum_{j=1}^{S_i} \tau\left(d(\mathring{u}_{ij}, \mathring{u}_{i,j+1})\right)$ 

Initialization : Rough dolphin model



Note: this is not the “mean shape”, but might be viewed as an initial estimate for it.

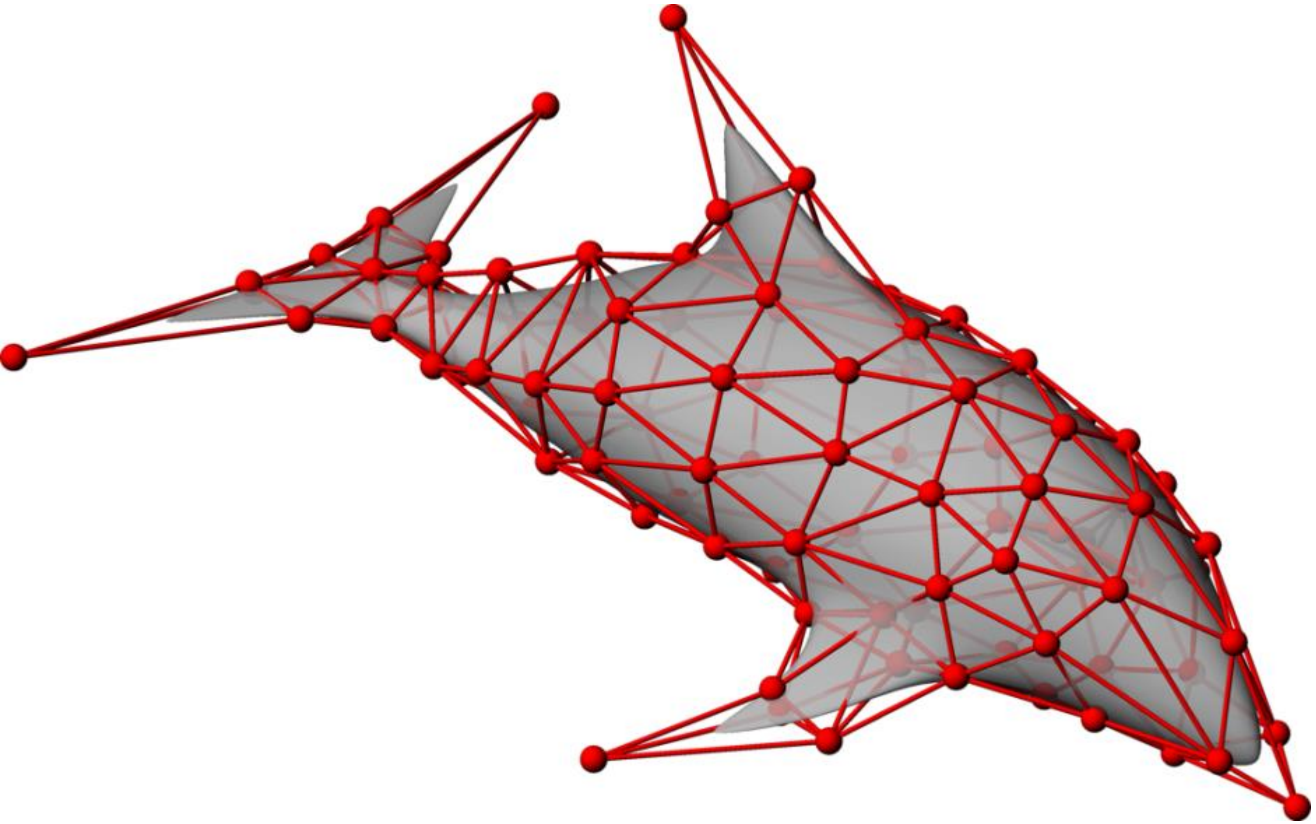
Initialization : Rough dolphin model



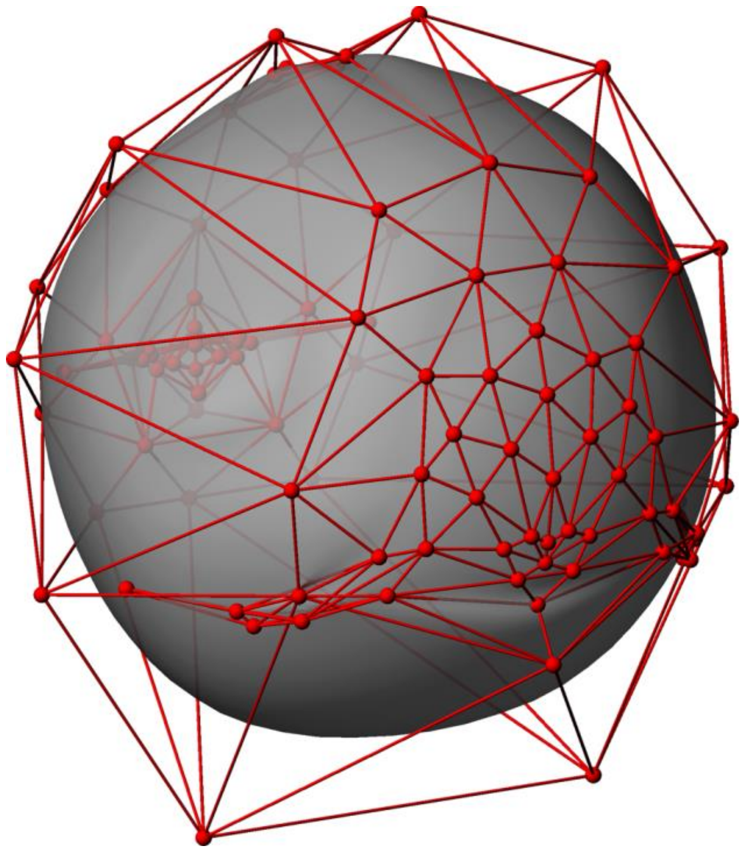
FiberMesh [Nealen et al]

Mesh model

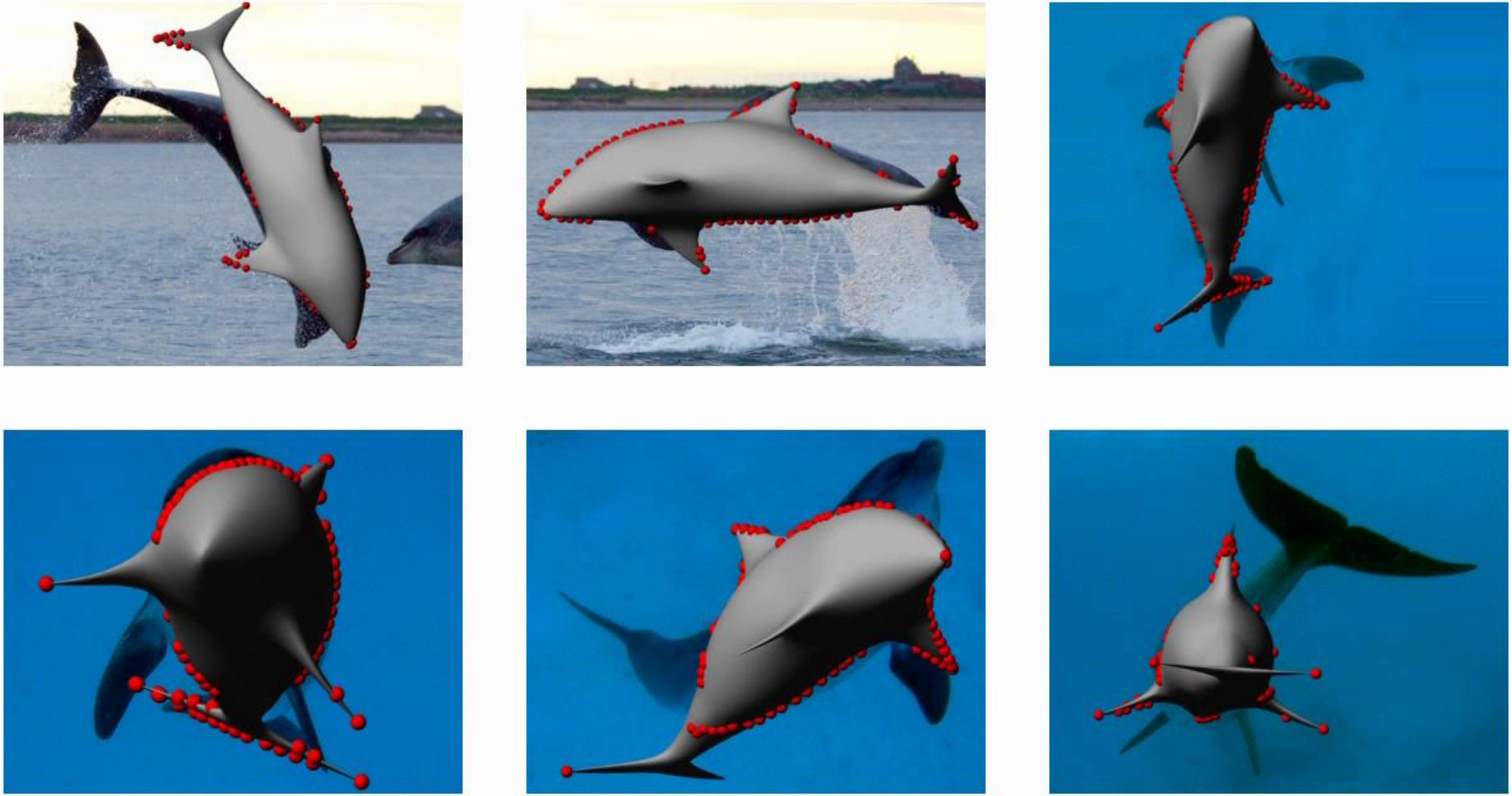
Initialization : Rough dolphin model



True template model

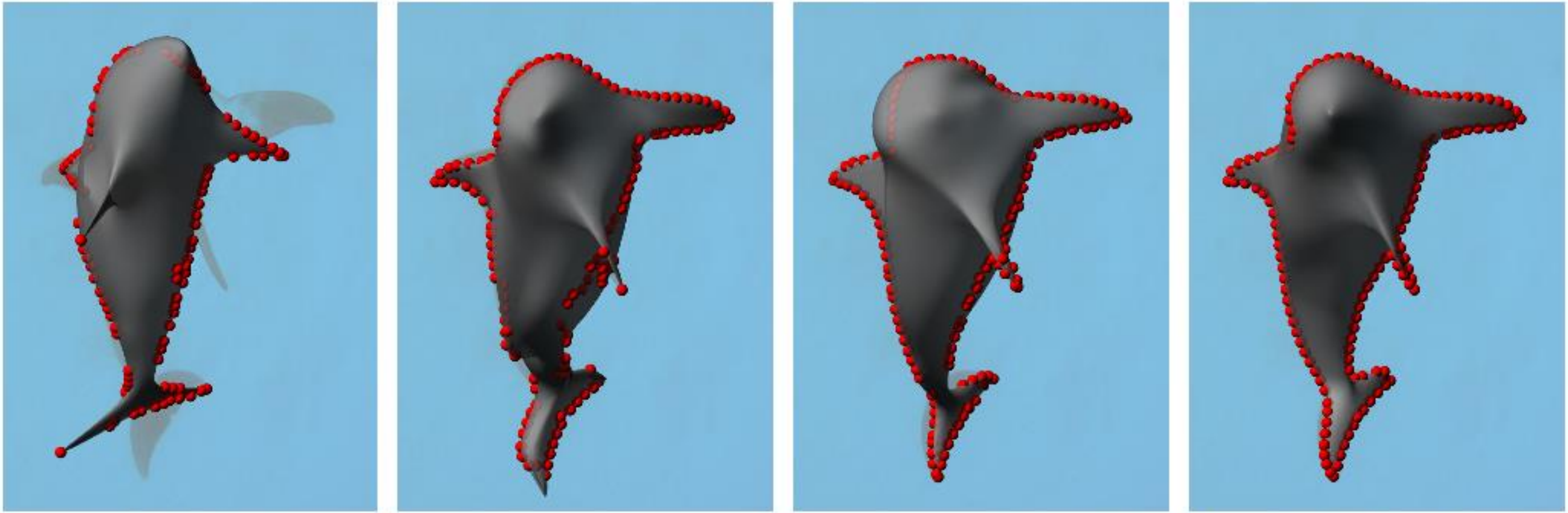


Also true but cheeky template



Morphable model parameters: 1

Optimization

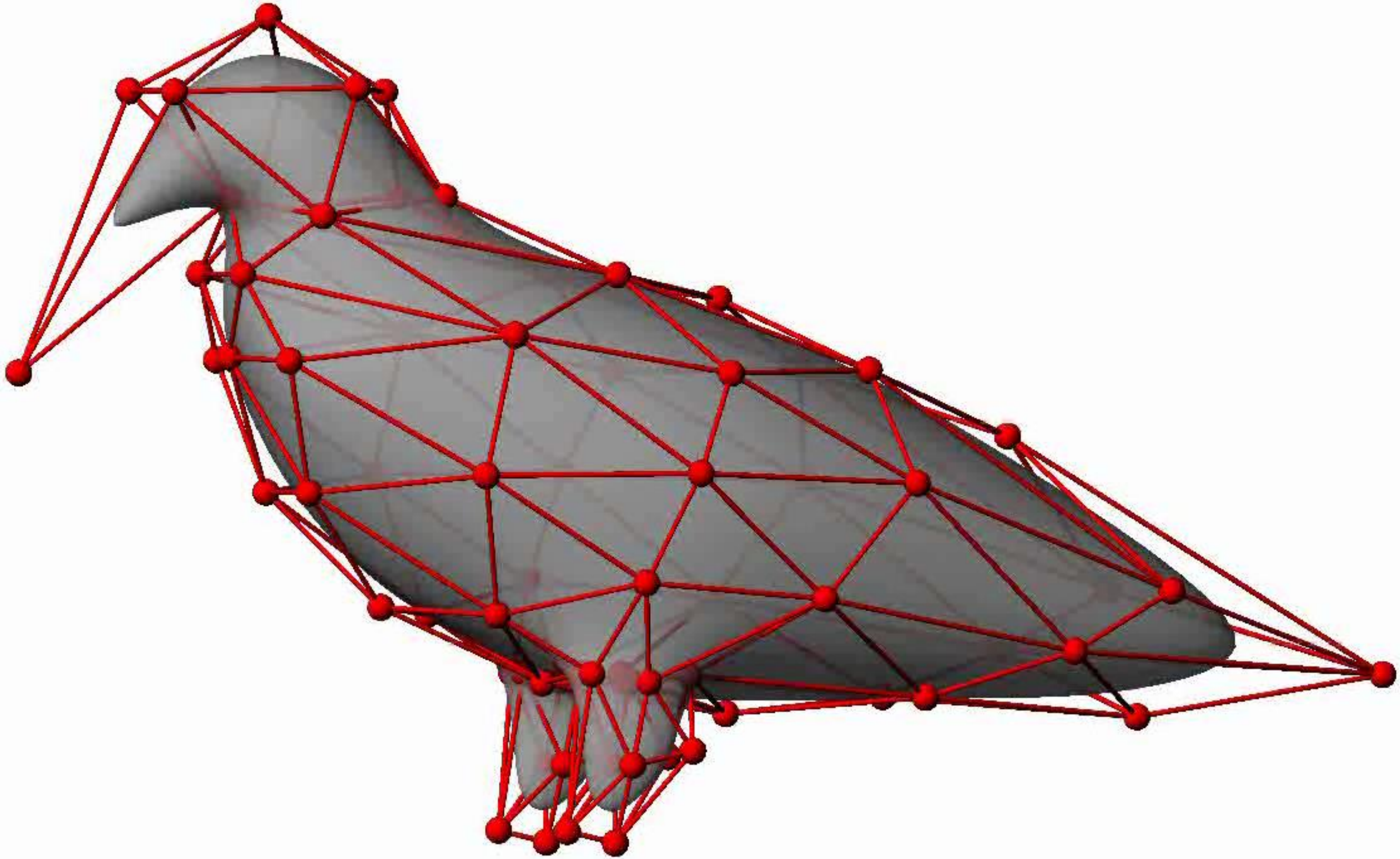


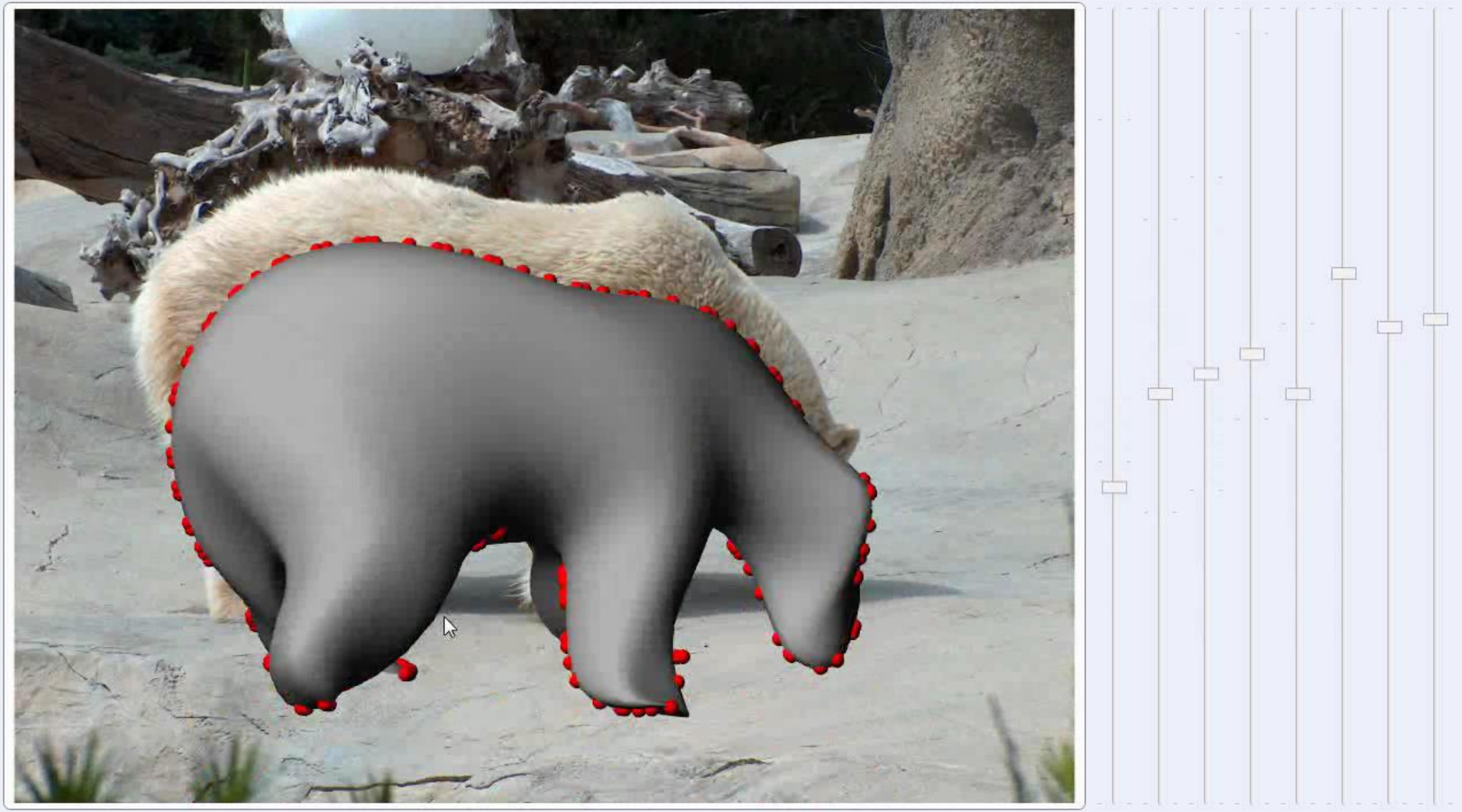
(a) Initial estimate.

(b) Only continuous local optimization, as described in Sec. 4.1.

(c) As (b), but including iterations of our global search (Sec. 4.2).

(d) As (c), but with reparametrization around extraordinary vertices.






DELL

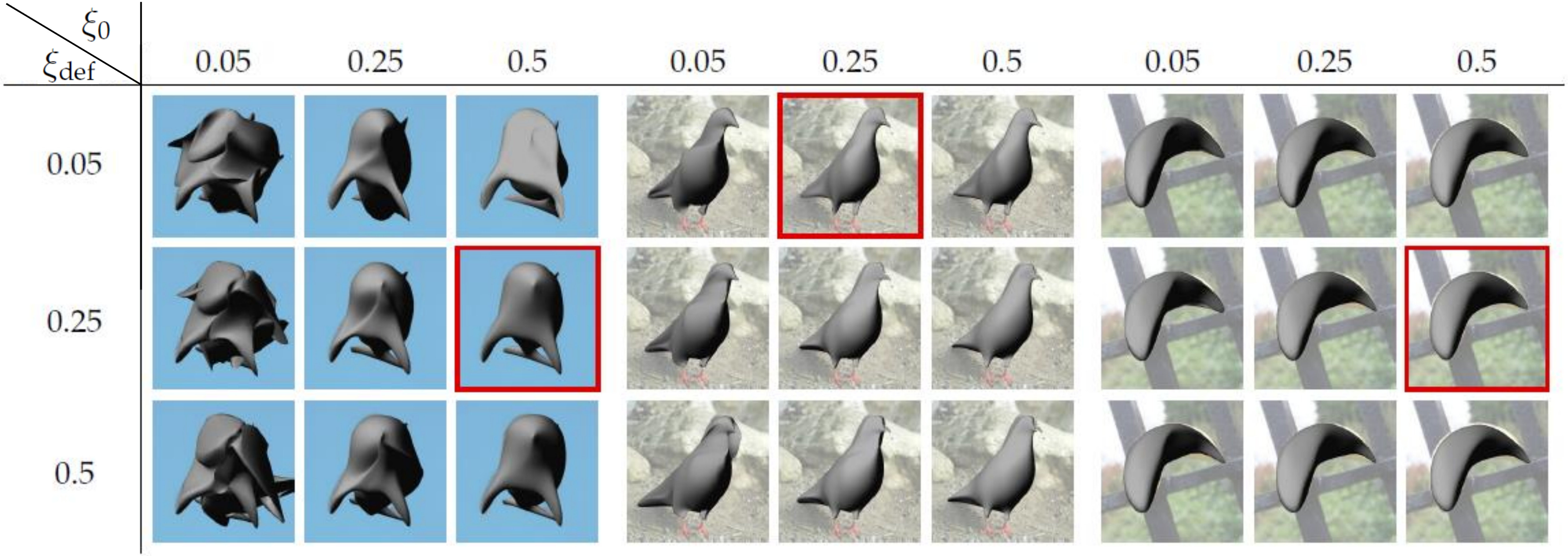
Parameter sensitivity

"Pixel" terms: noise level params

"Dimensionless" terms

"Smoothness" terms

$$E = \sum_{i=1}^{n}\left(E_i^{\text{sil}} + E_i^{\text{norm}} + E_i^{\text{con}}\right) + \sum_{i=1}^{n}\left(E_i^{\text{cg}} + E_i^{\text{reg}}\right) + \boldsymbol{\xi_0^2} E_0^{\text{tp}} + \boldsymbol{\xi_{\text{def}}^2} \sum_{i=1}^{n} E_m^{\text{tp}}$$



Reconstruction of *classes* from silhouettes

- With non-planar contour generators
- New results on subdivision surfaces
- And on rigid recovery from silhouettes

But room for improvement

- Better-than Gaussian model

- Discrete/continuous optimization
- Topology change, including sphere initialization
- Automation…
    1. Pose estimation
    2. Topology estimation

[All the above are the same problem]

Conclusions

- Yes, it requires manual input, but none of this was possible before.

- We need to understand what “automatic” means. We could implement an “automatic” version of this system, *to no advantage*.