COMPUTER VISION AND PHOTOGRAMMETRY

TEXTURING

CONTENT

- Texturing from Multi-view
- Paramterization
- Reflectance and approximations
  - The Parthenon Project
  - Flash/No-Flash Surface capture
  - Intrinsic Images

TEXTURING

PROBLEM

MULTI-VIEW TEXTURING

MULTI-VIEW TEXTURING

- Illumination and exposure changes
- Image scale
- Unreconstructed occluders
- Defocus Blurred Image

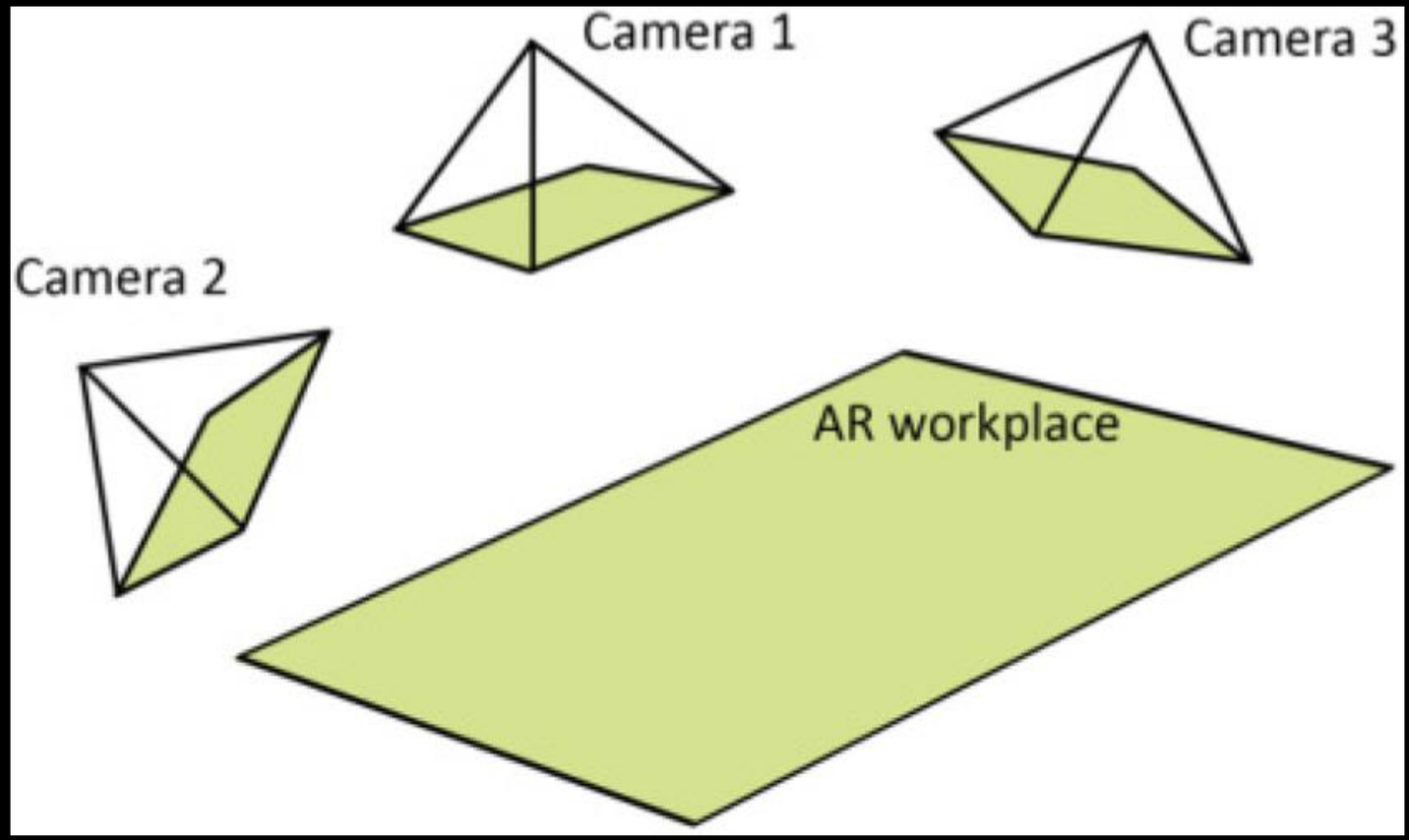
SOLUTION

TYPICAL APPROACH

- Pretty much the same as with panorama stitching.
- Select view
  - Angle, distance, … (best view)
  - Blending (weghts depend on quality) not necessary
- Reduce seams
  - Color correction

VIEW SELECTION
FOR EVERY TRIANGLE. . .

VIEW SELECTION: FRONT-TO PARALLEL



VIEW SELECTION: RESOLUTION (DISTANCE)

- Compare the area of the triangle in the reprojected image
- (s,t) = P Vertex

VIEW SELECTION:FOCUS

- The amount of defocus blur can be estimated with the gradient of the image
- Integrate the gradient magnitude of the triangle reprojection
  - Sobel operator
  - This integral already include the resolution and front viewing

VIEW SELECTION

VIEW SELECTION: OCLUSSIONS

- Visibility check
- Ray-Geometry intersections for a vertex
- You can also check by checking for overlapping on triangle reprojection and check for distance to camera.

VIEW SELECTION: UNMODELLED OCCLUSIONS

- Photo-consistency check

VIEW SELECTION: UNMODELLED OCCLUSIONS

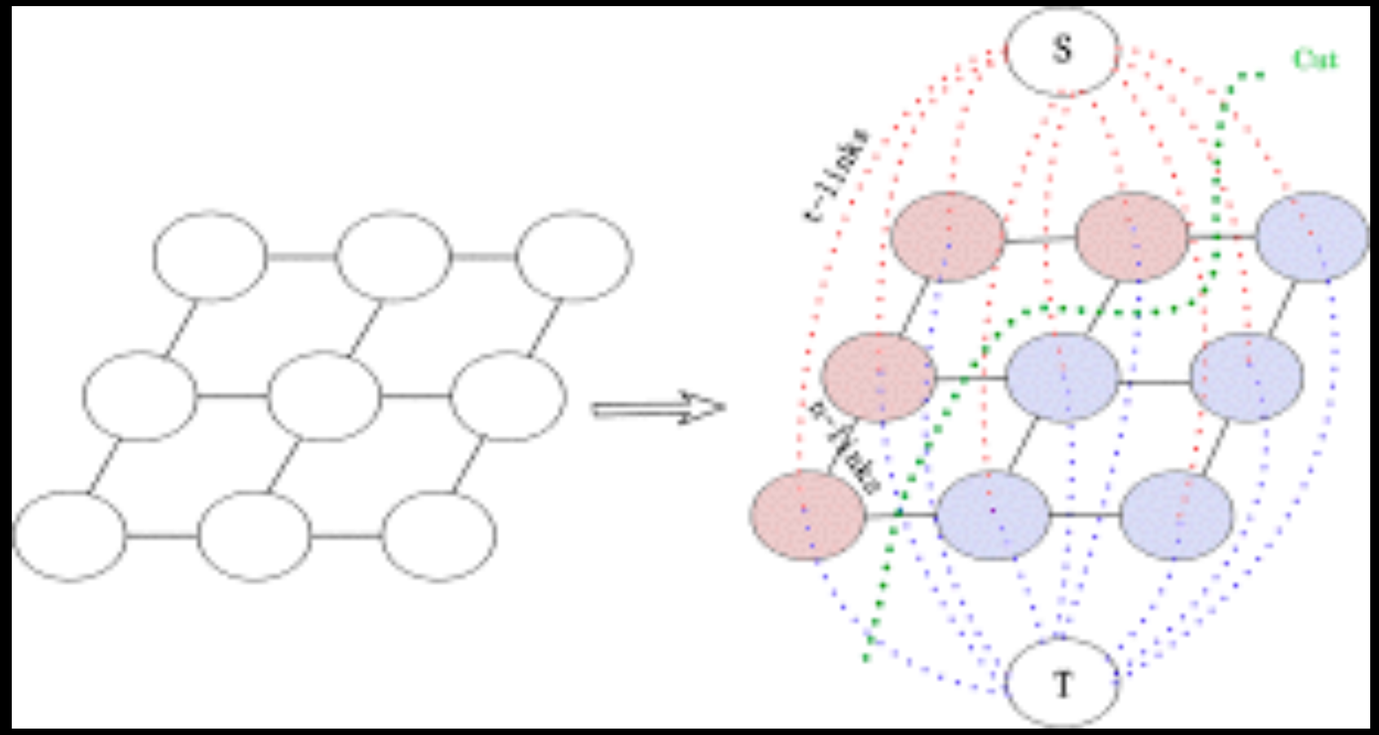
- Difference from the mean
- You can make it more robust by iteratively update the mean only with inliers
  - compute mean
  - remove colors far from the mean
  - repeat

CRITERIA

- Can be considered a labelling problem
- Include an smoothness term, so adjacent triangles tend to be textured from the same image
- Formulated as a Markov Random Field Optimization

MRF FORMULATION

MRF: GRID



MRF: ENERGY MINIMISATION

$$E(l) = \sum_{F_i \in \text{Faces}} E_{\text{data}}(F_i, l_i) + \sum_{(F_i, F_j) \in \text{Edges}} E_{\text{smooth}}(F_i, F_j, l_i, l_j)$$

- F are faces
- L are views

MRF: SMOOTHNESS TERM

- Complicated
  - Look at the differences between pixels on the edges
  - more differences - smaller the cost
- Simple
  - 0 if they are the same view
  - 1 if they are different

MRF: MINIMISATION WITH GRAPH CUTS

- Meant to produce the “optimal” trade-of between quality of data and larger patches

MRF: MINIMISATION WITH GRAPH CUTS

COLOR ADJUSTMENT

TEXT

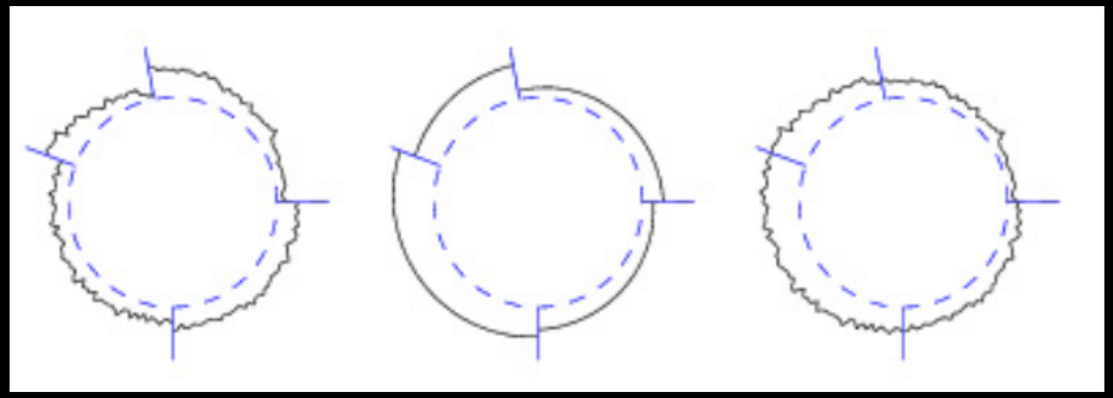
COLOR ADJUSTMENT: GLOBAL ADJUSTMENT

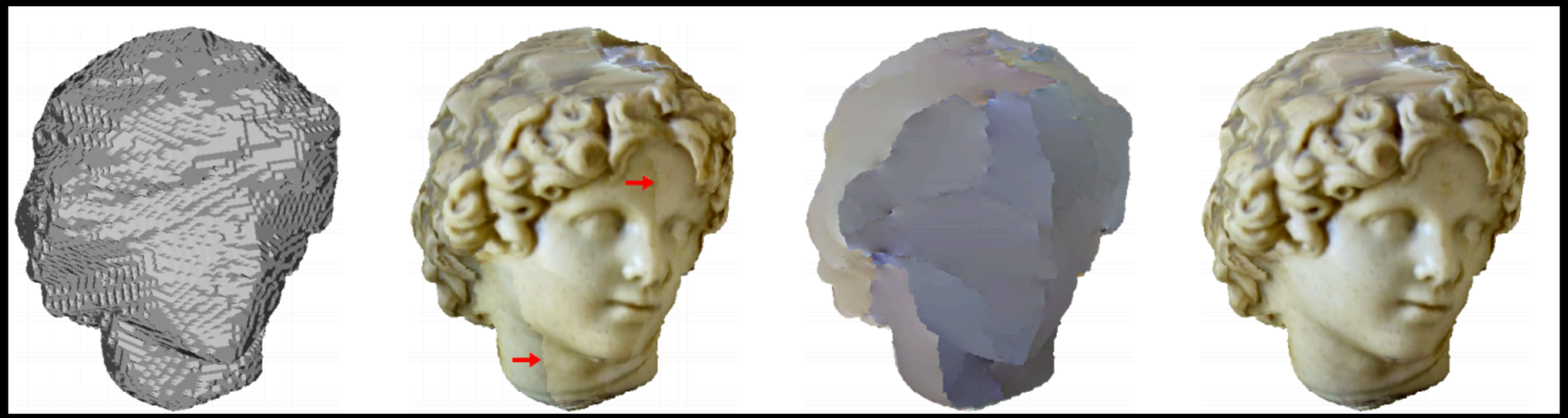
COLOR ADJUSTMENT: GLOBAL ADJUSTMENT

$$\underset{\mathbf{g}}{\operatorname{argmin}} \sum_{\substack{v \text{ (split into} \\ v_{\text{left}} \text{ and } v_{\text{right}}) \\ \text{lies on a seam}}} \left(f_{v_{\text{left}}} + g_{v_{\text{left}}} - \left(f_{v_{\text{right}}} + g_{v_{\text{right}}}\right)\right)^2 + \frac{1}{\lambda} \sum_{\substack{v_i, v_j \text{ are ad-} \\ \text{jacent and in} \\ \text{the same patch}}} \left(g_{v_i} - g_{v_j}\right)^2$$

- f colour at a vertex in a seam
- g is the correction factor
- minimize the difference at the seam and the difference in the correction factor at adjacent vertices

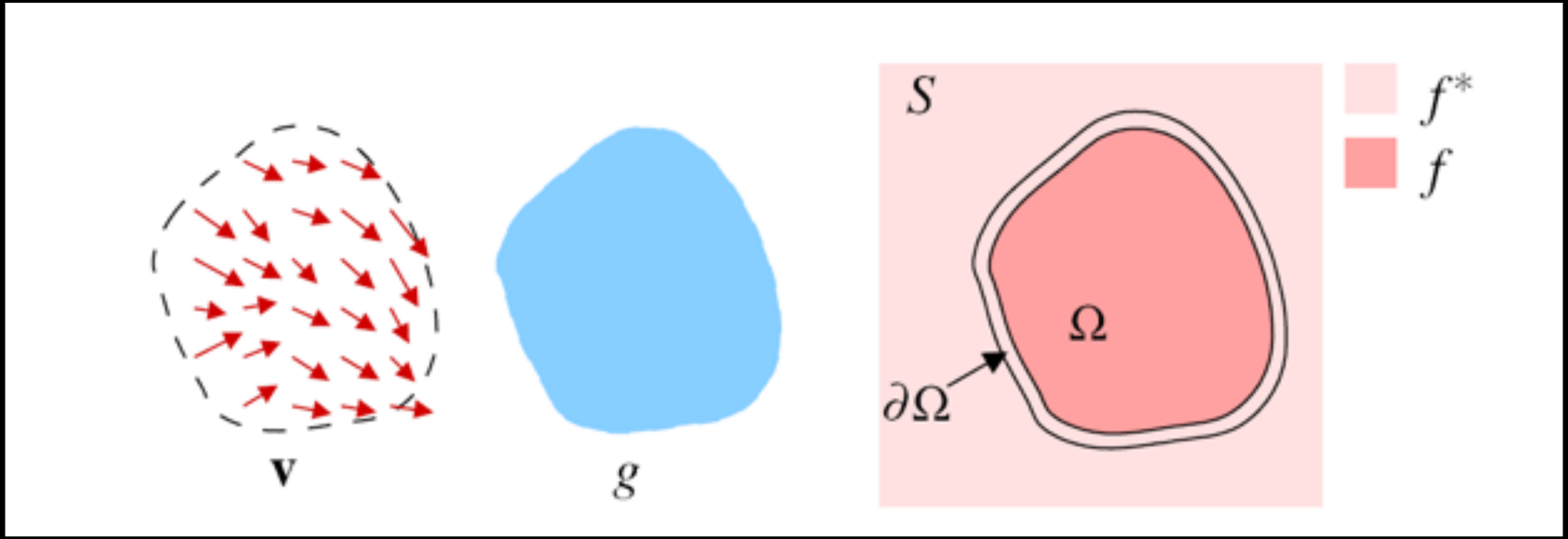
COLOR ADJUSTMENT: GLOBAL ADJUSTMENT





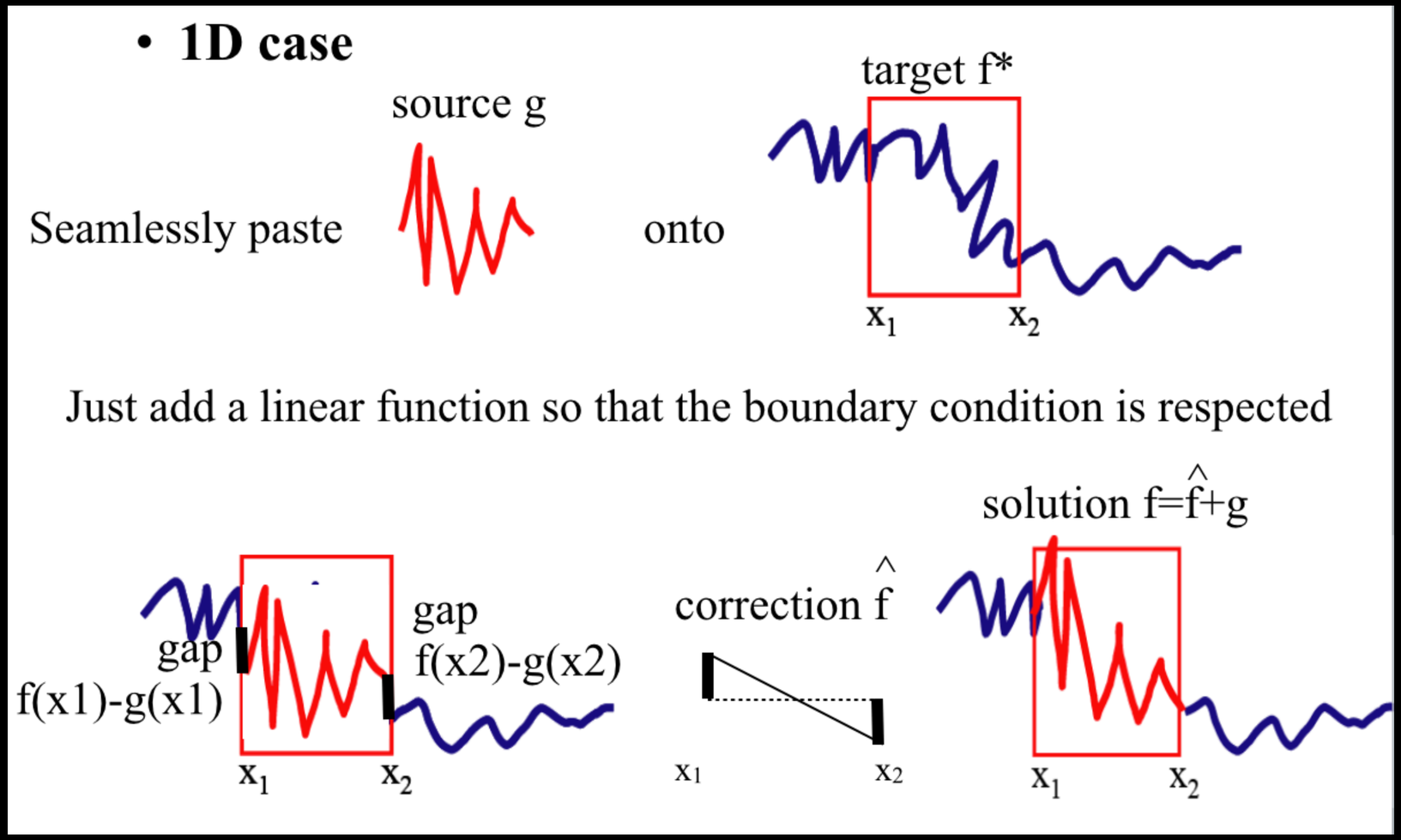
COLOR ADJUSTMENT: POISSON EDITING

COLOR ADJUSTMENT: POISSON EDITING



$$\min_f \iint_\Omega |\nabla f - \mathbf{v}|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

COLOR ADJUSTMENT: POISSON EDITING



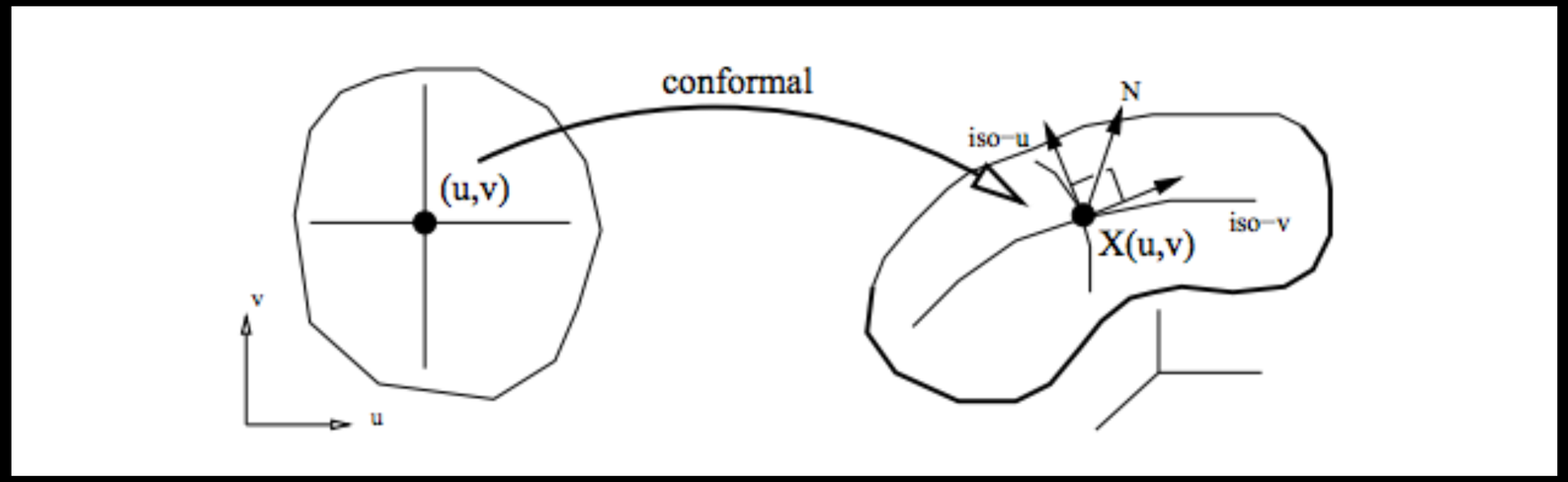
PARAMETERIZATION

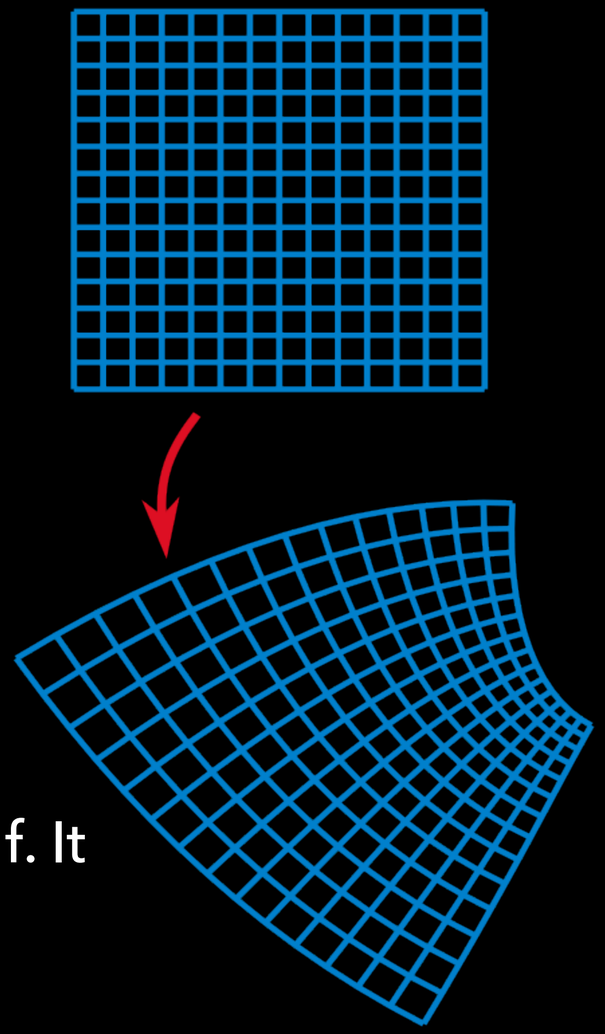
TEXTURE ATLAS GENERATION

- Segmentation
- Parameterization
- Packing
- Least-Squares Conformal Maps
  - minimizes angle deformations and non-uniform scalings
  - unique minimum
  - large charts
  - no triangle flipping
  - independent of mesh resolution

CONFORMALITY

- The curves are orthogonal and have the same norm.





- A rectangular grid (top) and its image under a conformal map f. It is seen that f maps pairs of lines intersecting at 90° to pairs of curves still intersecting at 90°.

SEGMENTATION: FINDING THE CUTS

TEXTURE ATLAS

- Charts: part *homeomorphic* to a disc
- Trivial: one chart per triangle
  - Impractical for texture painting, difficult for processing
- Criteria
  - Geometry to be easy to get a conformal map
  - Cuts to be in difficult places to detect

TEXTURE ATLAS

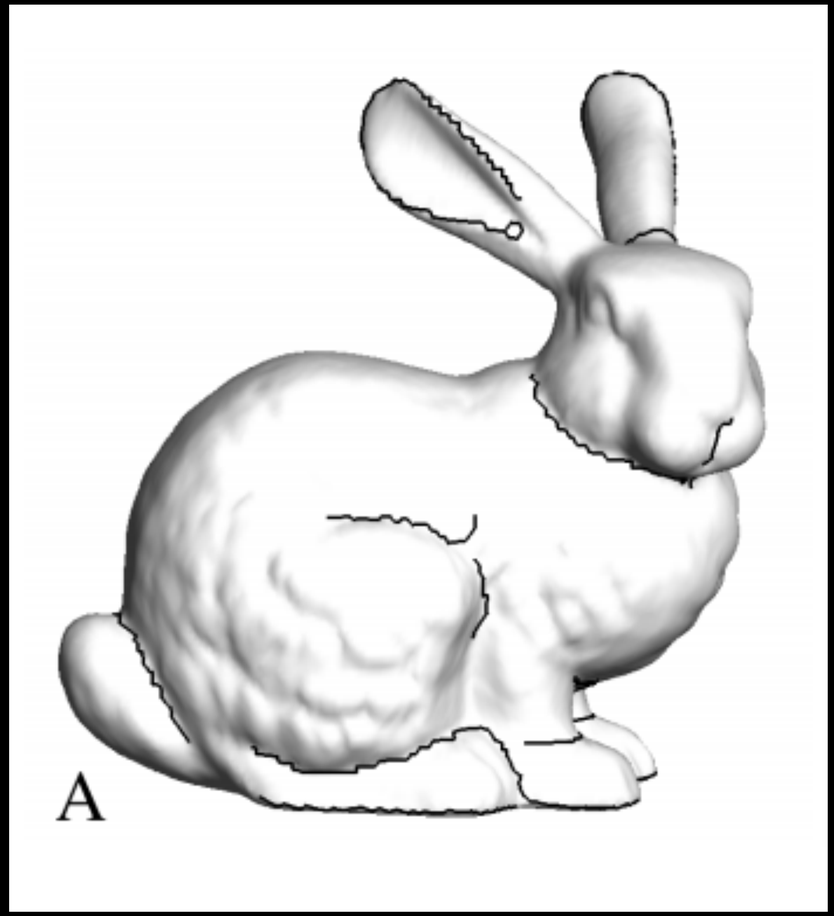
- LSCM
  - Charts boundaries at places where don’t cause texture artifacts
  - Charts must be homeomorphic to discs and parametrisable without much distortion

SEGMENTATION: LSCM

- To avoid texture artefacts -> cuts in zones with high curvature
- To decompose in “discs” -> decompose the model into cylinders
- Algorithm:
  - detect high curvature zones
  - chart growing them to meet at theses zones

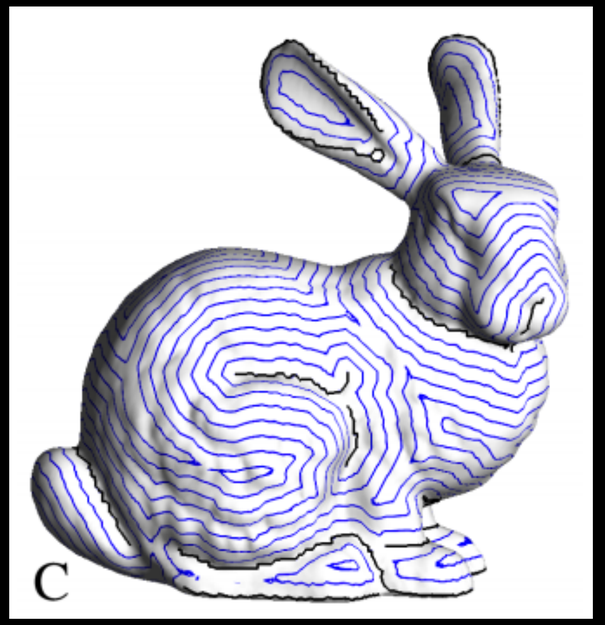
SEGMENTATION: DETECT FEATURES

- Compute sharpness criterion: angle between normals
- Choose threshold to keep a % of edges (they keep 5%)
- Grow “features” to anticipate best cut path and remove short features

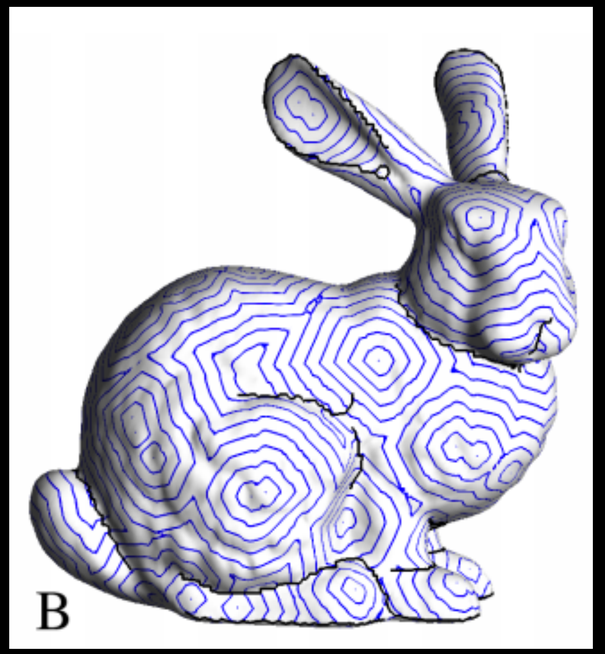


SEGMENTATION: EXPAND CHARTS

- Greedy algorithm expanding all charts simultaneously from *seeds*
- Seed are local maxima of *distance_to_feature* function
- Propagate from seed using distance_to_feature



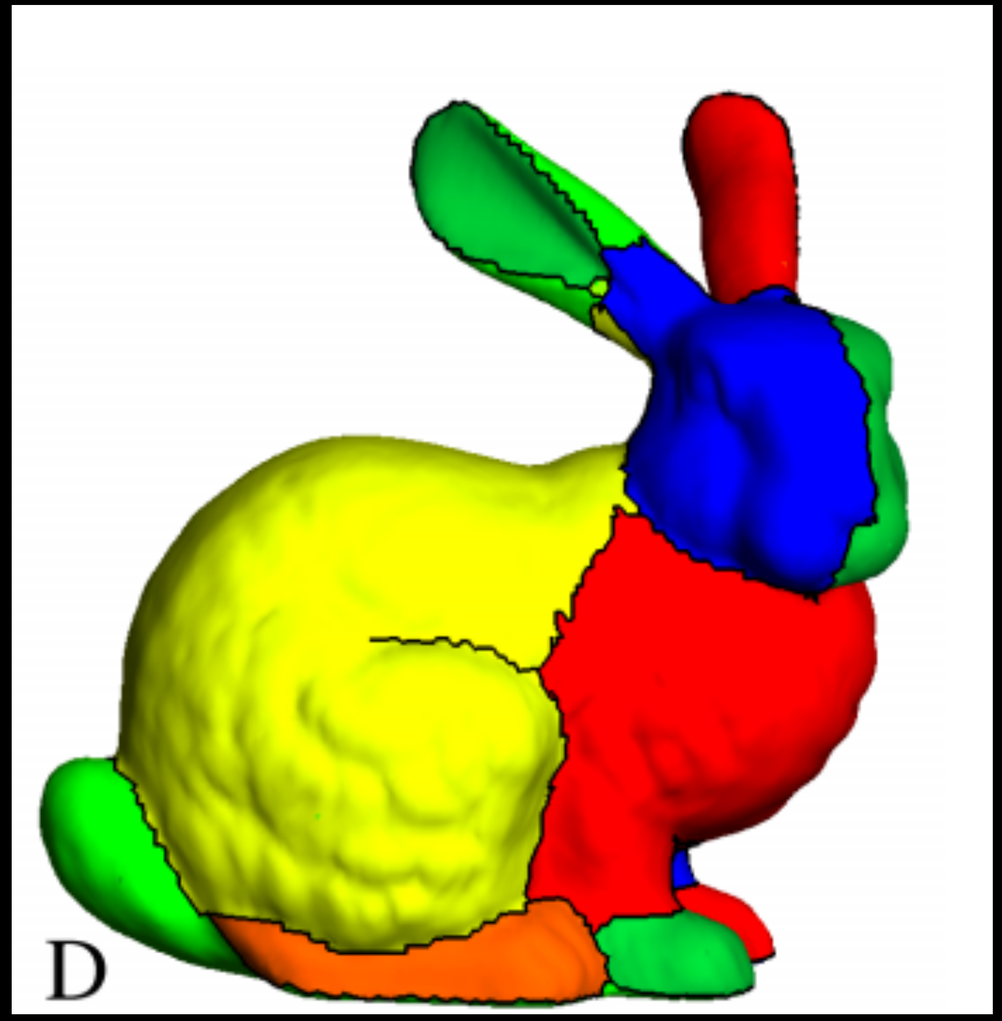
DISTANCE TO FEATURE



DISTANCE TO SEED
SEEDS ARE AT THE CENTRES

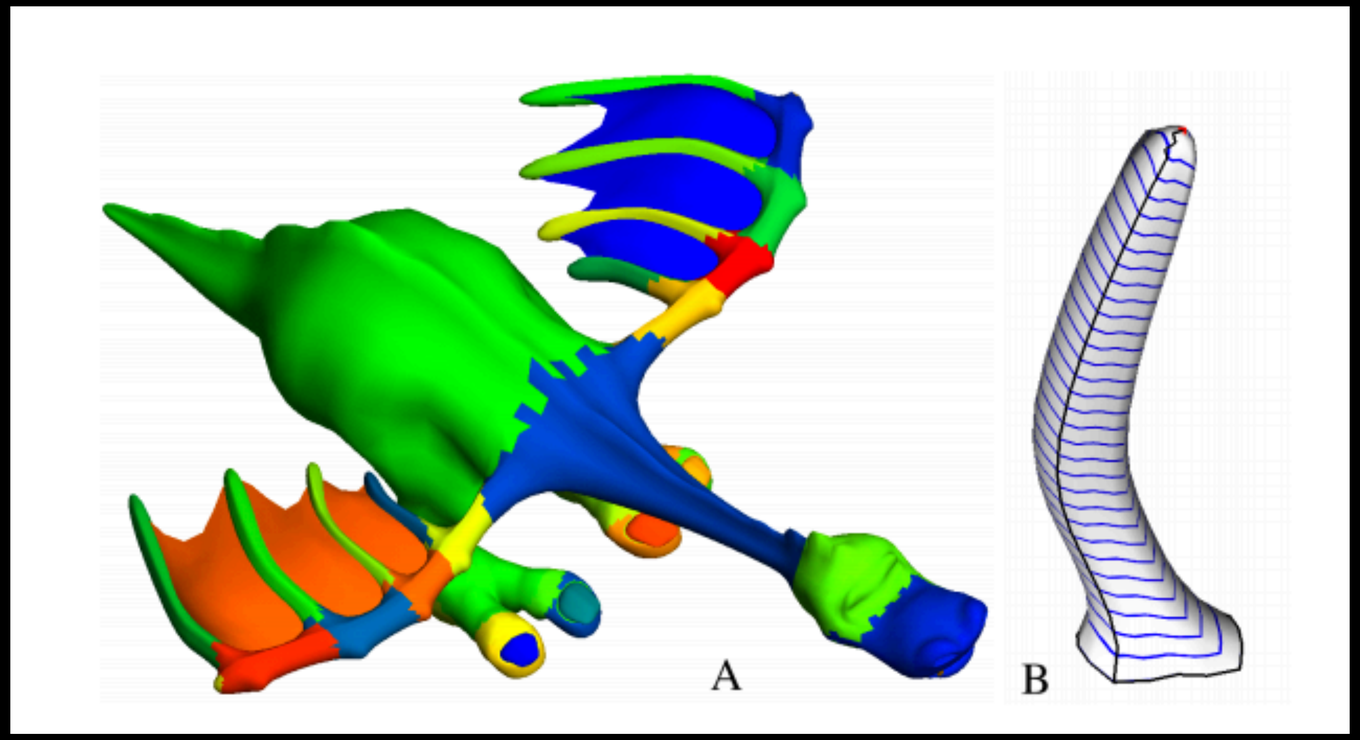
SEGMENTATION: DETECT FEATURES

- Charts are merged if they meet at a small distance from their seed
- Choose threshold to keep a % of edges (they keep 5%)
- Grow “features” to anticipate best cut path and remove short features



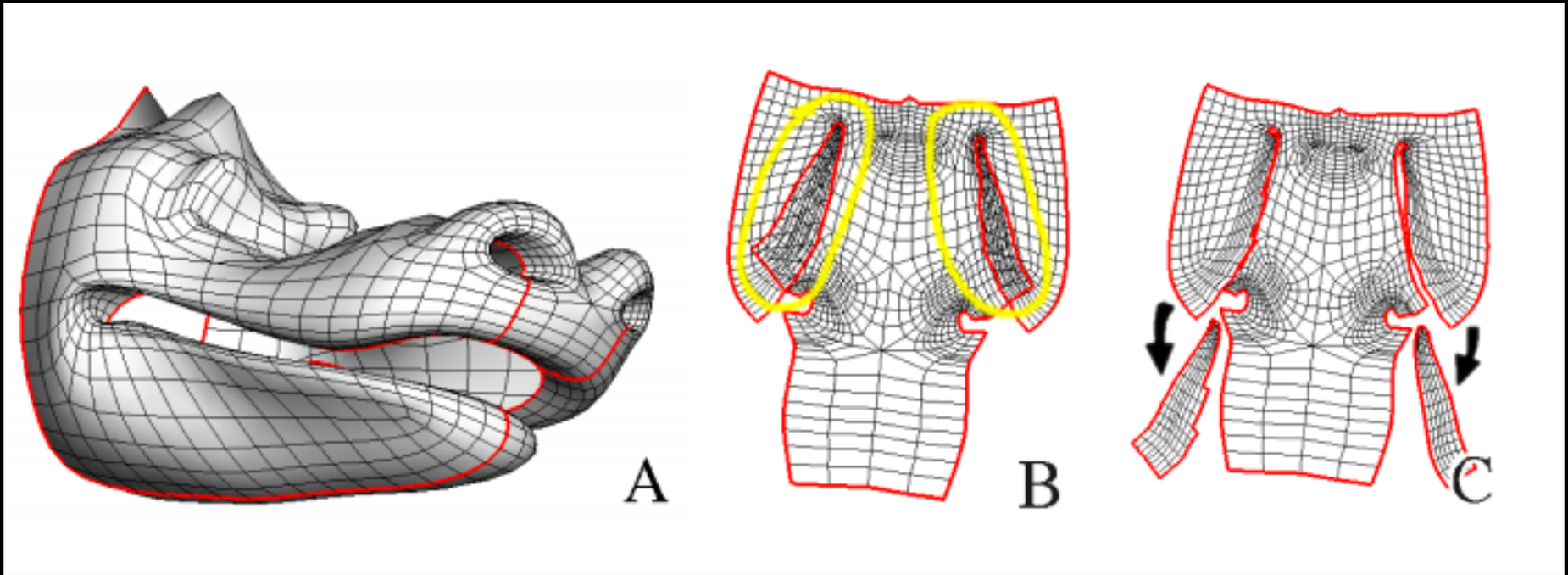
SEGMENTATION: EXPAND CHARTS

- Finger/Sock like areas are detected and segmented
- Add an extra cut doe they are easier to parameterise



SEGMENTATION: EXPAND CHARTS

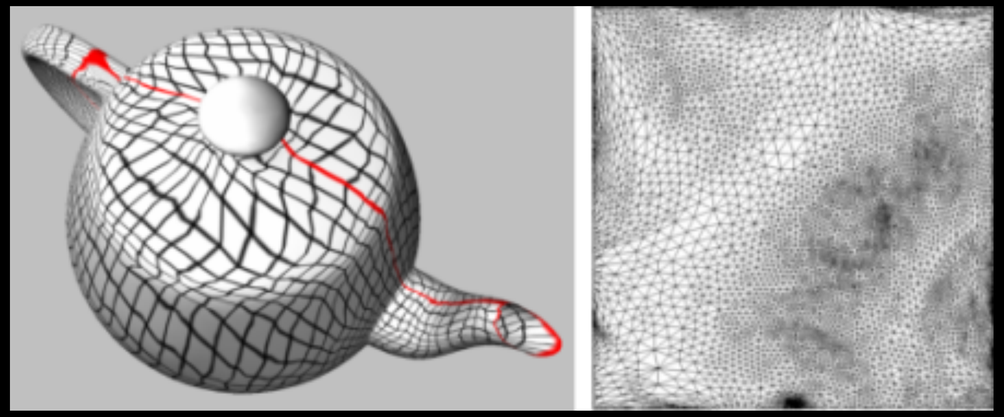
- If some overlap happens (unlikely with this method) new cuts are created

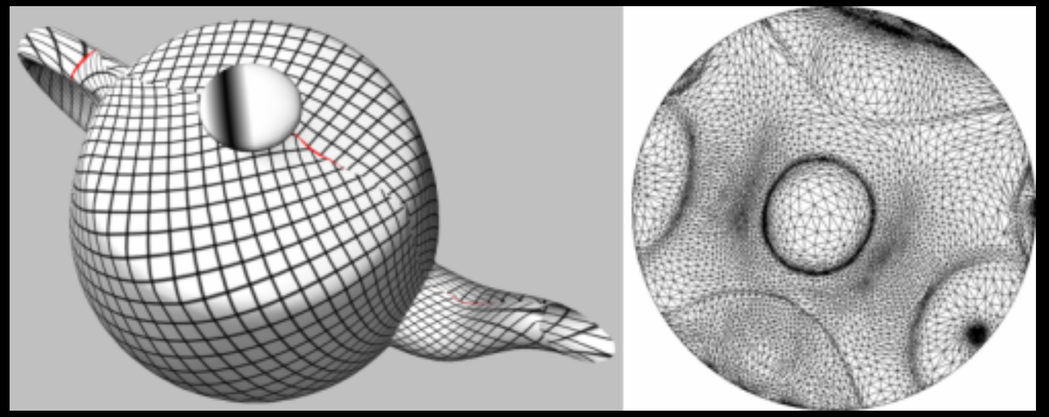


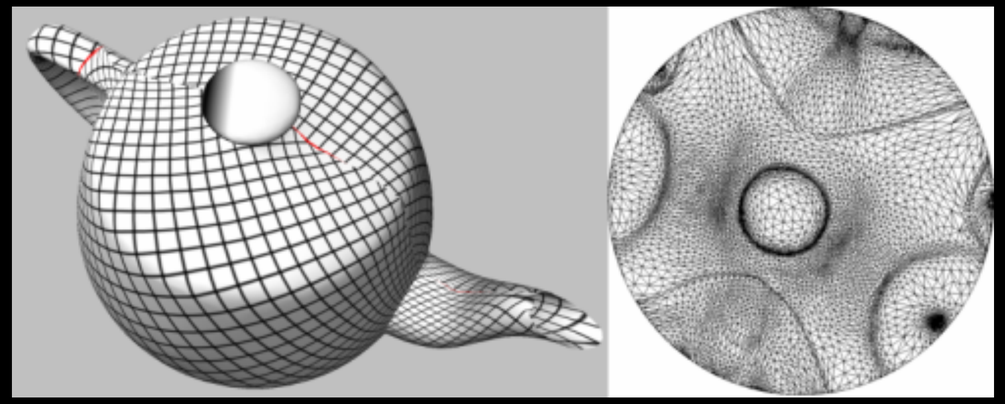
PARAMETERIZATION: FLATTENING

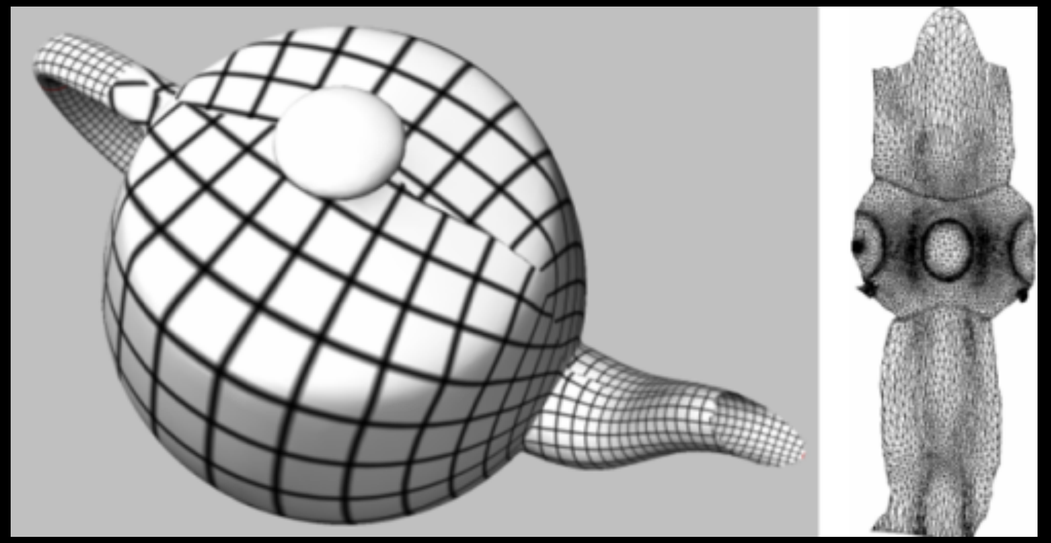
TEXT

MANY METHODS









MINIMIZE DISTORTION

- Least square solution that minimises the violation of conformality.
  - This is formulated as satisfaction of the Cauchy-Riemann equation.
- The matrix A has full rank when the number of pinned vertices, i.e. p, is larger than or equal to 2.
- As a consequence, the minimization problem has a unique solution when $p > 2$.
- It solves a (2 × #triangles) × #vertices sparse linear system in the least squares sense.

TAKE AWAY MESSAGE

- Conformal map
  - You can map your mesh to quads.
- You want a mapping that changes angles, and scale as little as possible.

SELECT A BOUNDARY

- some methods require to specify a boundary
  - circle, square, (previous method: free boundary)
- minimize scaling
- minimize angle deformation
- minimize deformation depending on texture (low frequency information, more deformation)

CGAL: THE COMPUTATIONAL GEOMETRY ALGORITHMS LIBRARY

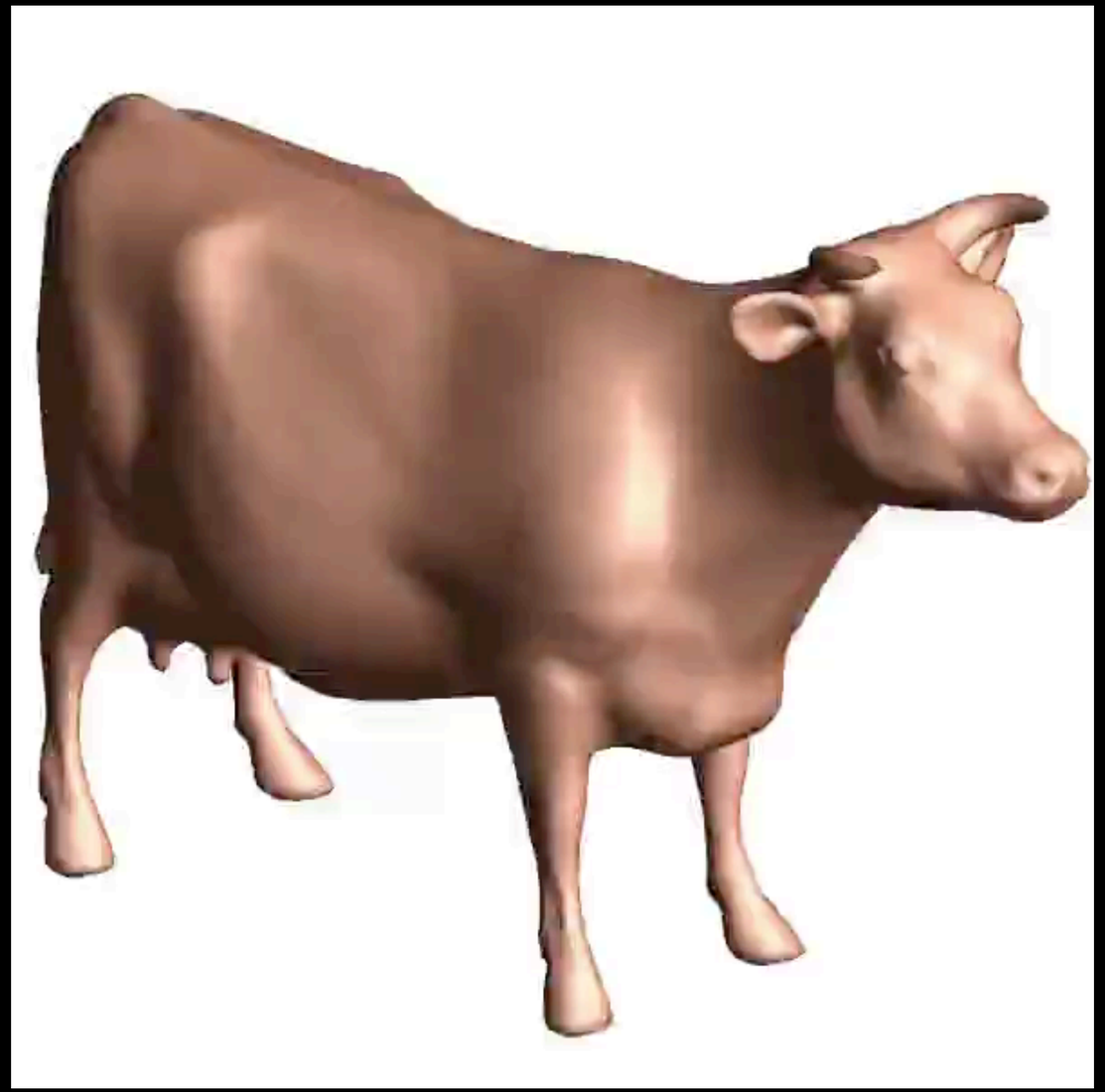
- Same library as for meshing algorithms
- More about paramterization:
- http://doc.cgal.org/latest/Surface_mesh_parameterization/index.html

GEOMETRY IMAGES

- Parametrise to an square

GEOMETRY IMAGES

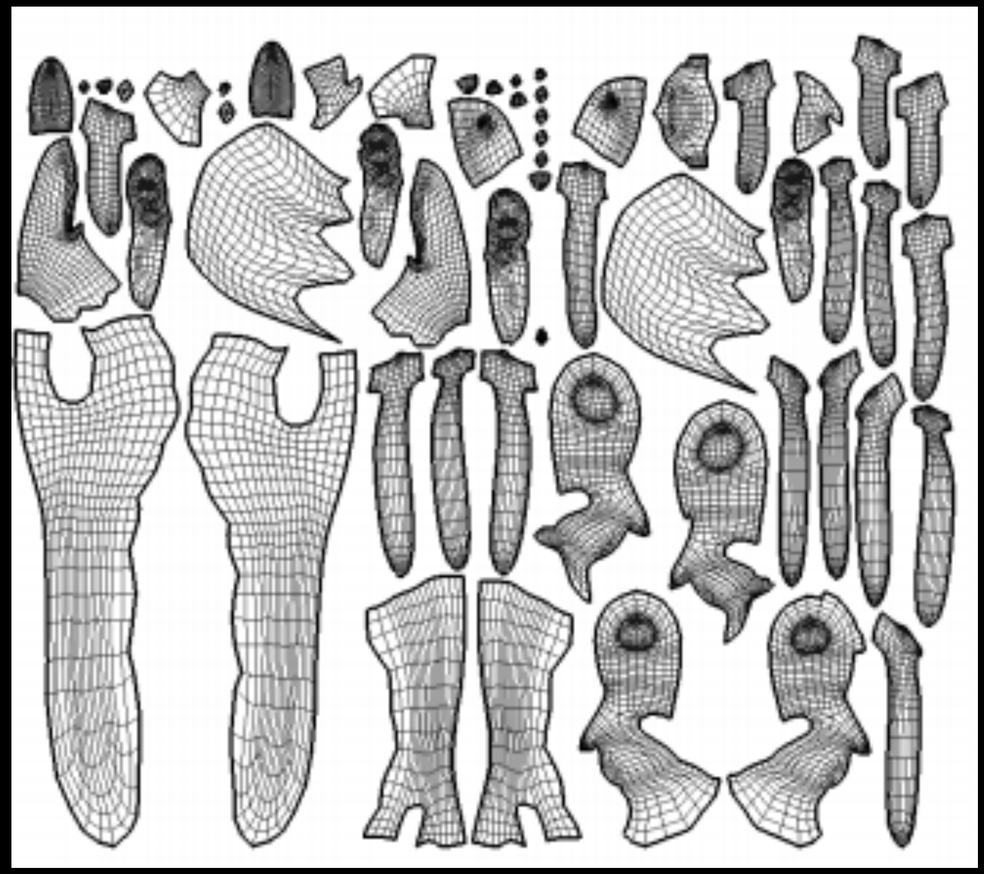
- Parametrise to an square
- *“Assuming that the cow is a sphere”*



PACKING

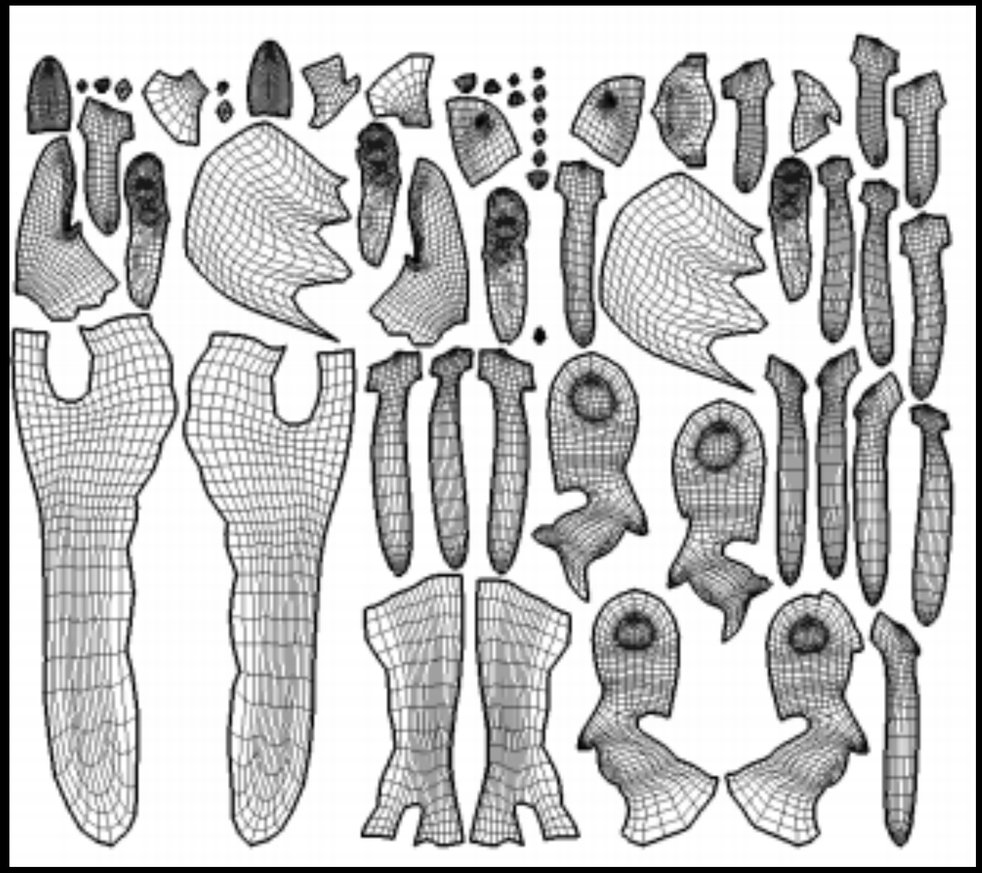
PACKING CHARTS

- Optimize the position of the charts to minimise the waste of space
- The use the actual boundary
  - other methods use the bounding box
  - “Tetrix” inspired

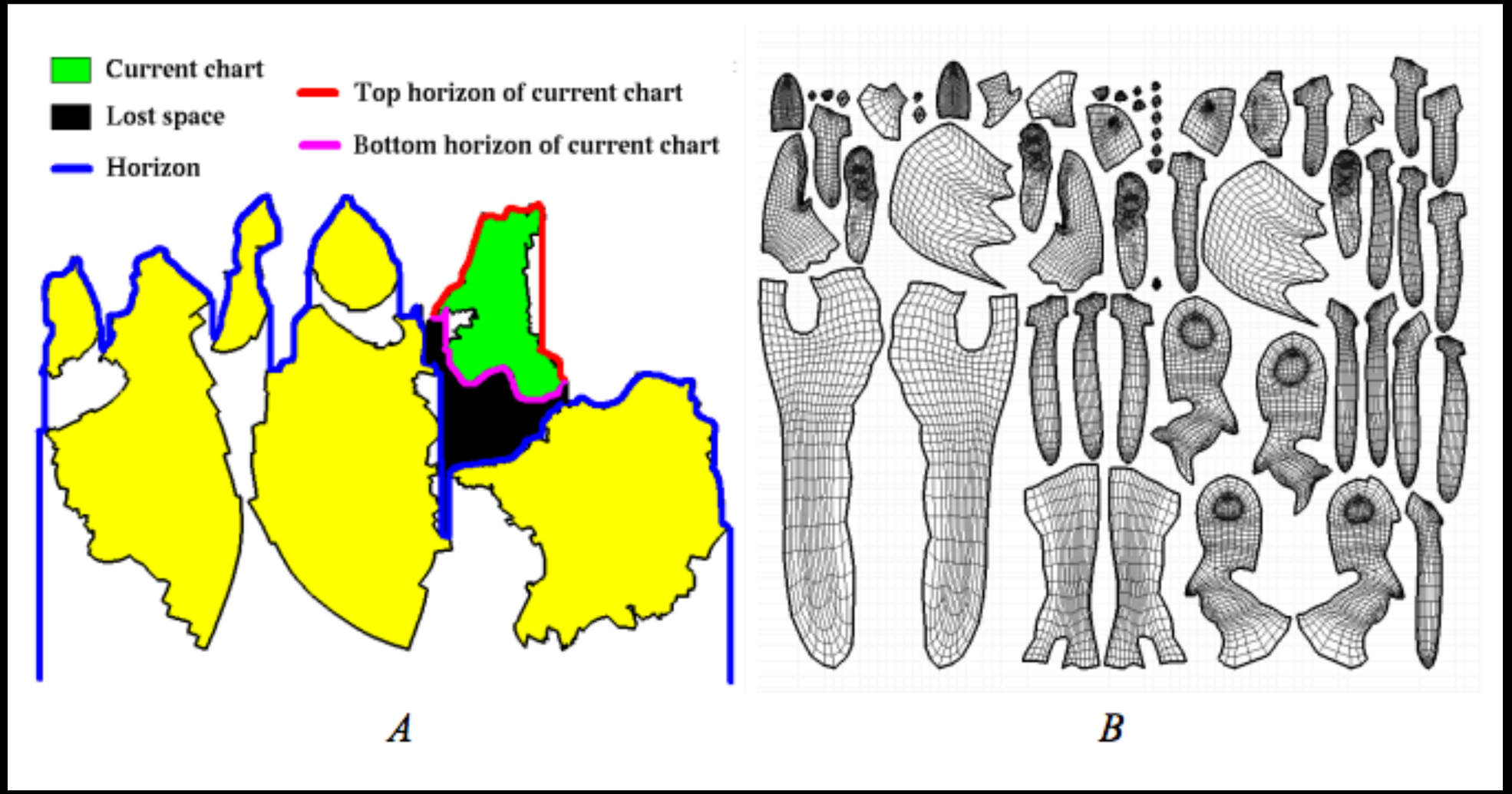


PACKING CHARTS

- Scale area in (u,v) to match its area in 3D
- maximum diameter oriented vertically
- approximate area
- add some border to avoid overlap with mip-mapping

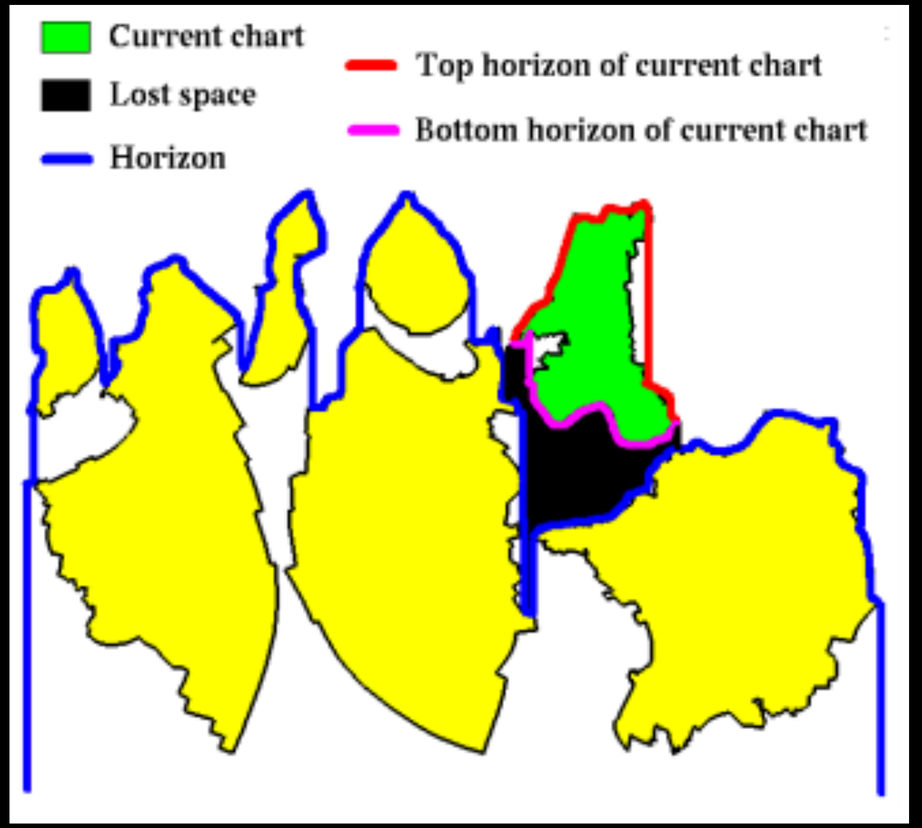


PACKING CHARTS

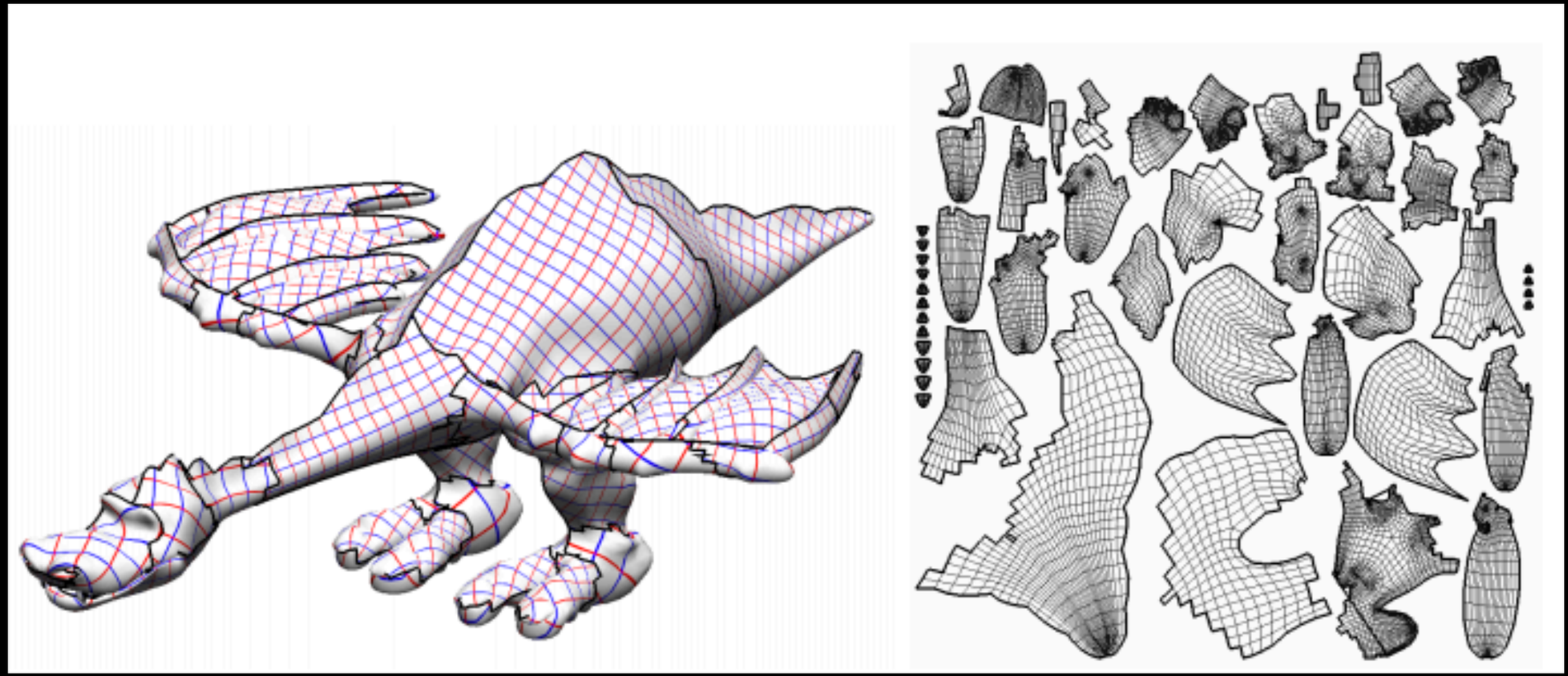


PACKING CHARTS

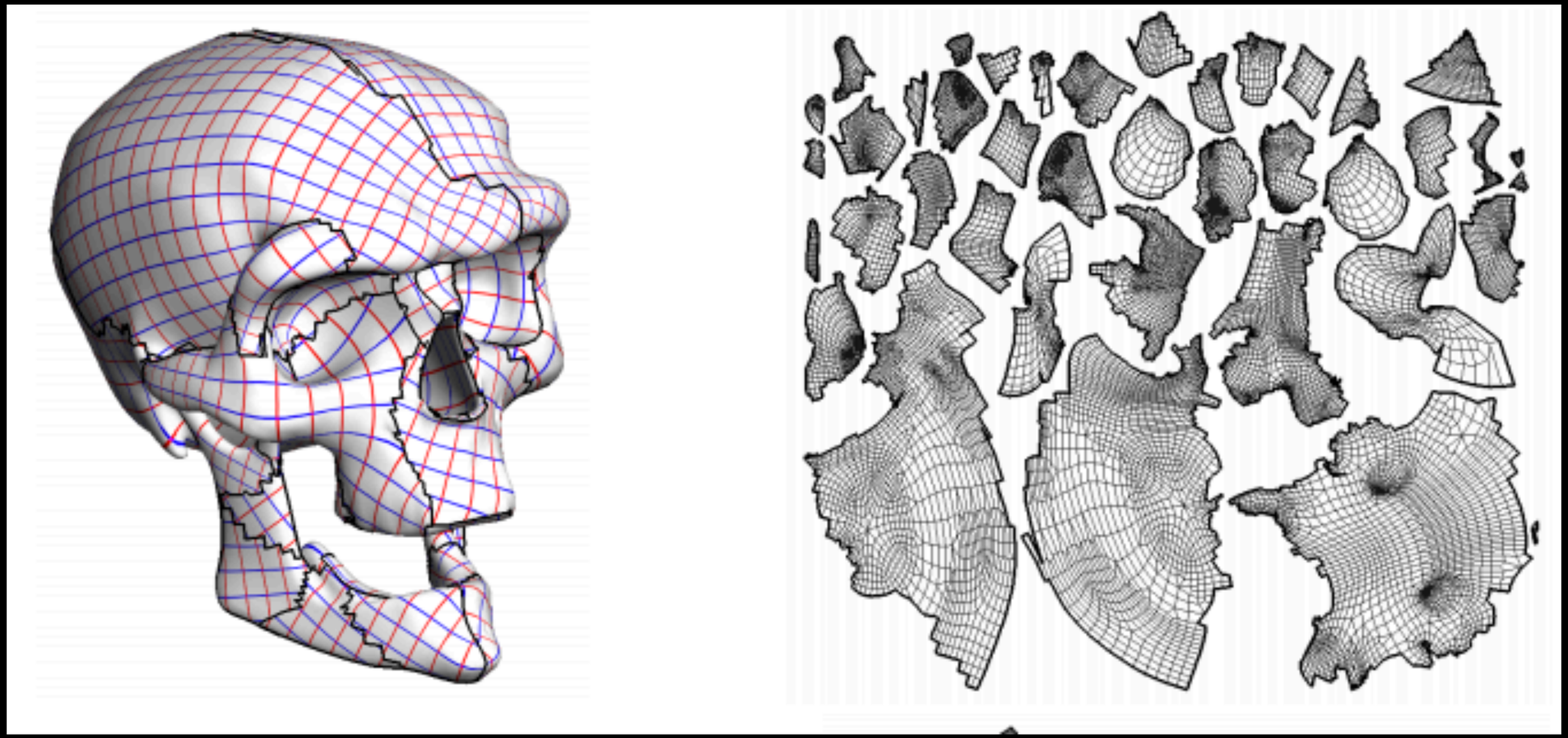
- Insertion method
  - Minimize the waste space
  - try with all vertical positions in horizon for the lower-left corner



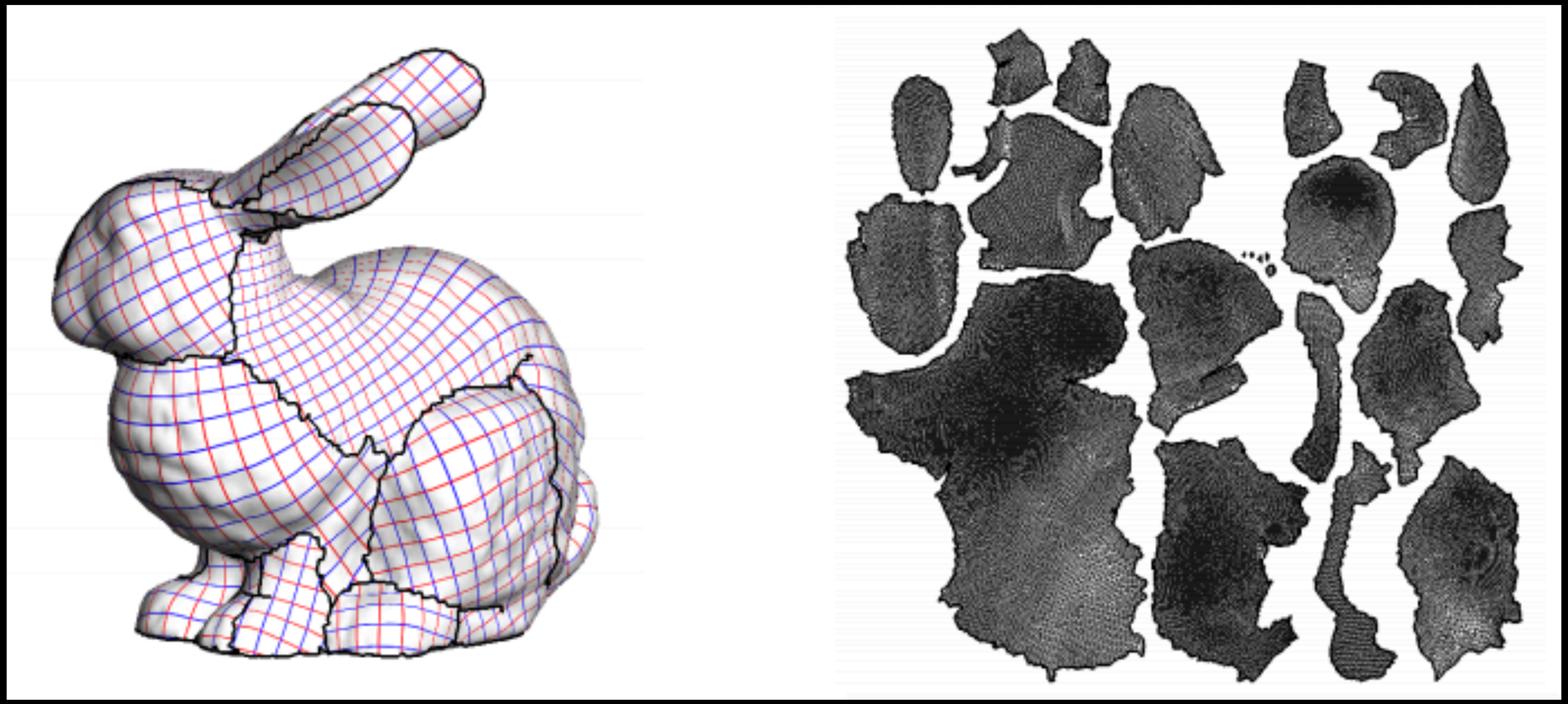
PACKING CHARTS



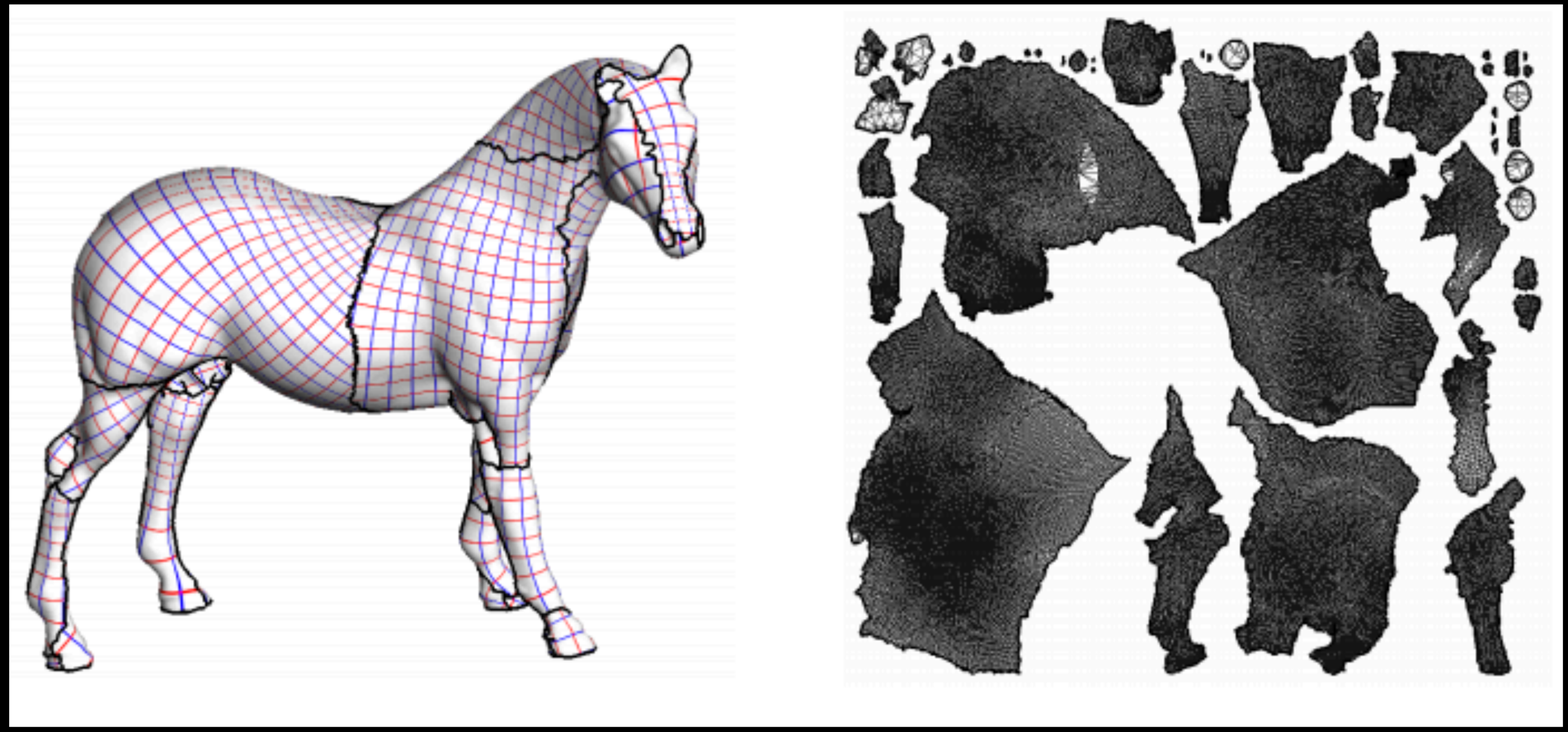
PACKING CHARTS



PACKING CHARTS



PACKING CHARTS



SUMMARY

- Cut the model to homographic to disc
  - Segment the geometry
- Parameterize it (flat it)
- Pack the charts to minimise space

ALSO IN TEXTURE SPACE...

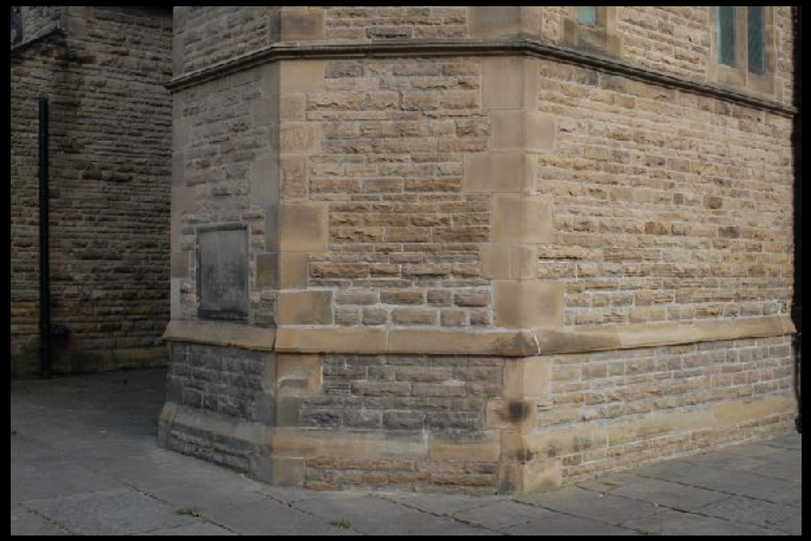
OPTIMIZATION IN TEXTURE SPACE

- The same as before, can be done in texture space (optimize per pixel, rather than per face.)

OPTIMIZATION IN TEXTURE SPACE

- From the geometry, get the “texture space” representation.

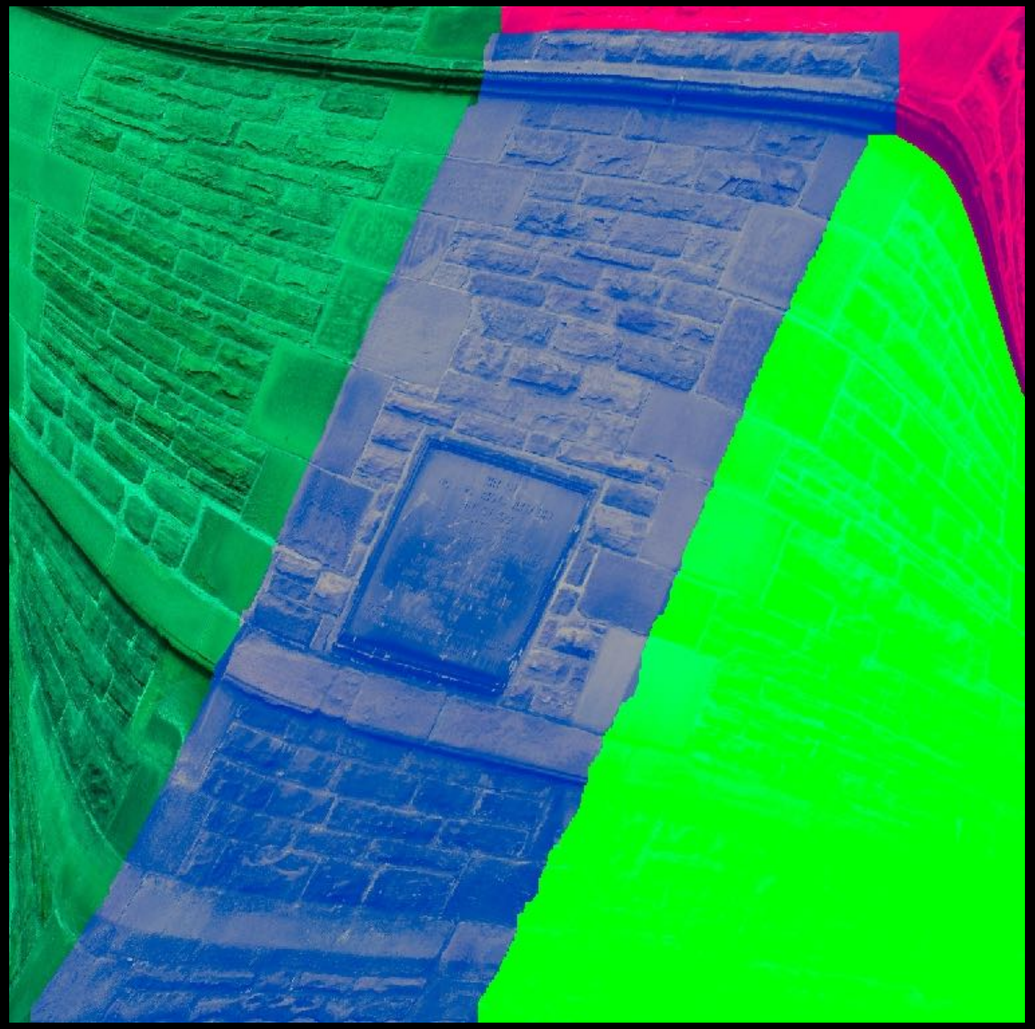




OPTIMIZATION IN TEXTURE SPACE

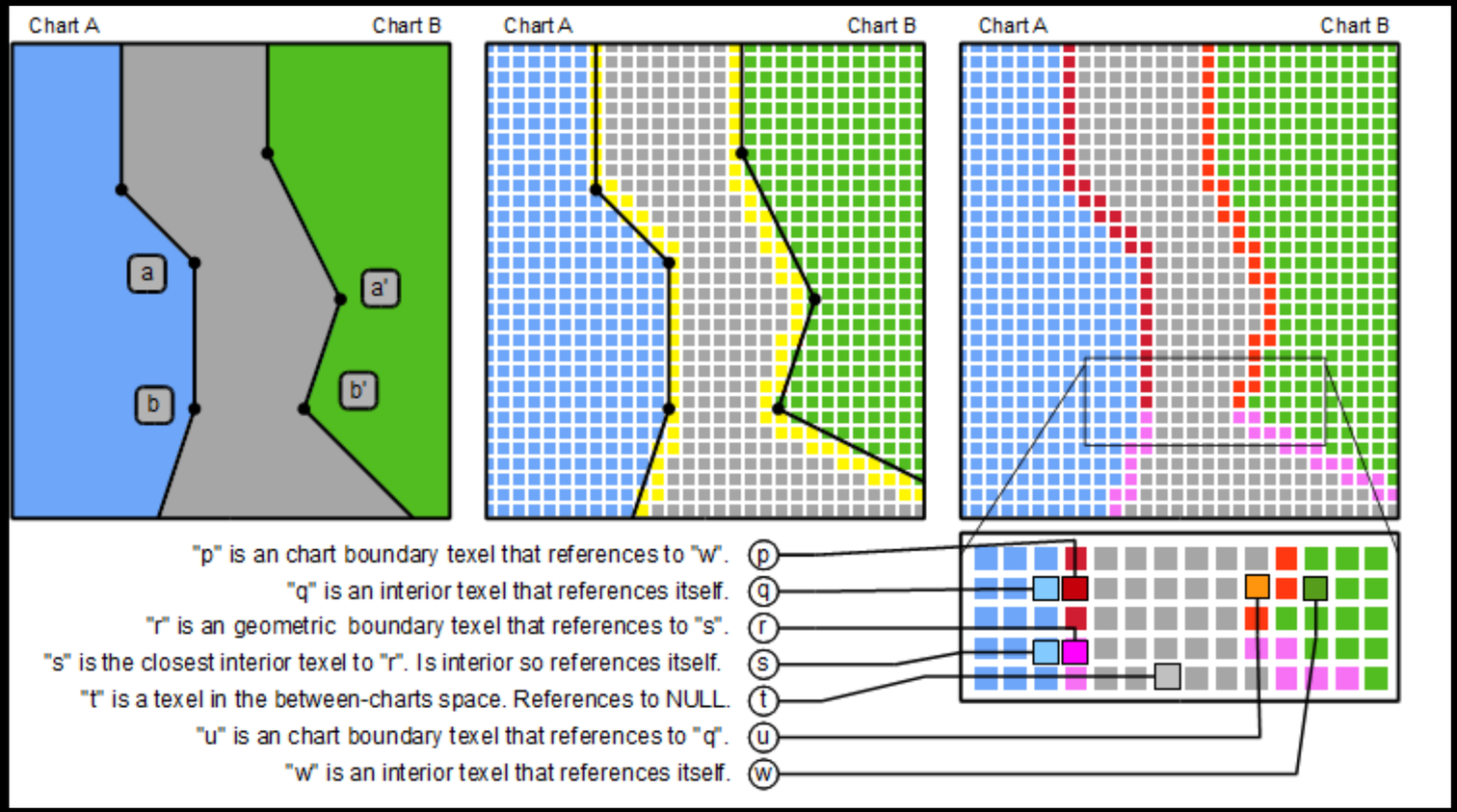
- Reproject all the

OPTIMIZATION IN TEXTURE SPACE



OPTIMIZATION IN TEXTURE SPACE

OPTIMIZATION IN TEXTURE SPACE



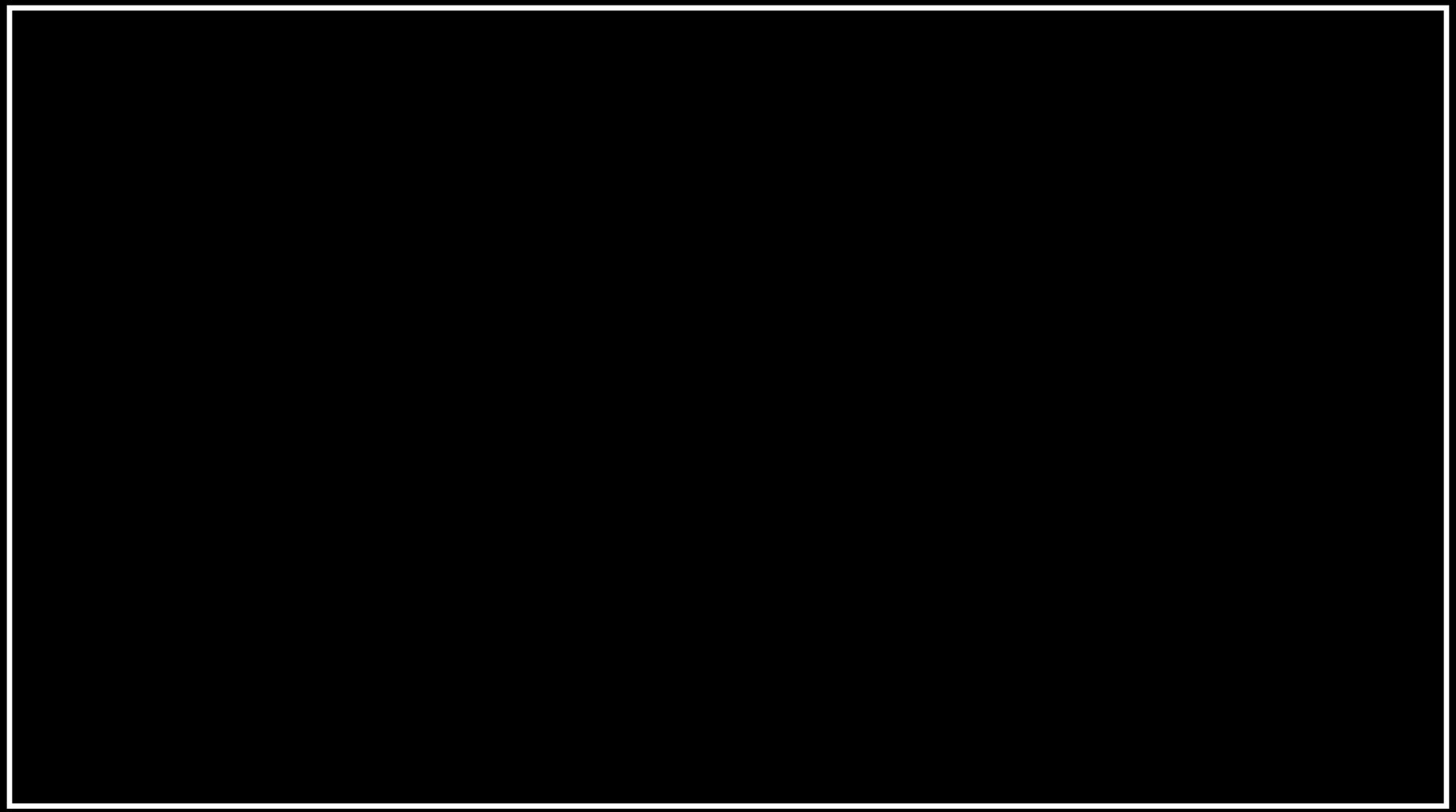
IS IT BETTER?

- Easier to operate with images: e.g. Multiresolution
- You can optimise per pixel
- Blending only once

- Disconnected pixels problem - creating an indirection map
- Quality depends on the parameterization

TEXTURE IS NOT ENOUGH

THE PARTHENON



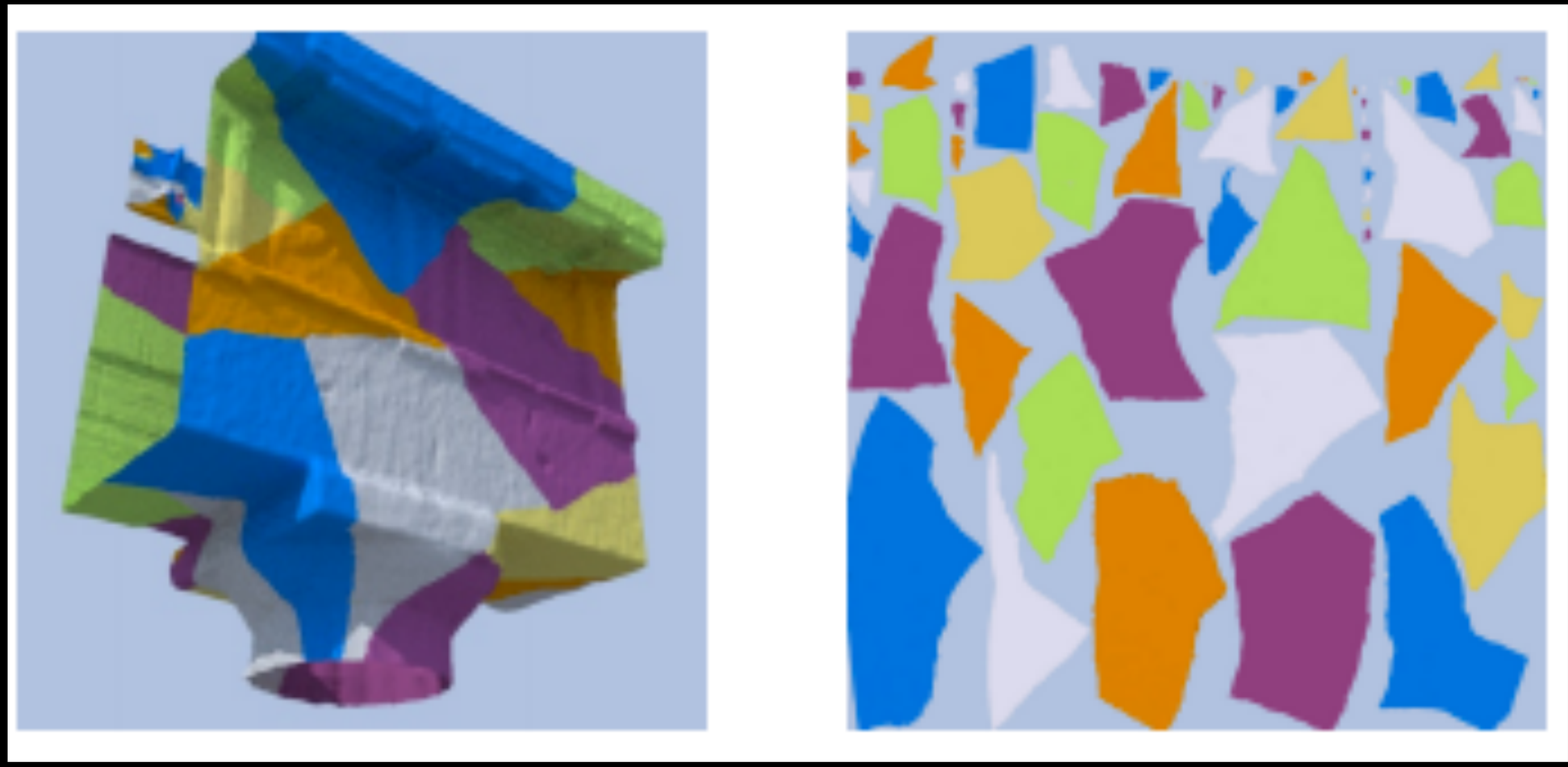
TEXT

THE PARTHENON

Fig. 1: (a) One of eight input photographs (b) Estimated reflectance properties (c) Synthetic rendering with novel lighting

THE PARTHENON

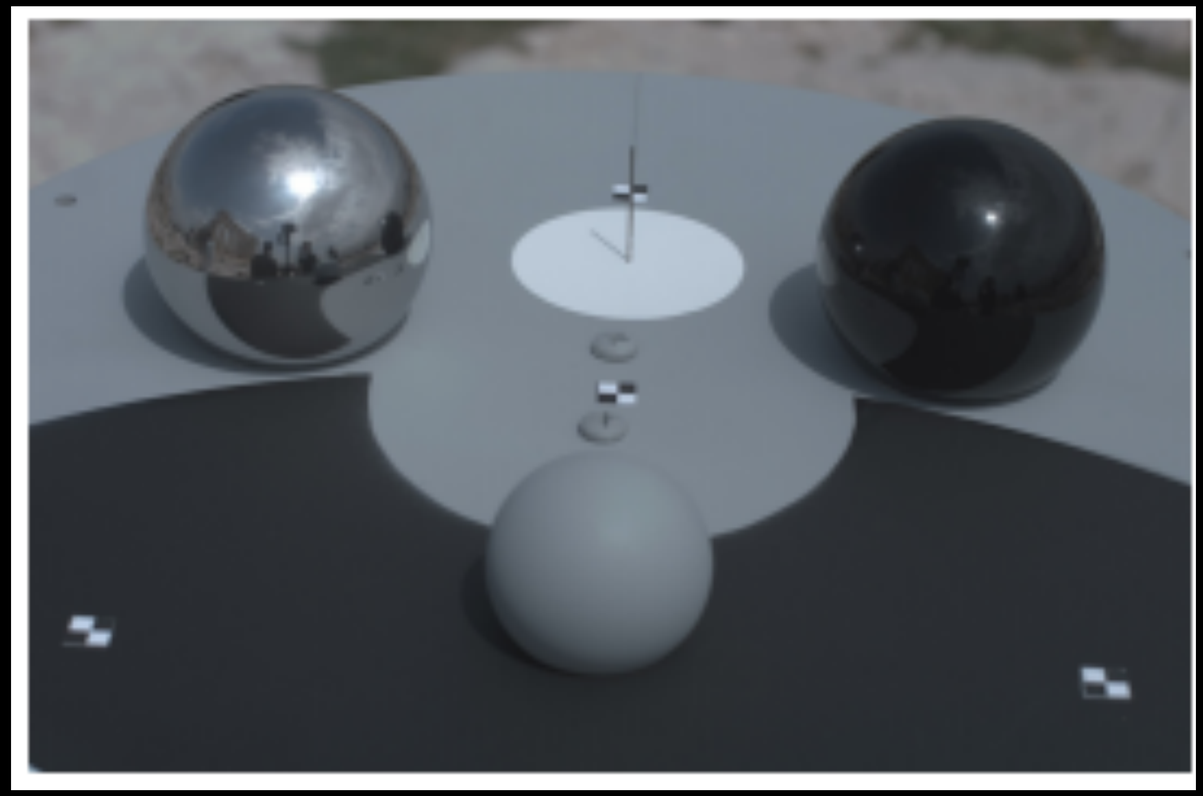
THE PARTHENON



THE PARTHENON

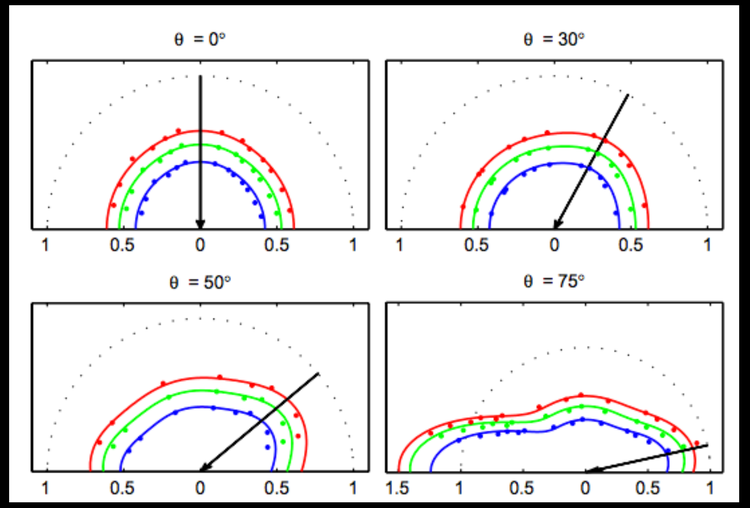
CAPTURING LIGHT

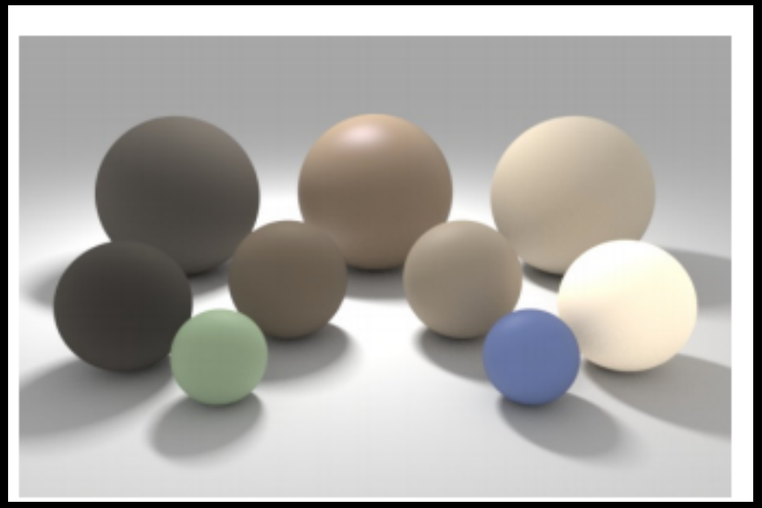
- Many photographs for texturing
- Synchronised lighting capture
- Camera matching
- Synthetic rendering and comparison



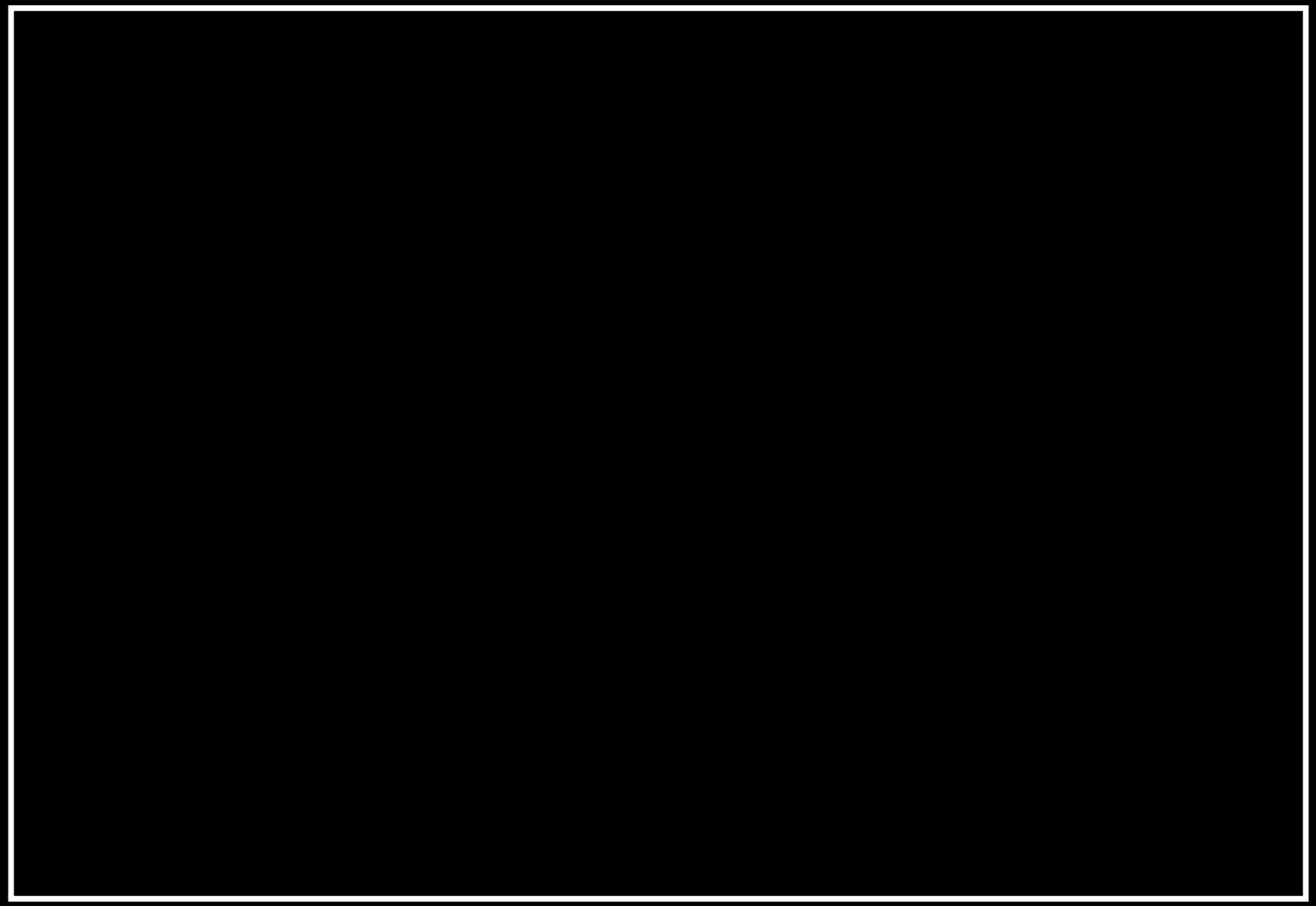
TEXT

CAPTURING BRDFS





THE PARTHENON



TEXT

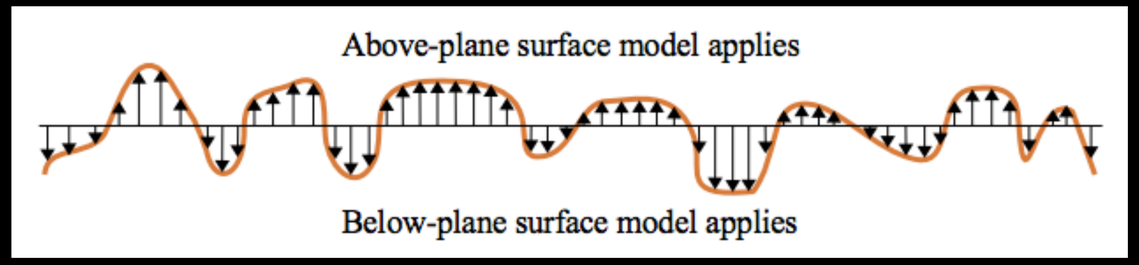
SURFACE DEPTH HALLUCINATION

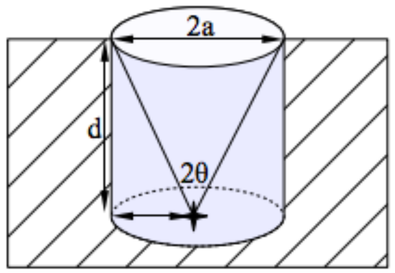
FLASH AND NO-FLASH PICTURES

SURFACE DEPTH HALLUCINATION

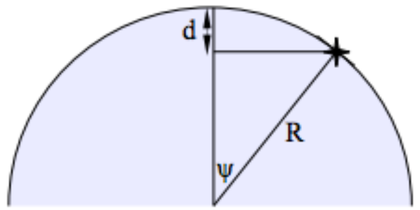
Albedo map + Shading Image

SURFACE DEPTH HALLUCINATION

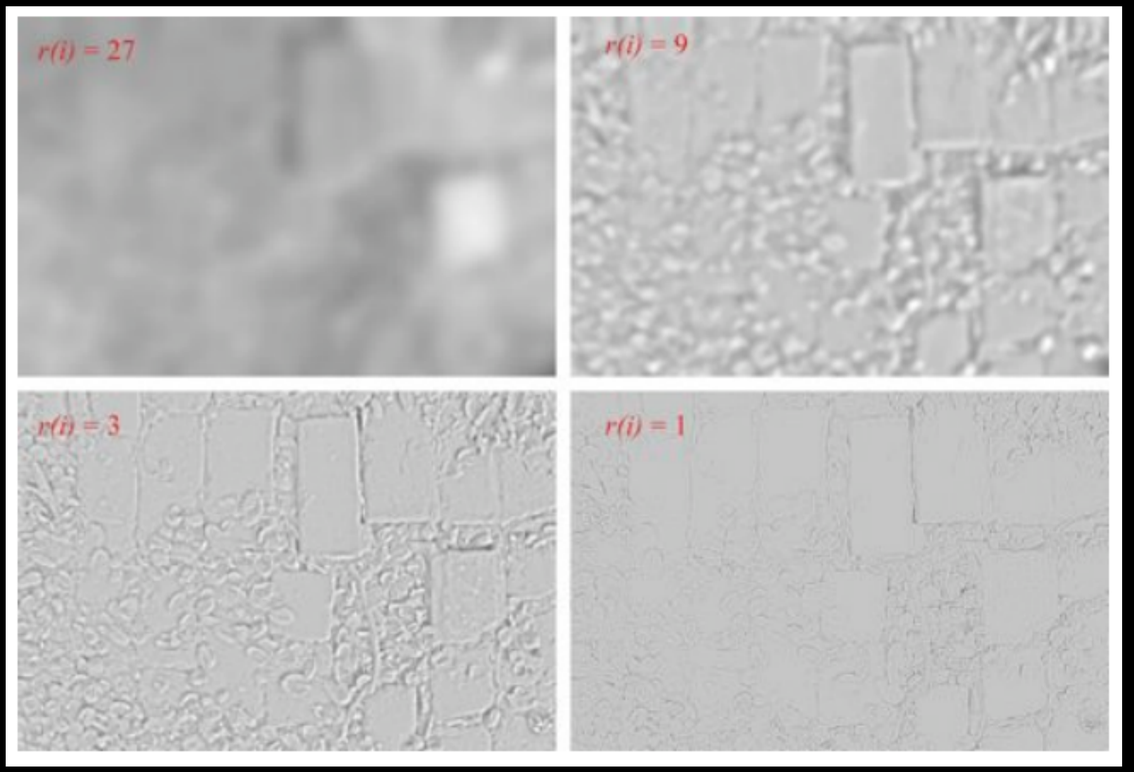




(a) Cylinder pit model.

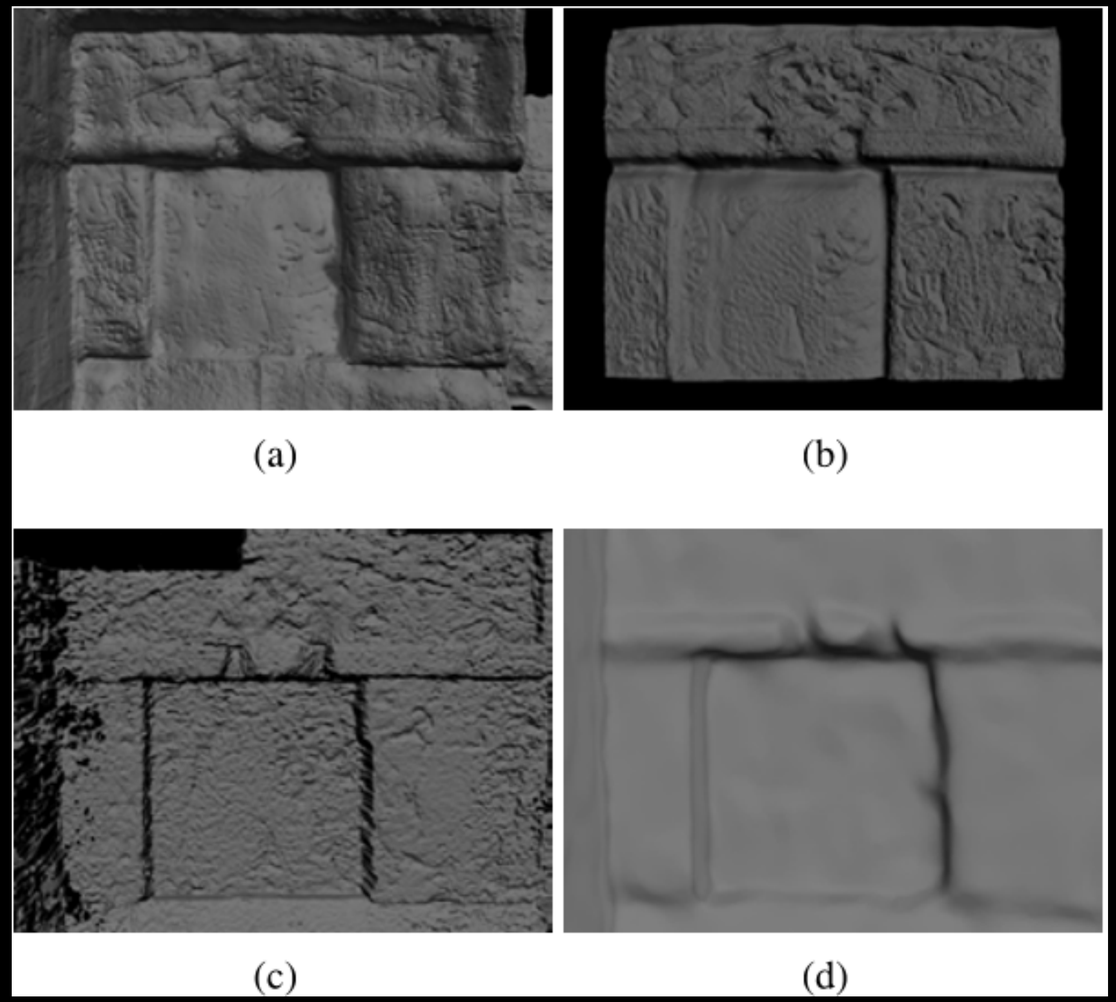


(b) Hemisphere protrusion model.

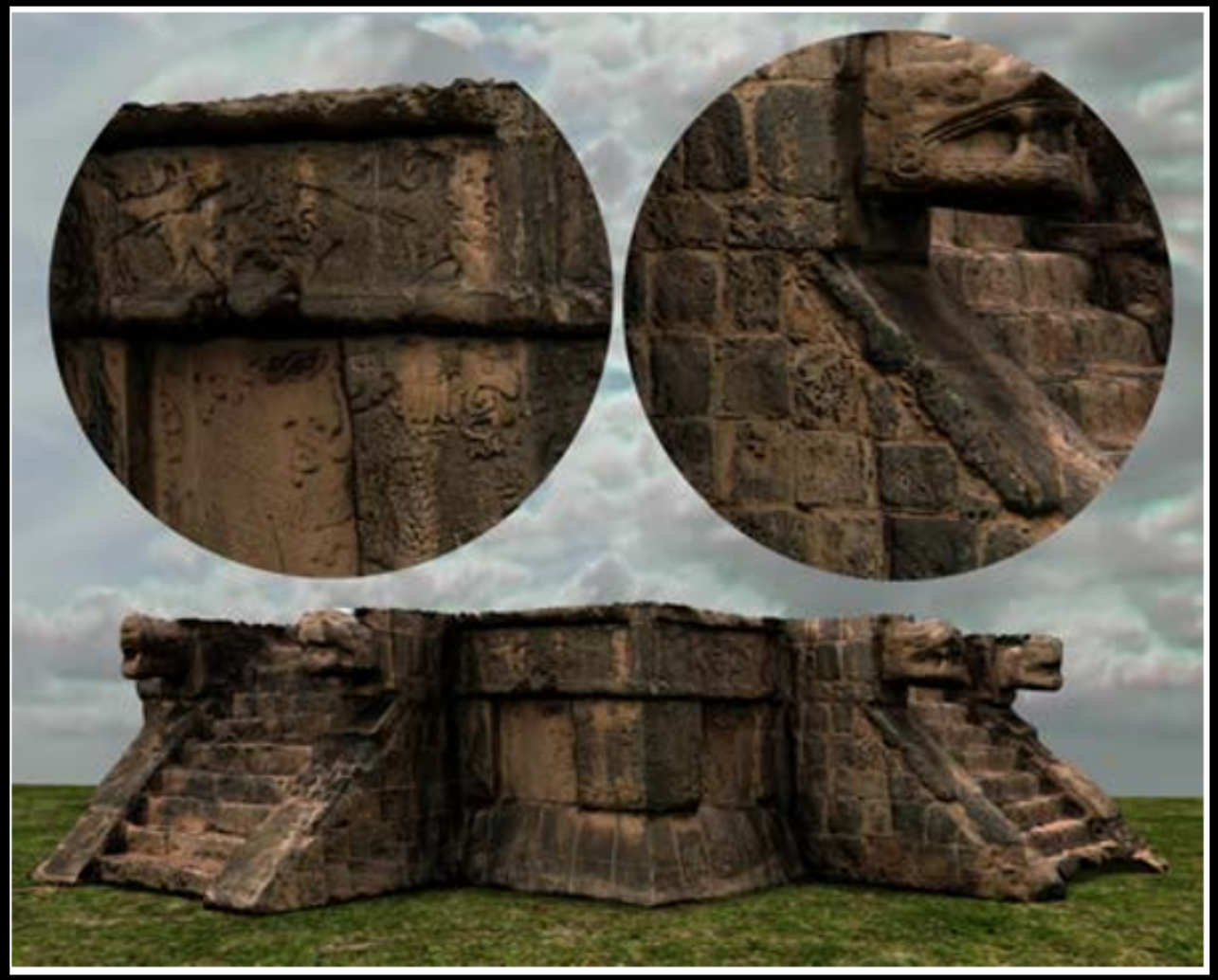


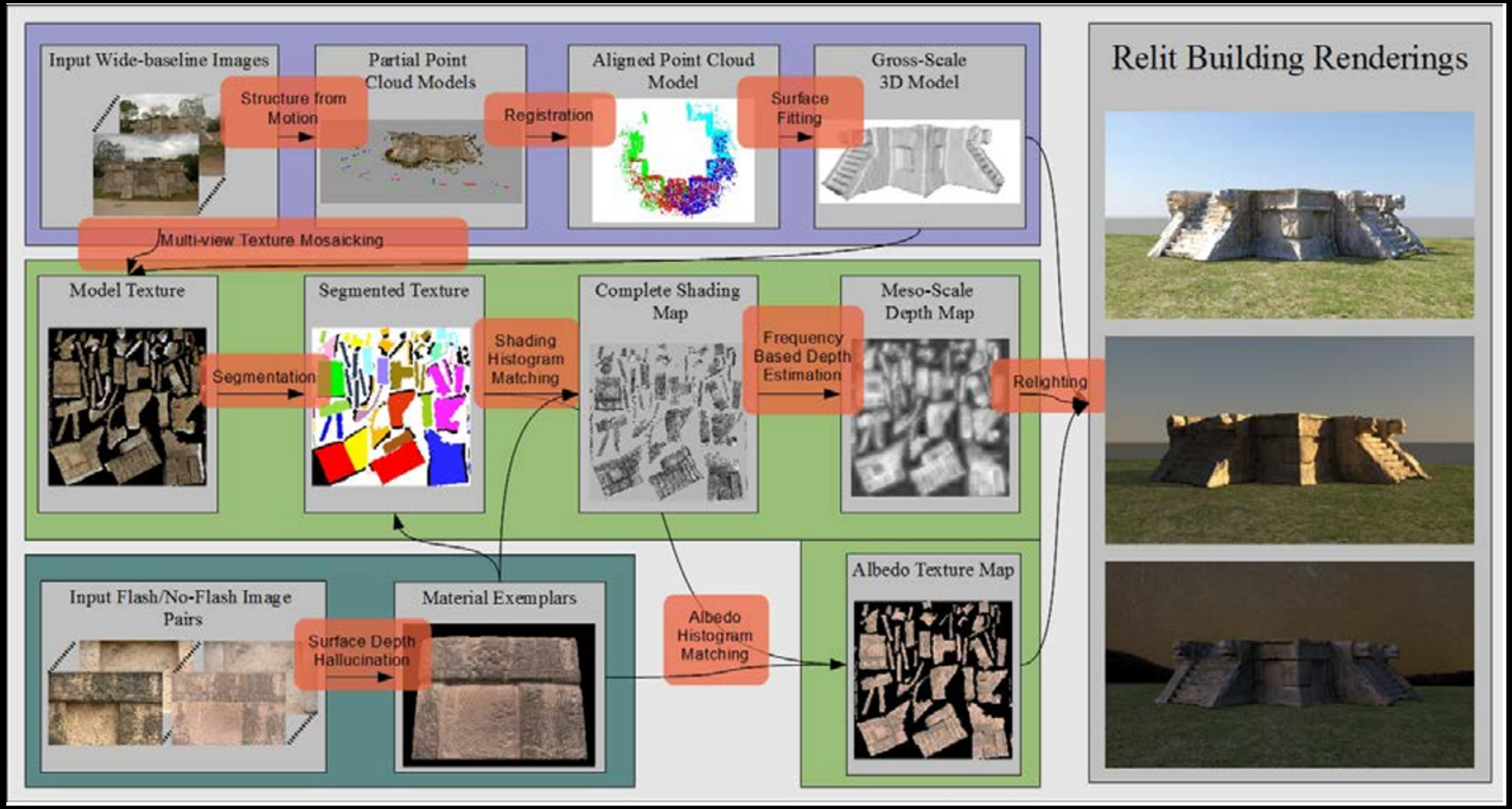
SURFACE DEPTH HALLUCINATION

SURFACE DEPTH HALLUCINATION



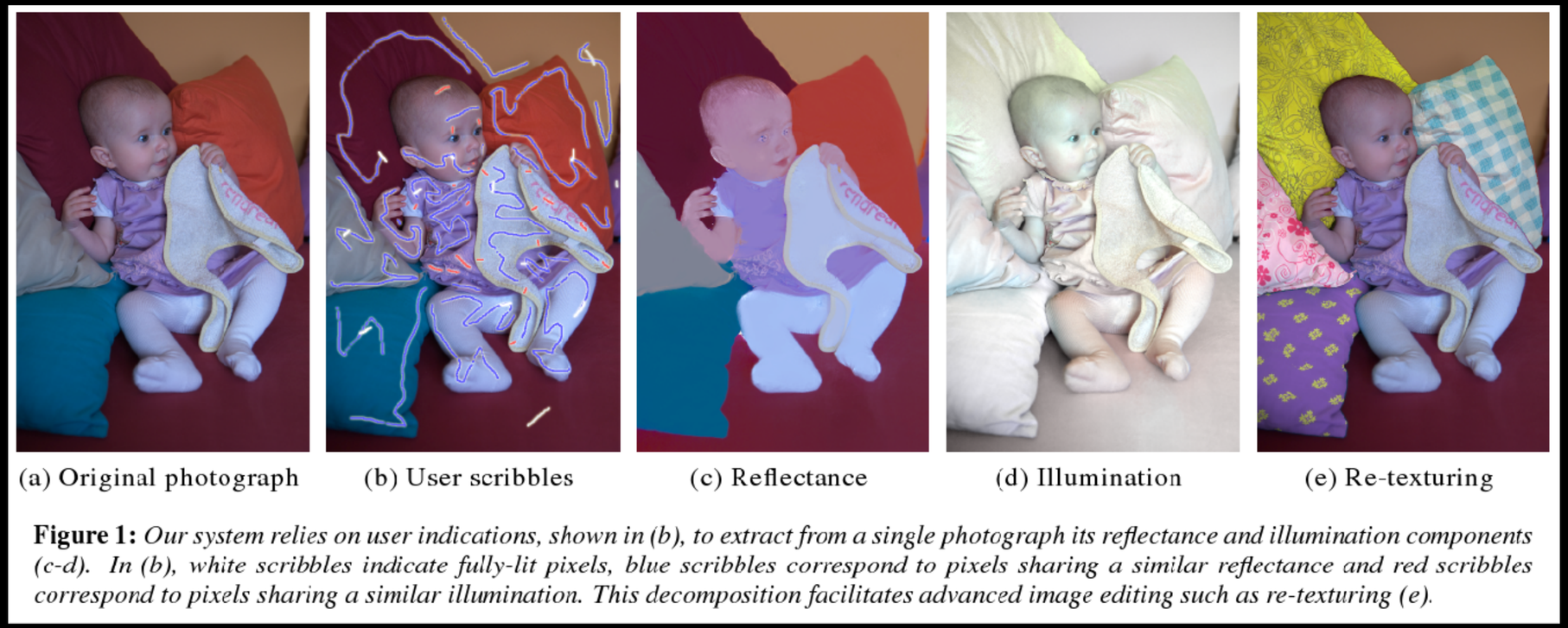
SURFACE DEPTH HALLUCINATION




Input Wide-baseline Images
Structure from Motion
Partial Point Cloud Models
Registration
Aligned Point Cloud Model
Surface Fitting
Gross-Scale 3D Model
Multi-view Texture Mosaicking
Model Texture
Segmentation
Segmented Texture
Shading Histogram Matching
Complete Shading Map
Frequency Based Depth Estimation
Meso-Scale Depth Map
Relighting
Input Flash/No-Flash Image Pairs
Surface Depth Hallucination
Material Exemplars
Albedo Histogram Matching
Albedo Texture Map
Relit Building Renderings

INTRINSIC IMAGES

INTRINSIC IMAGES

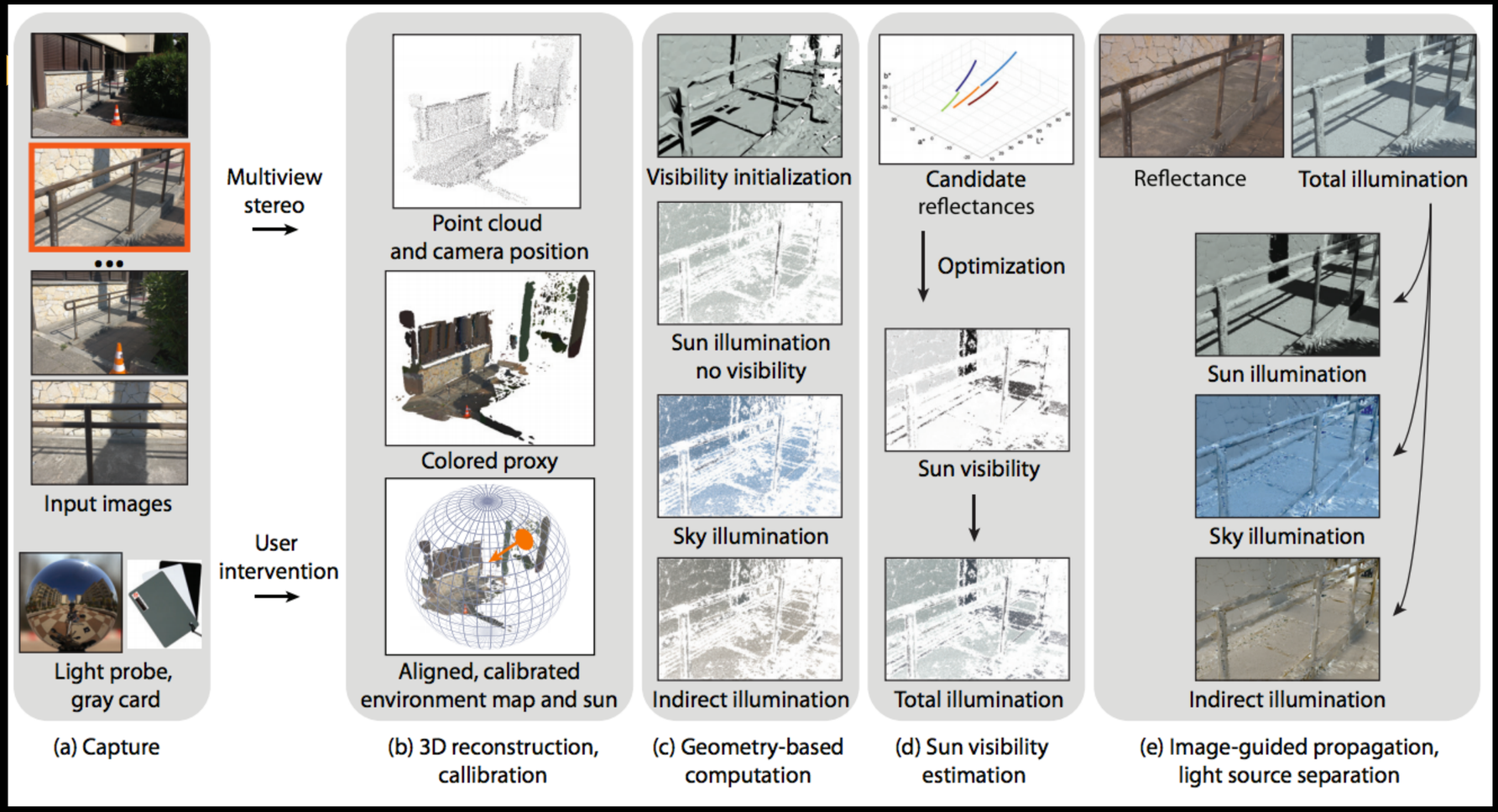


(a) Original photograph (b) User scribbles (c) Reflectance (d) Illumination (e) Re-texturing

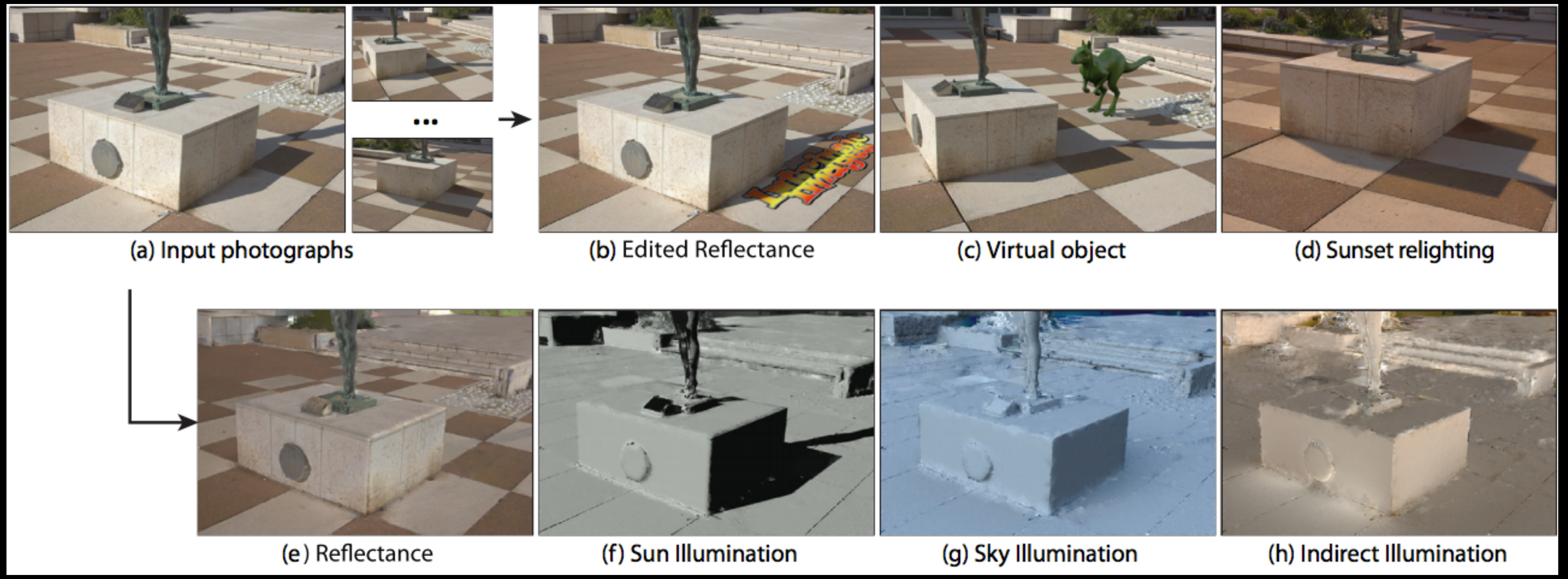
Figure 1: *Our system relies on user indications, shown in (b), to extract from a single photograph its reflectance and illumination components (c-d). In (b), white scribbles indicate fully-lit pixels, blue scribbles correspond to pixels sharing a similar reflectance and red scribbles correspond to pixels sharing a similar illumination. This decomposition facilitates advanced image editing such as re-texturing (e).*

User Assisted Intrinsic Images (2009)
Adrien Bousseau, Sylvain Paris, Frédo Durand

RICH INTRINSIC IMAGE DECOMPOSITION



RICH INTRINSIC IMAGE DECOMPOSITION



RICH INTRINSIC IMAGE DECOMPOSITION

TEXT

RICH INTRINSIC IMAGE DECOMPOSITION

- lafont 1

(a) Input photograph

(b) Illumination constraints

(c) Estimated total illumination

(d) Estimated reflectance

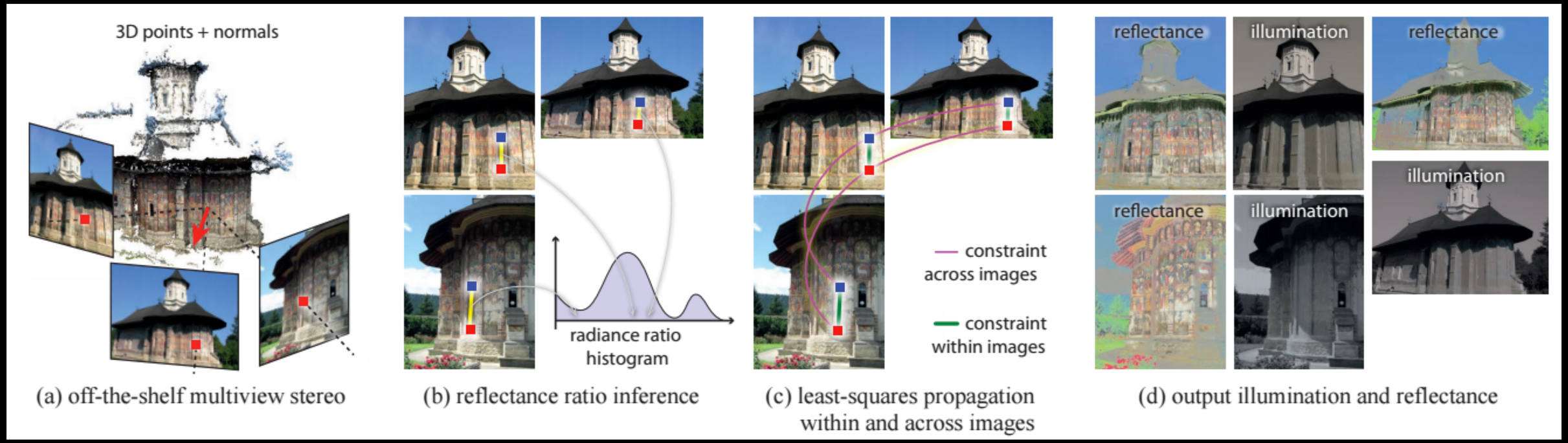
TEXT

COHERENT INTRINSIC IMAGES FROM PHOTO COLLECTIONS

Figure 1: *Our method leverages the heterogeneity of photo collections to automatically decompose photographs of a scene into reflectance and illumination layers. The extracted reflectance layers are coherent across all views, while the illumination captures the shading and shadow variations proper to each picture. Here we show the decomposition of three photos in the collection.*

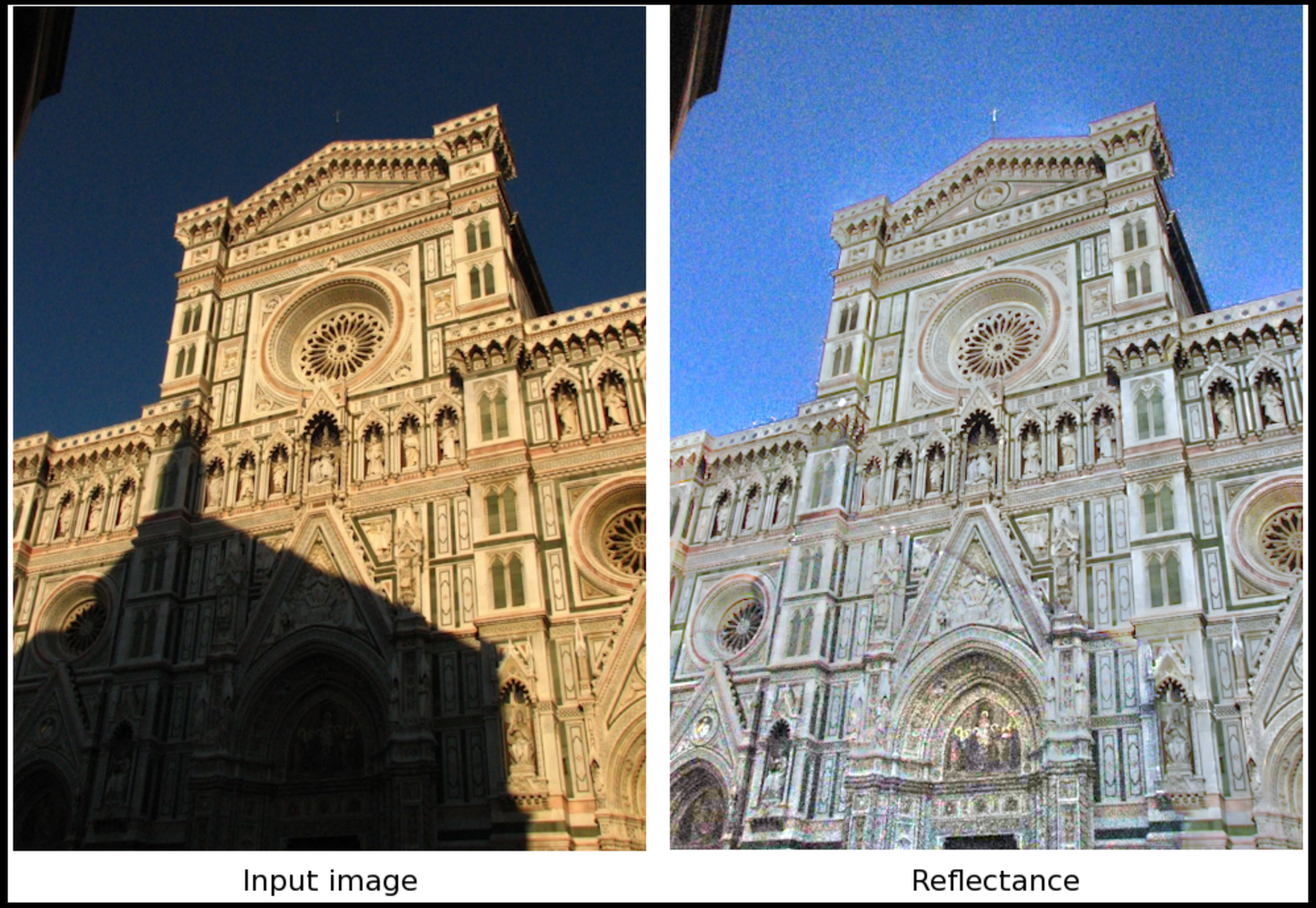
TEXT

COHERENT INTRINSIC IMAGES FROM PHOTO COLLECTIONS



(a) off-the-shelf multiview stereo (b) reflectance ratio inference (c) least-squares propagation within and across images (d) output illumination and reflectance

COHERENT INTRINSIC IMAGES FROM PHOTO COLLECTIONS



Input image

Reflectance

REFERENCES

- http://www.gcc.tu-darmstadt.de/home/proj/texrecon/
- https://members.loria.fr/Bruno.Levy/papers/LSCM_SIGGRAPH_2002.pdf
- http://www-sop.inria.fr/reves/Basilic/2012/LBPDD12/
- http://gl.ict.usc.edu/Research/reflectance/