3D Digitization for Cultural Heritage

MARCO CALLIERI

VISUAL COMPUTING LAB

ISTI-CNR PISA, ITALY

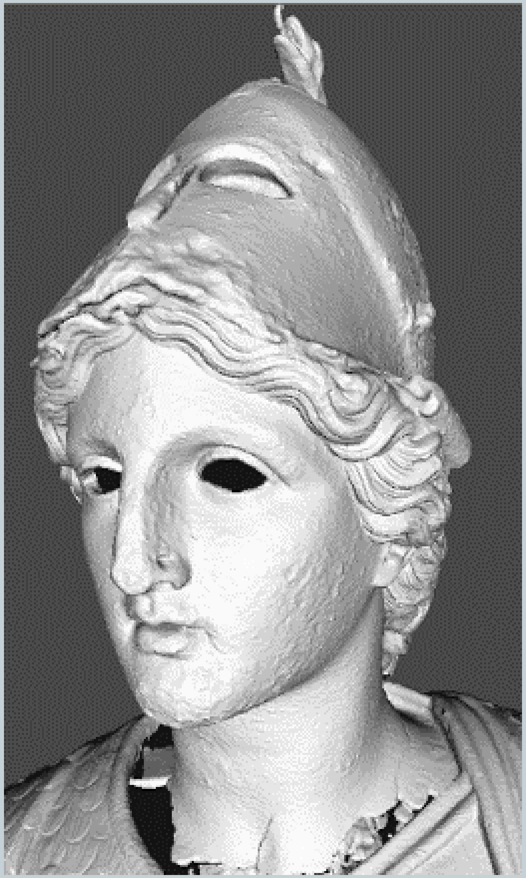
The importance of color information

Precision vs. Perception

3D scanned geometry

Photo

Color and appearance



Pure geometry

"Pure" color

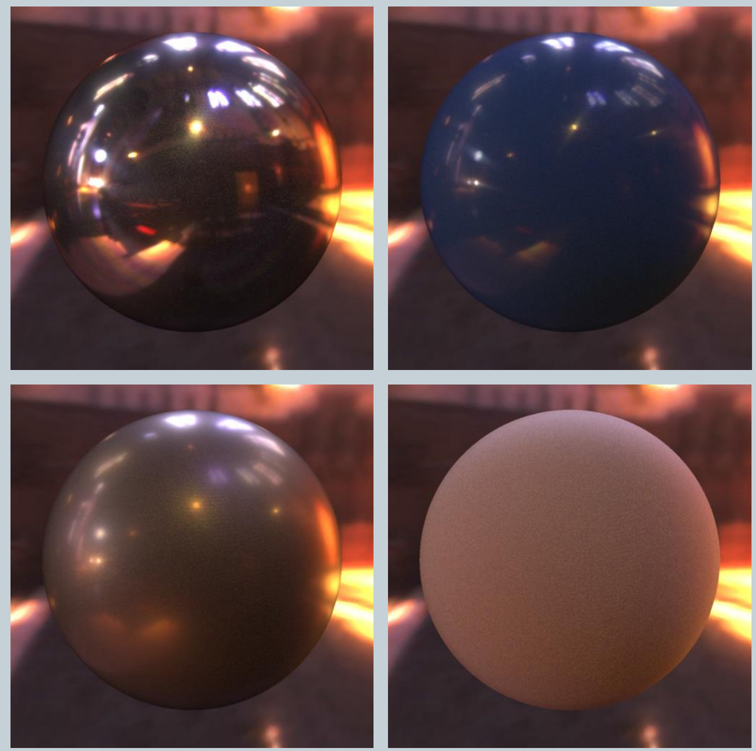
Rendering of material properties

Visual Appearance

Color due to the interaction between the lighting environment (intensity, position, …) and the properties of the object surface and material.

LIGHT

MATERIAL

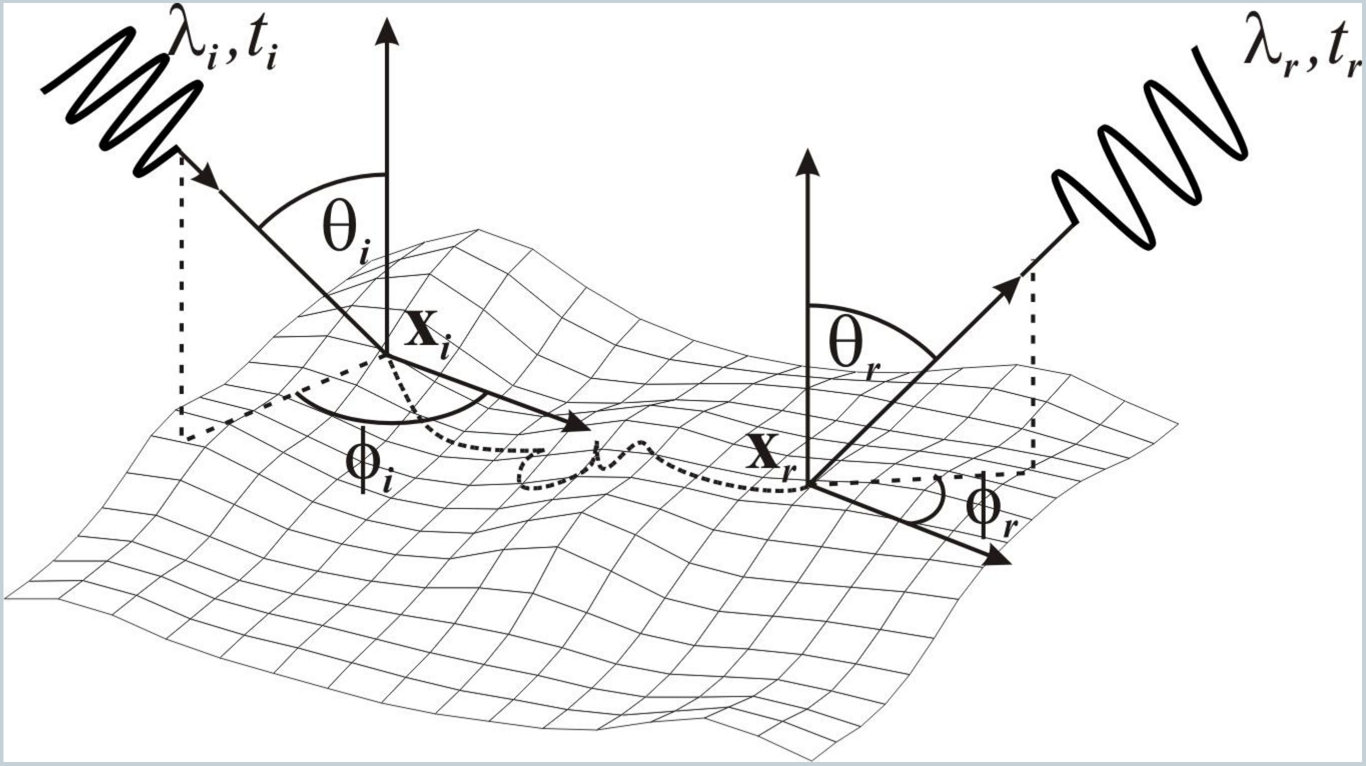


Visual Appearance: Definition

Reflectance Scattering Function (12D)

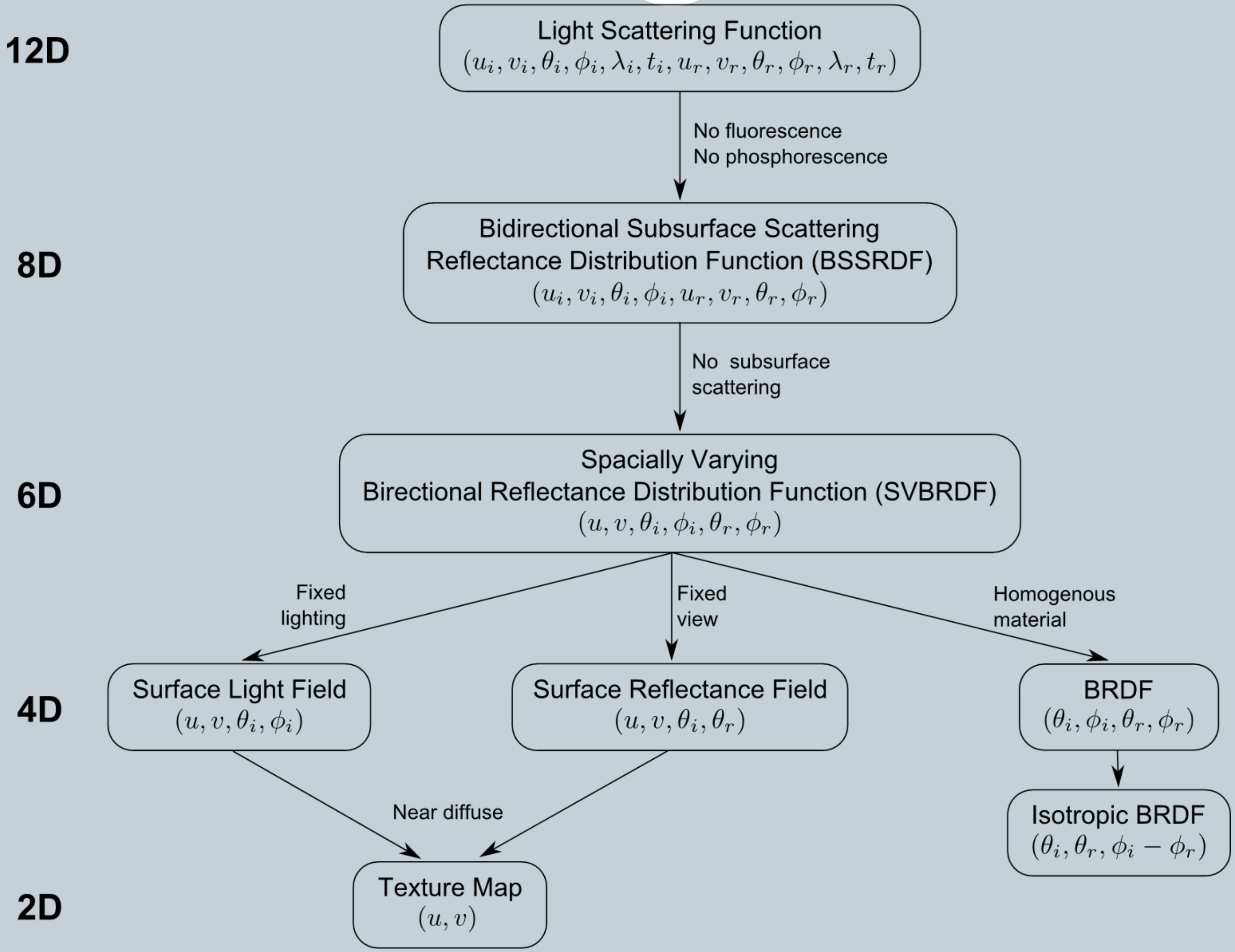
(Light and view direction, Incident and outgoing surface point

Wavelength, Time)



And this is just for NON-transparent materials ……

Visual Appearance: Definition

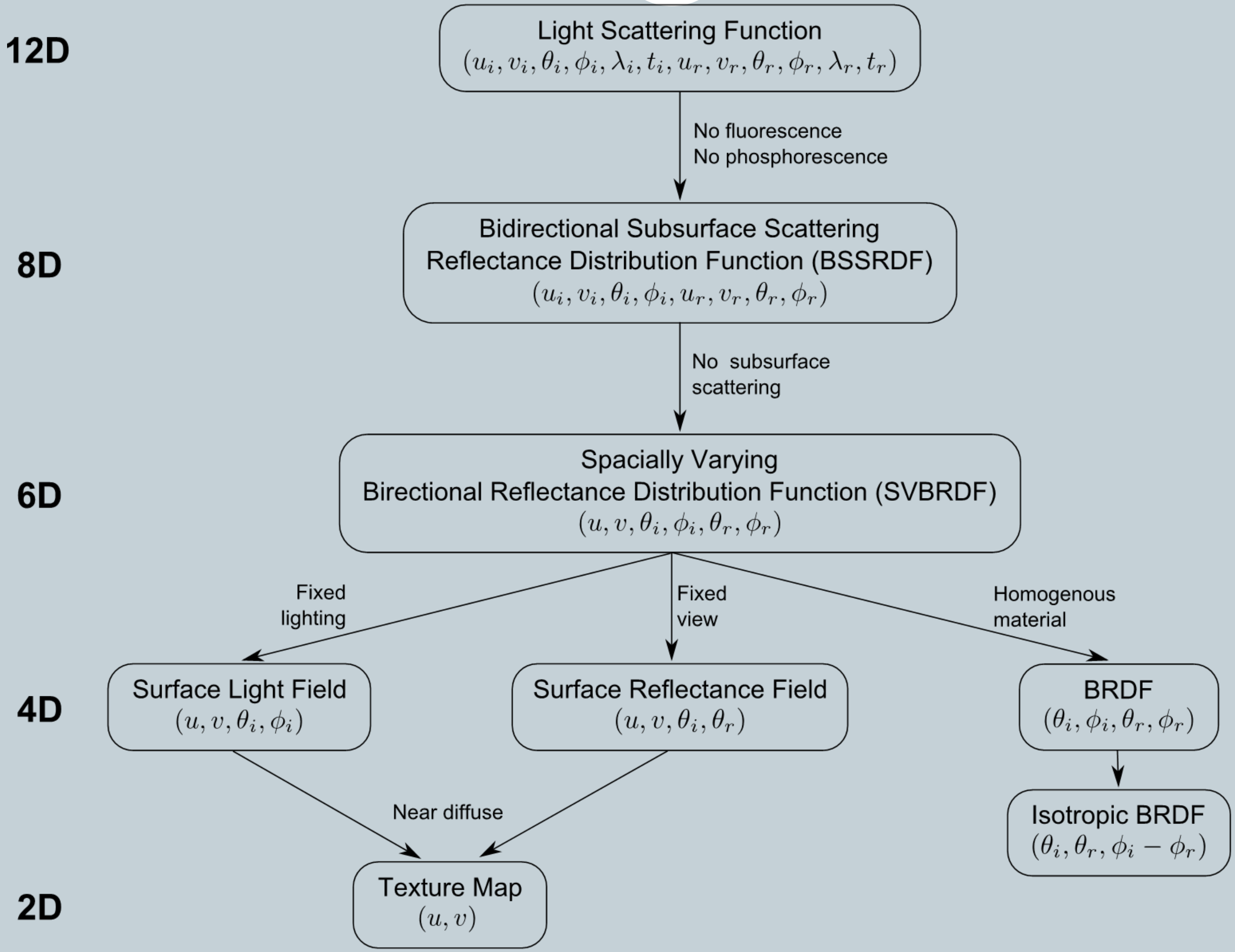


Visual Appearance: Definition

Reflectance Scattering Function (12D)

- No mathematical formulation
- Measurement impractical
- Simplification by constrains on the set of possible reflectance effects
  - Phosphorescence
  - Fluorescence
  - Subsurface scattering
  - Specular scattering
  - Backscattering
  - Diffuse scattering

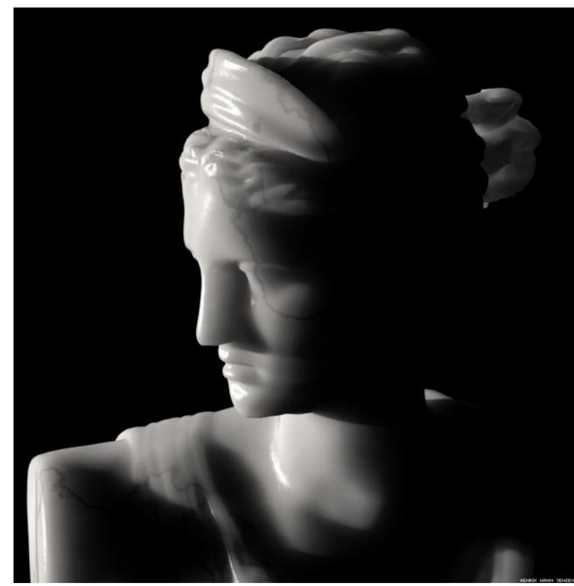
Visual Appearance: Definition



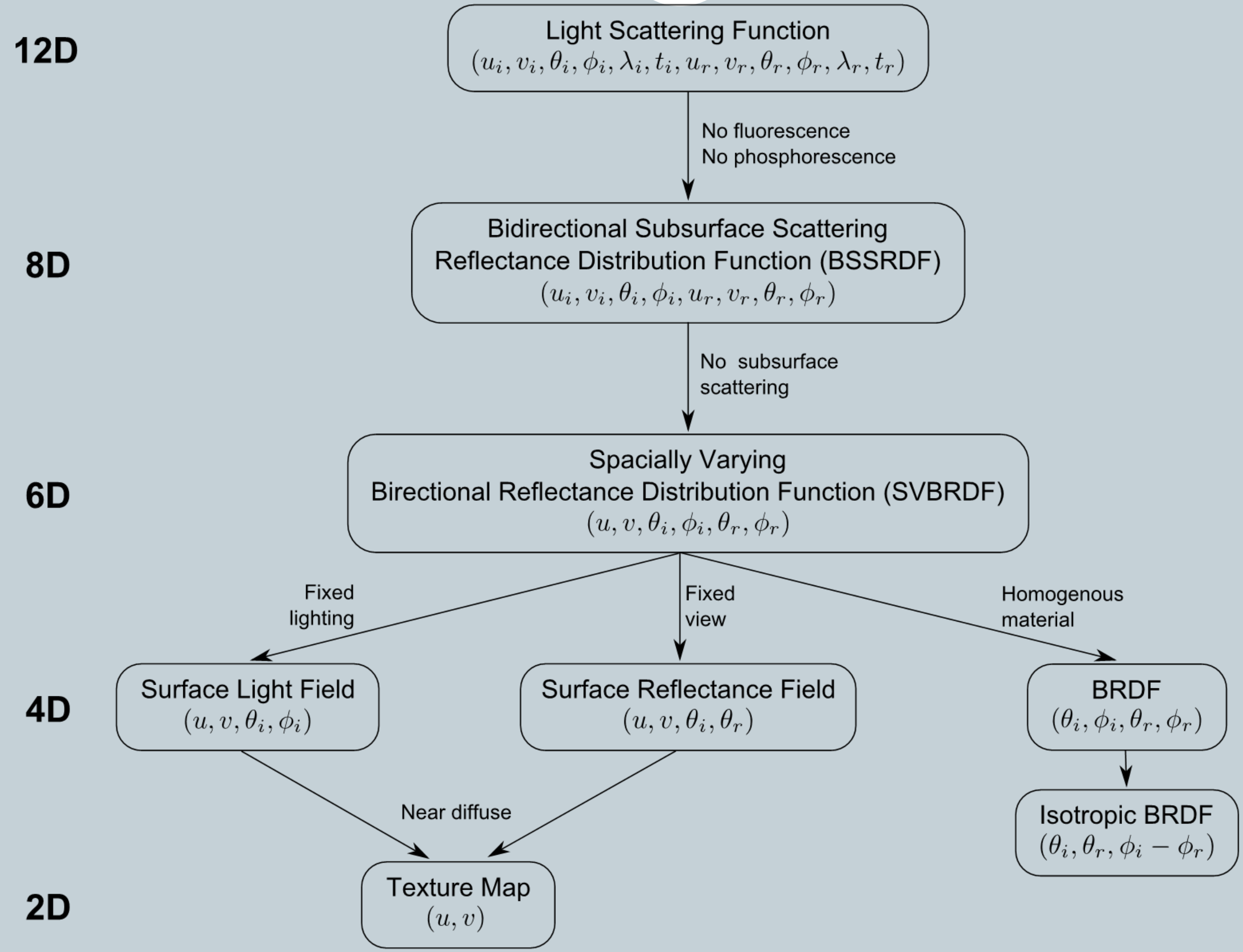
Visual Appearance

BSSRDF (8D)

- No fluorescence (no wavelength change)
- No Phosphorescence (zero time light transport)
- Subsurface scattering (translucent material)



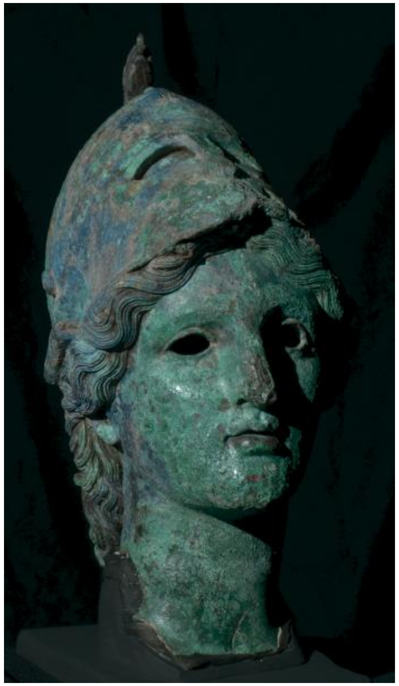
Visual Appearance: Definition



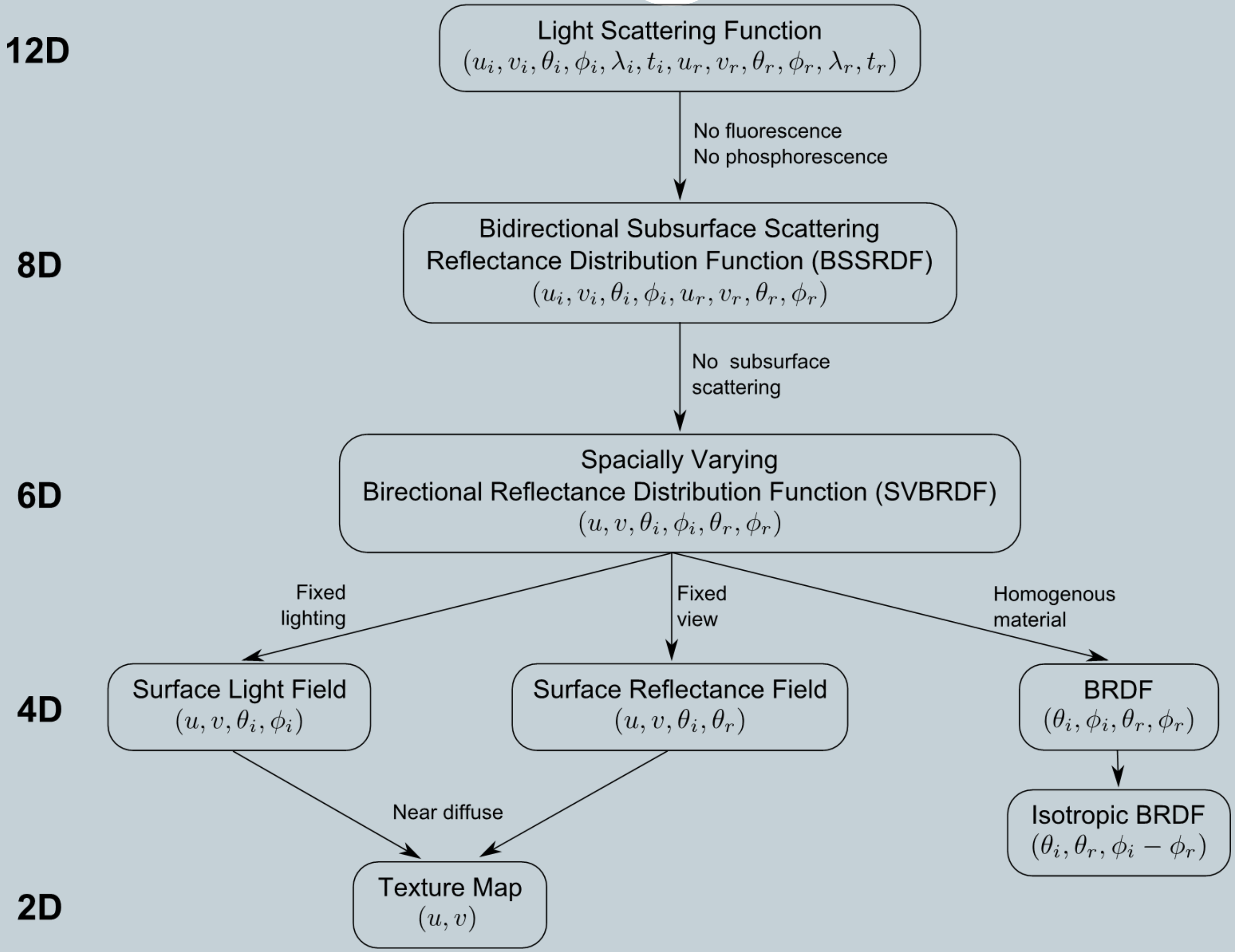
Visual Appearance

SVBRDF (6D)

- No Subsurface scattering (translucent material)
- Opaque material (reflection on the same place)
- Spatially varying glossy material



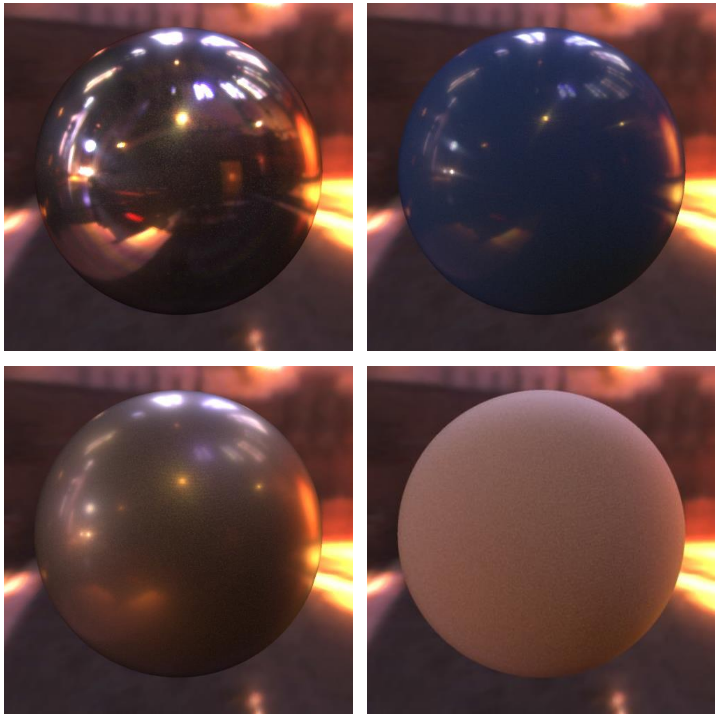
Visual Appearance: Definition



Visual Appearance

BRDF (4D)

- No spatially varying
- Uniform material



Visual Appearance

Light Field (4D)

- Amount of light faring in every direction through every point in space (simplified plenoptic function)
- Fixed lighting condition and variable view direction
- Spatially varying
- Image-based rendering (no geometry)

Surface Reflectance Field (4D)

- Fixed view position and variable light direction
- Spatially varying
- Image-based relighting **(RTI)**
- Implicit geometry

Visual Appearance: details

BSSRDF and BRDF

Model-based rendering

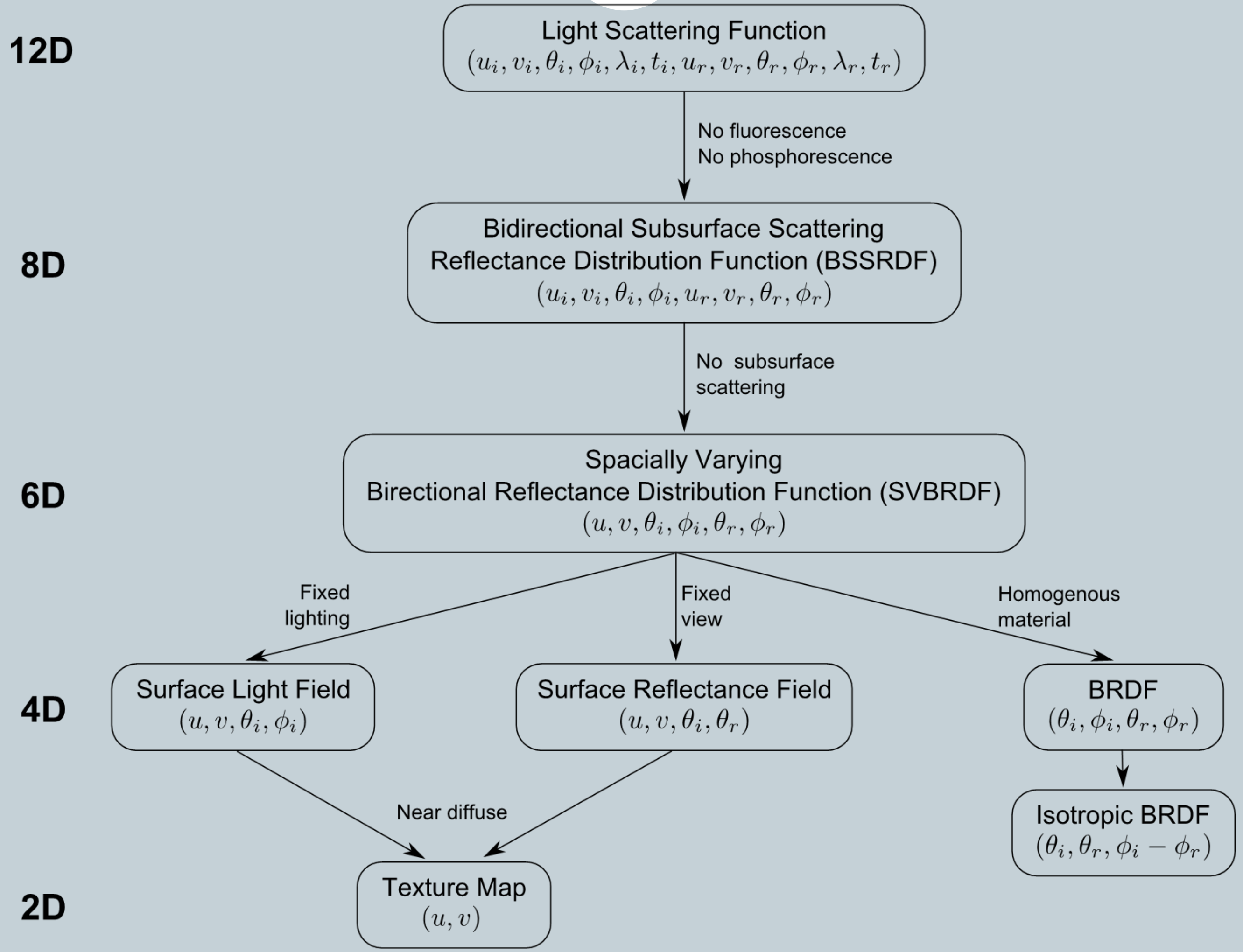
- Explicit geometry
- Modeling or acquisition of the appearance
- Global illumination algorithm
- More precise but computational heavy

Light Field and Reflectance Field

Image based rendering

- Set of photos (“interpolation”)
- No geometry or “implicit” geometry
- Realistic rendering but trade-off between data and precision

Visual Appearance: Definition



Acquisition of reflection properties

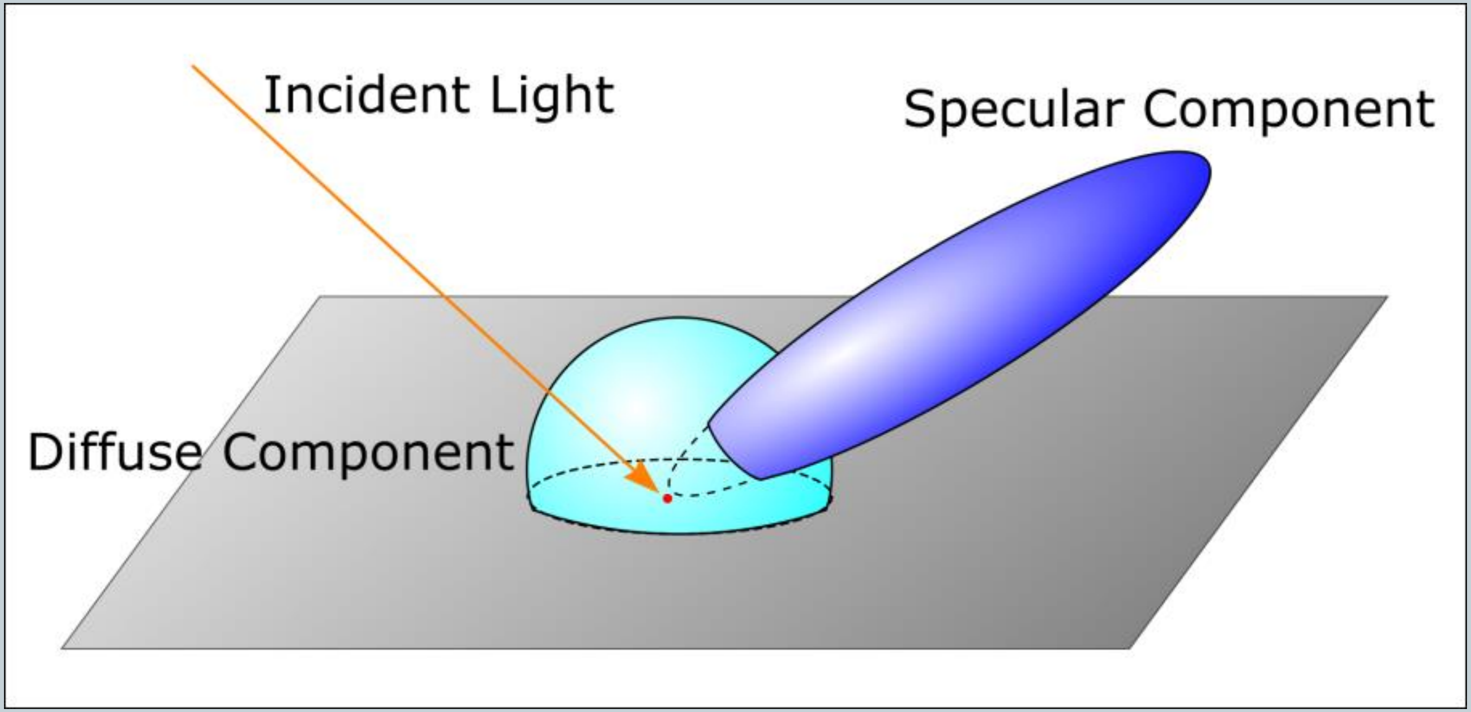
Various degrees of realism:

- **Apparent color**
  - no lighting effects, no moving highlights
- **Albedo**
  - Assume Lambertian material, removal of shading & highlights
- **Spatially varying reflection properties** (Bidirectional Reflection Distribution Function, **BRDF**)
  - relightable representation of the real object interaction with light

Image by MPI (Lensch, Goesele)

BRDF

BRDF(θi,Φi,θo,Φo,u,v)



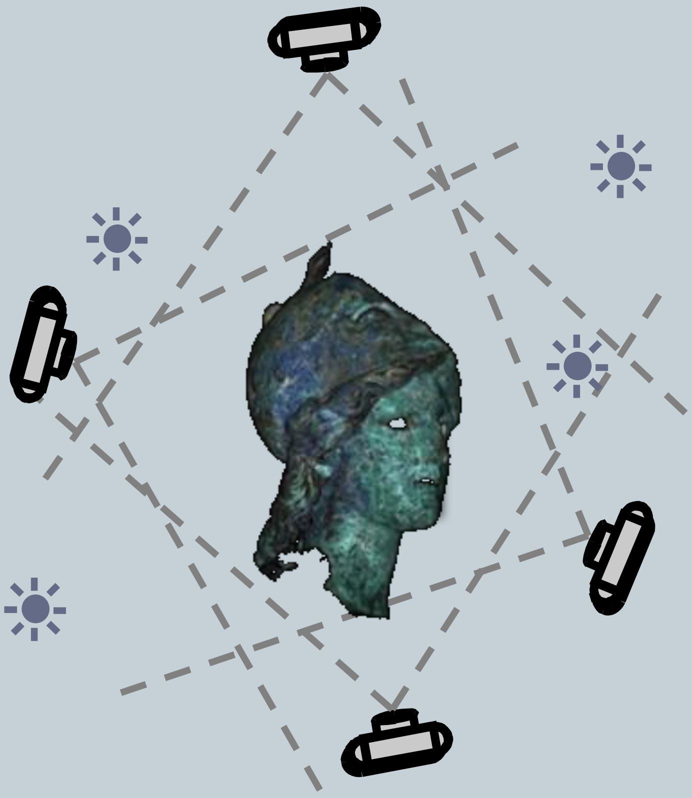
BRDF acquisition

Appearance Acquisition Pipeline:

- 3D Scan of the artifact → 3D model
- Take high-dynamic range images under controlled light conditions
- Align images to 3D geometry
- Process data to estimate per-pixel reflection properties (Estimation of BRDF parameters)

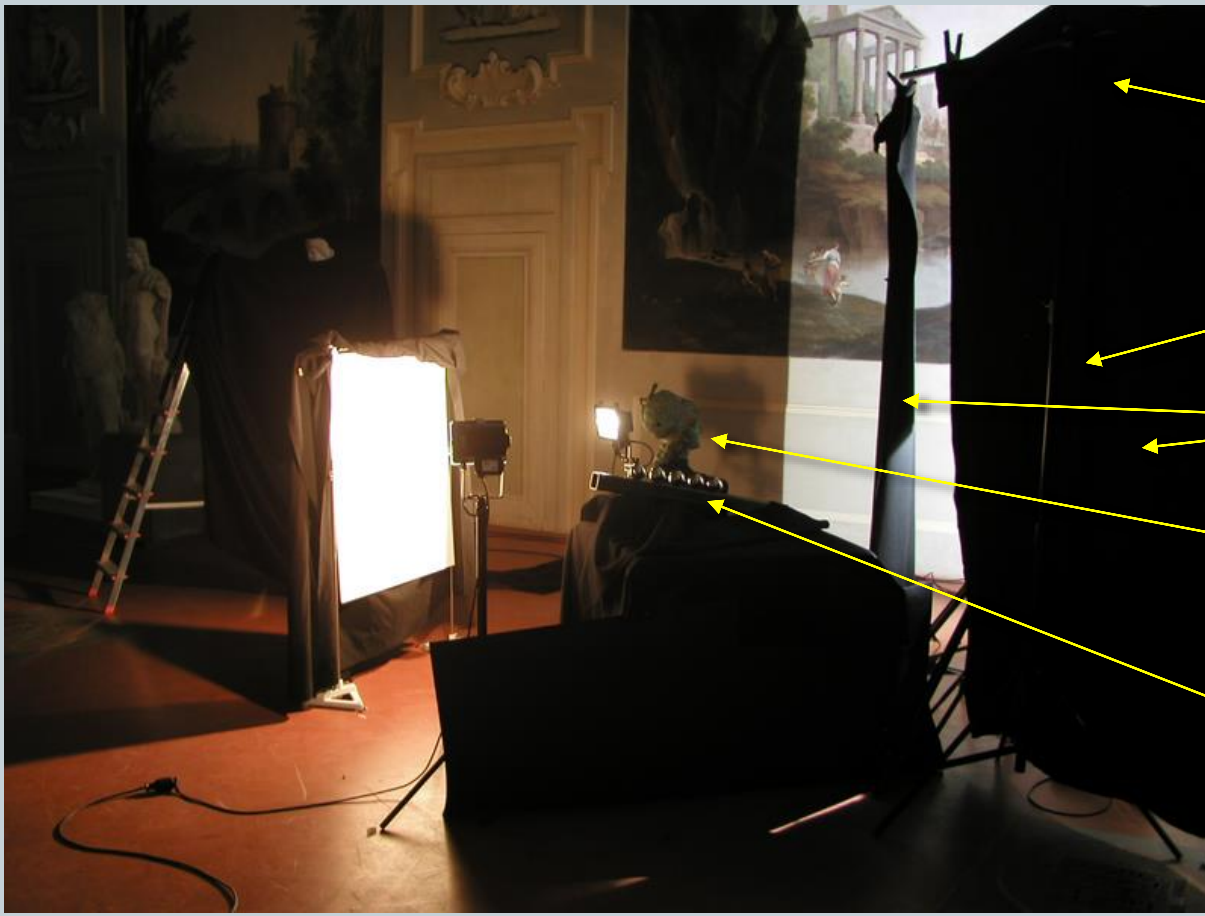
Requires:

- Controlled lighting environment
- Acquisition & processing of many input images (1,000..10,000)



BRDF acquisition

On-site acquisition setup



Issues for practical usage of BRDF methods

- **Flexibility**
  - Deliver an acquisition gantry to a museum? Travel by plane?
  - How to cope with large artifacts?
- **Time required**
  - several hundreds images to acquire & process
- **Controlled lighting conditions?**
  - Often impossible in field conditions (museums, archeological sites)

Further research needed to make acquisition of BRDF (or approximations) **practical** & **usable** in CH applications

Issues for practical usage

Real acquisition conditions:

- Scanning Greek statues at the National Archeological Museum (Athens, Greece, summer ‘07)
  - Lights interfering with laser detection ==> some noise in the sampled data

    Q: Is it possible to turn off the light just above the statue?

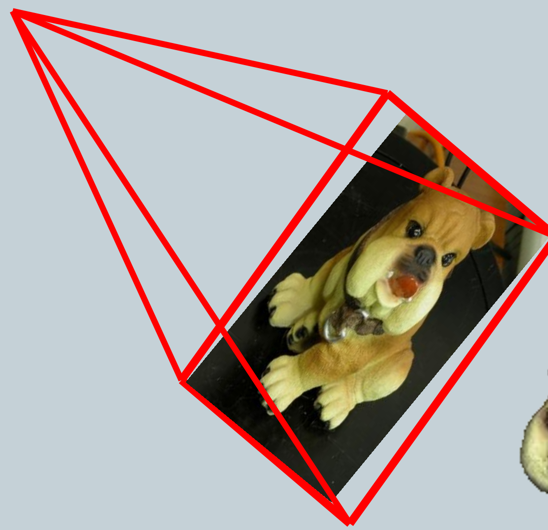
    A: NO! The museum hall has just one single switch
  - Imagine asking for more sophisticated controlled illumination…

An alternative solution: color projection

Alternatively, we can start from a set of photos covering the surface of the object. In a photo, color information is stored according to optical laws of perspective ...

If camera parameters can be recovered, it is possible to project back the information onto the geometry

Simple and effective...

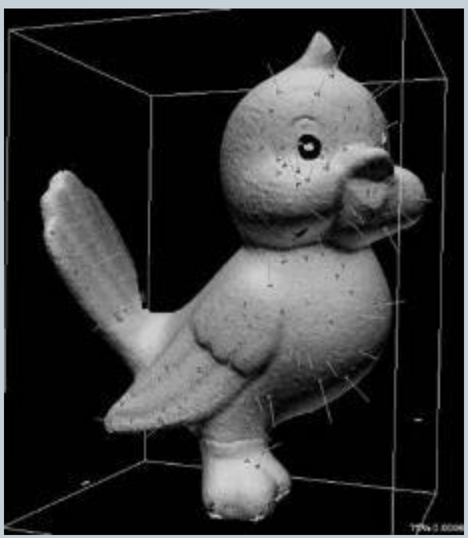


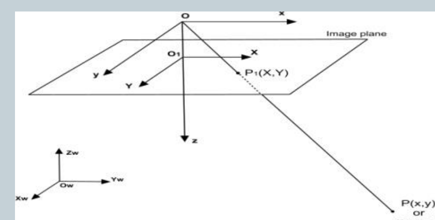
Texture building from photos: Input data

- A complete 3D model
- A set of photos
- Registration info
  (camera data)

→ From scanner

→ (Manual) Registration





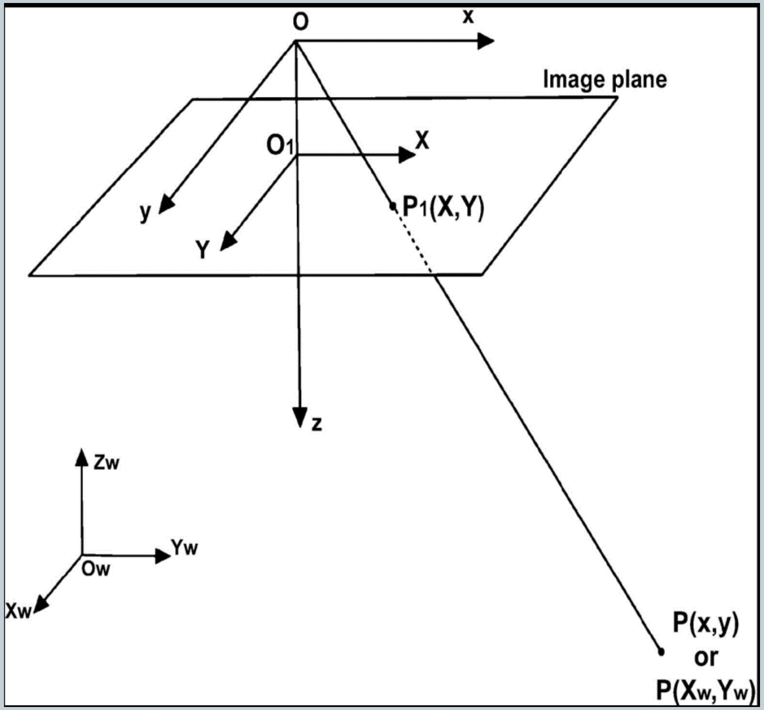
- Position
- Orientation
- Focus distance

……………

Registration info: parameters estimation

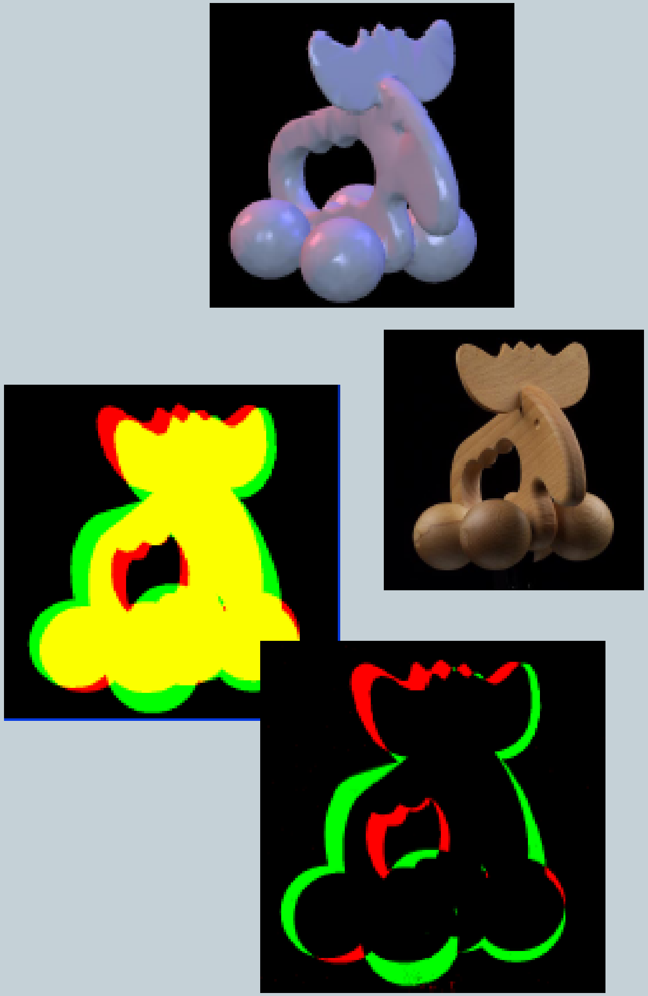
Intrinsic and **extrinsic** parameters

- Extrinsic parameters: rotation matrix and translation vector
- Intrinsic parameters: focal length, lens distortion…



Automatic alignment

- Automatic camera computation via **silhouette matching**
  - Compute silhouette on image
  - Render the 3D model monocrome and compute silhoutette
  - Compute no. pixels covered by just one silhouette (img XOR)
  - Greedy iteration, by small rotations, until silhouette matching error is below a threshold



Limitation: all object visible in each image ==> just small objects!

Lensch et al, "Automated Texture Registration and Stitching for Real World Models", Pacific Graphics '00

Estimation using photogrammetry tools

Photogrammetry tools do estimate the camera parameters and can be used to recover intrinsics and extrinsics to project color on a 3D model. Many tools can as well do the entire texturing process, for small models

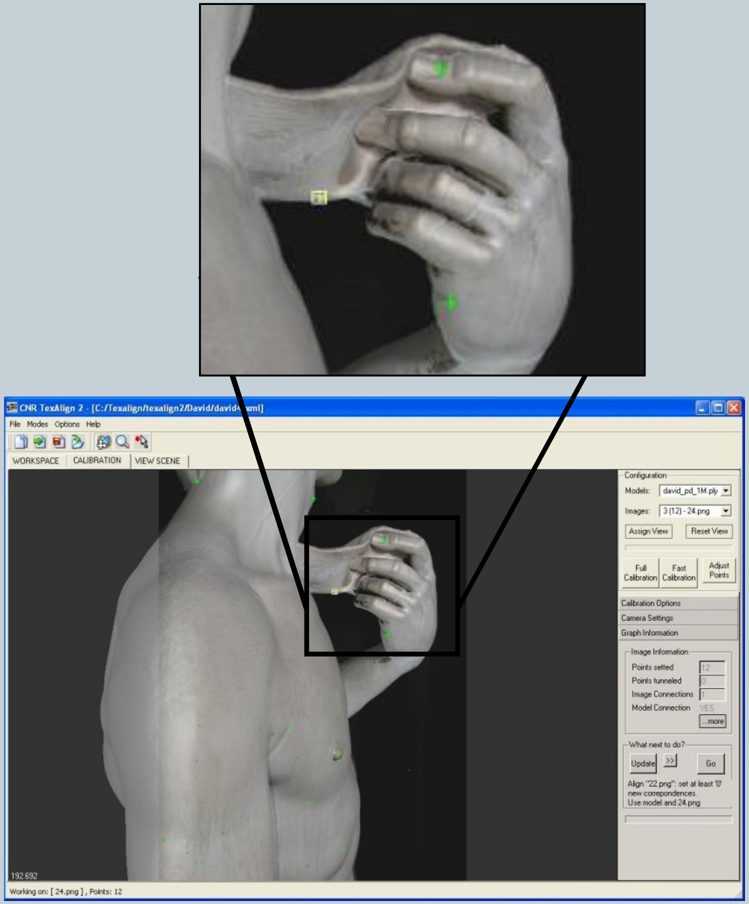
Parameters estimation using correspondences

Parameters estimation:

- Setting of some correspondences between image and geometry
- Minimization of error function

Different algorithms and implementations:
- TSAI (old faithful)
- GARCIA (fast but need good start)
- intel OpenCV (hard to integrate)

Minimizing user intervention in registering 2D images to 3D models
T. Franken, M. Dellepiane, F. Ganovelli, P. Cignoni, C. Montani, R. Scopigno 2005



Automatic alignment using mutual information

- Mutual information is used with geometric features ***correlated in some way*** to the visual appearance of the objects but *invariant* to the lighting environment.

Main idea

- Visual appearance of the renderings of object's geometry and photographic images of the same object is, in general, very different.

Main Idea

- **How to measure similarity?**
  => Mutual information (MI) between such features and the image to register is a very good candidate.
- MI is widely used in medical imaging for the registration of images coming from different sensors, such as magnetic resonance (MR), computerized tomography (CT), PET, x-rays, and so on..
- It is able to catch non-linear correlation.
- The registration problem becomes a maximization problem.

Mutual Information

•MI is the amount of information about B that A contains. Using the joint probability it can be expressed as:

$$\mathcal{I}(I_A, I_B) = -\sum_{(a,b)} \log p(a,b) \frac{p(a,b)}{p(a)p(b)}$$

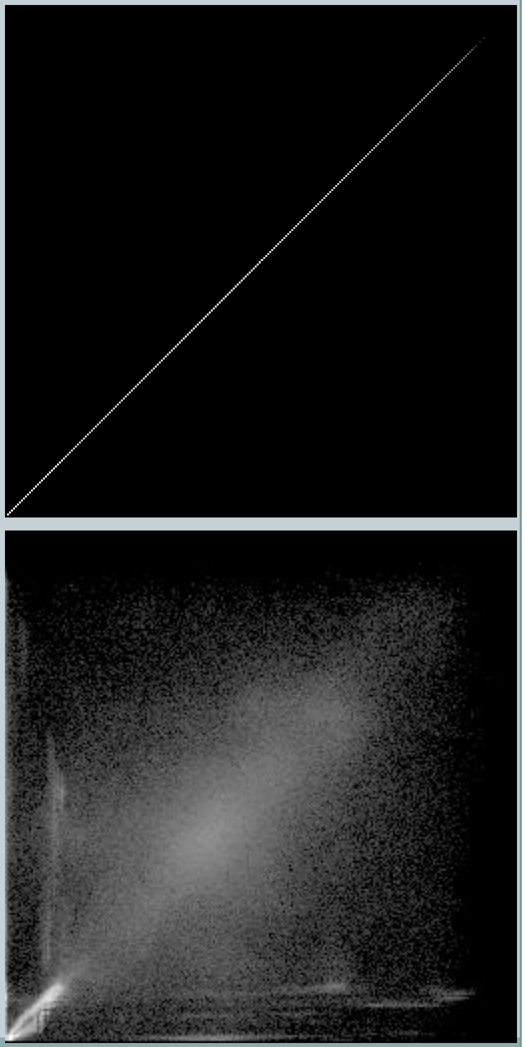
•The joint events are stored in a *joint histogram*

•The probability of the joint event p(a,b) is the number of occurrences divided the total number of pixels

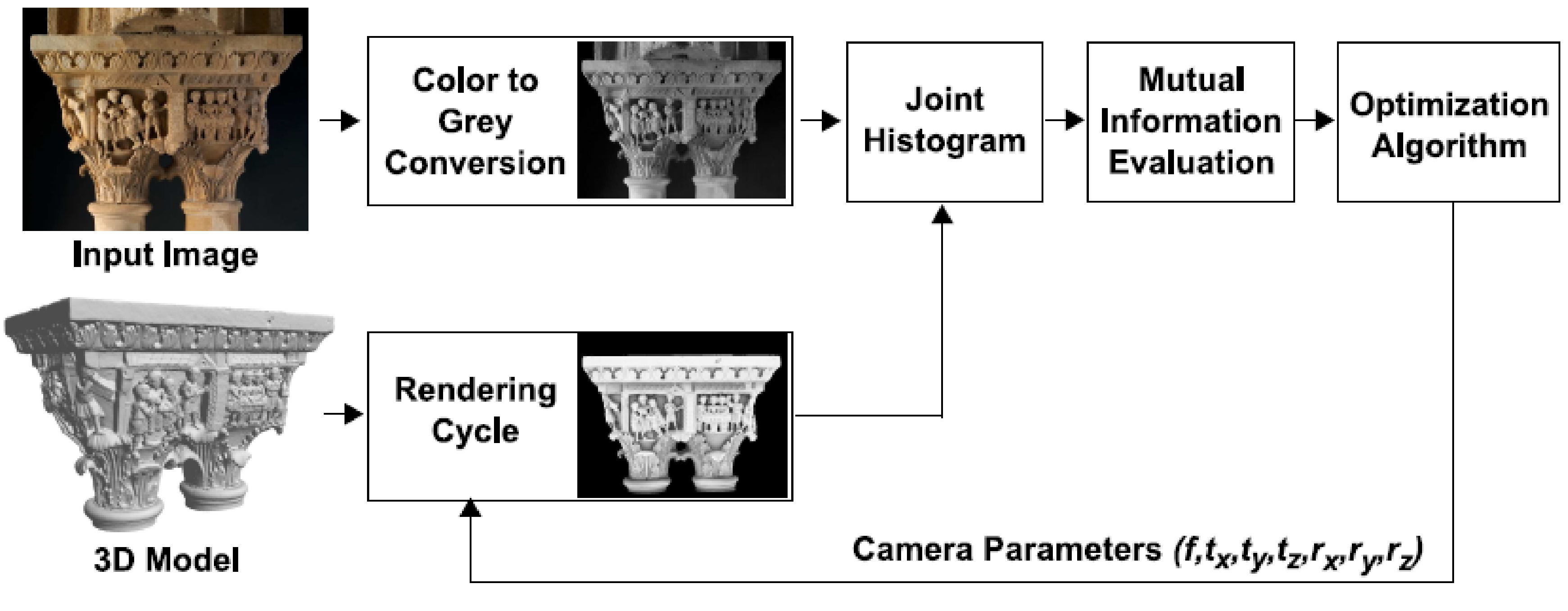
Image A

Image B

Joint Histogram



Registration Framework



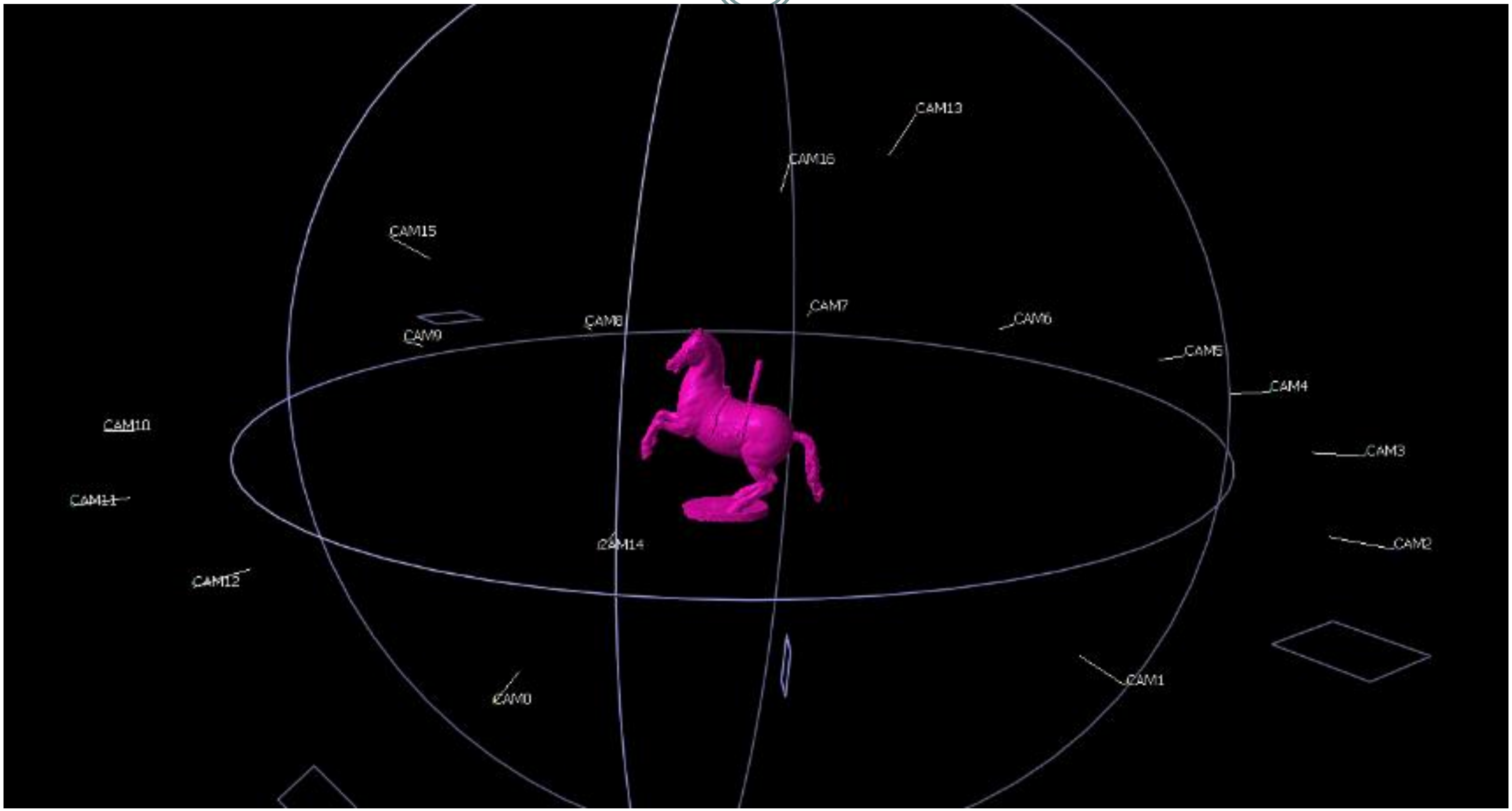
- Color-to-grey conversion is obtained with standard CIE Y conversion
- Optimization is done with NEWUOA (Powell et al., 2004) methods that is a greedy approach based on quadric approximation

Image registration: pros and cons

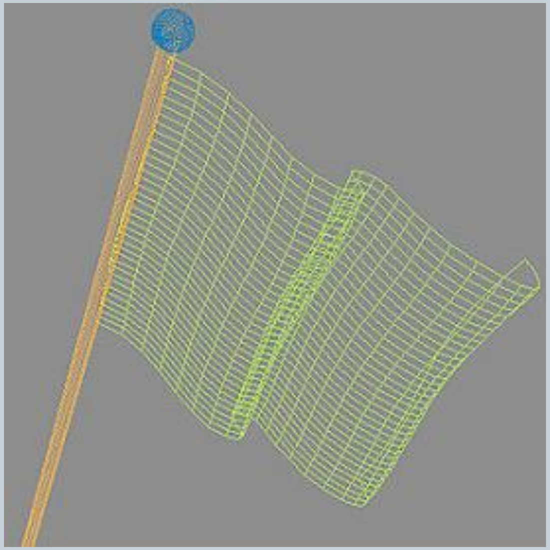
- User friendly
- Tens of images on one model
- Very flexible (from statues to buildings)
- Extensible

- Extrinsics/intrinsics
- Dependent on correspondences / Starting point
- Measure of alignment quality

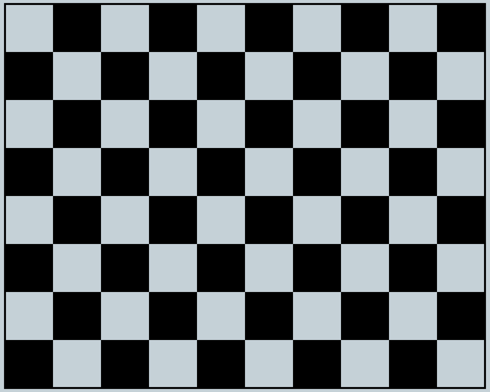
3D model, photos, camera parameters… and now?



Encoding the color information: Texture mapping



+



=

3D geometry

RGB texture
2D

(color-map)

Encoding the color information: Texture mapping

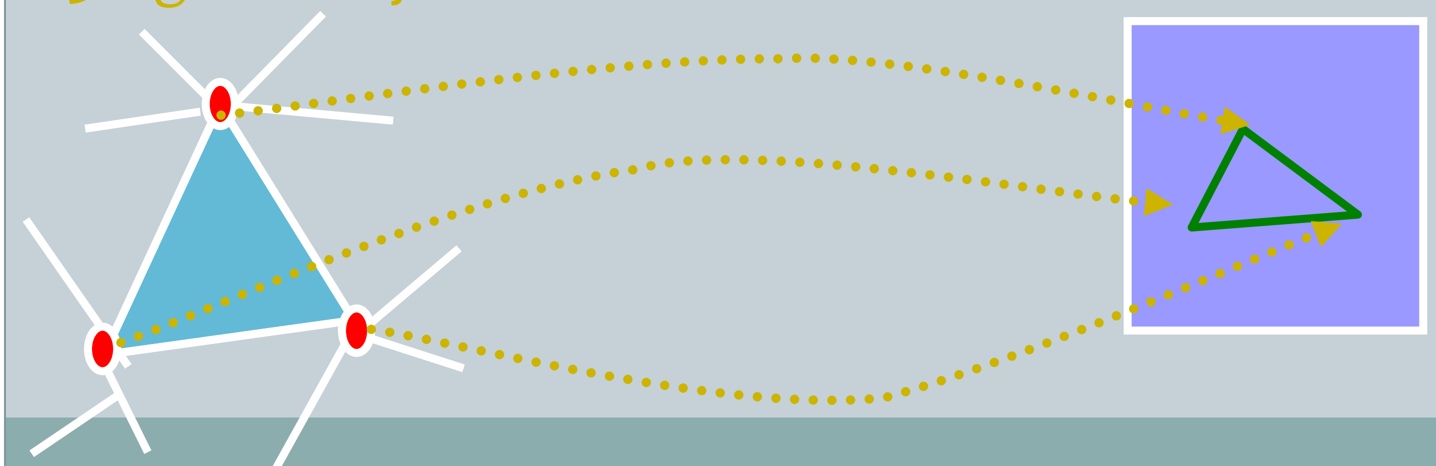
+

=

3D geometry

Texture: image…

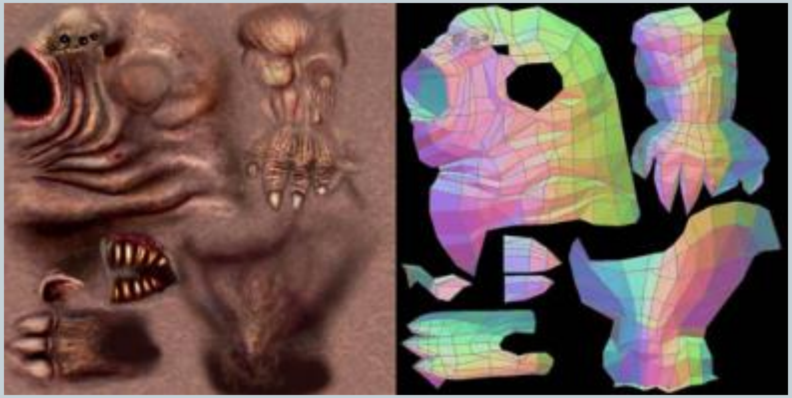
Texture mapped rendering



Encoding the color information: Texture mapping

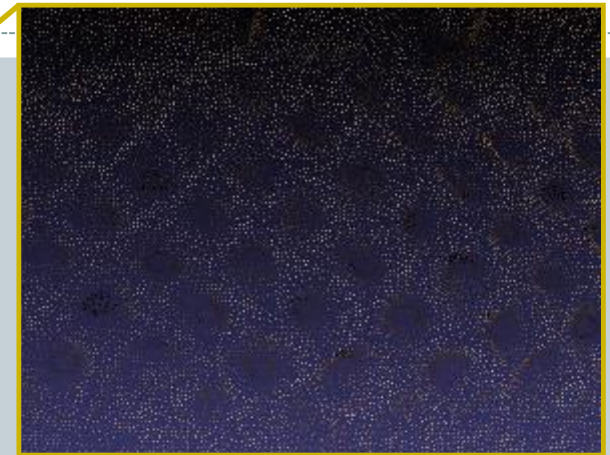
Encoding the color information: Texture mapping

Hand made, or automatized



Encoding the color information: Color per vertex

A color value is assigned to each vertex of the model.
The space between points is filled via interpolation.



Encoding the color information

Texture Mapping

- Independent of geometric density
- Compact 2D Structure
- Editable, compressible, easily accessible structure

- Parameterization
-Use with “multiresolution” or adaptive structures
-Need to pack data without losing detail
- Blending between photos

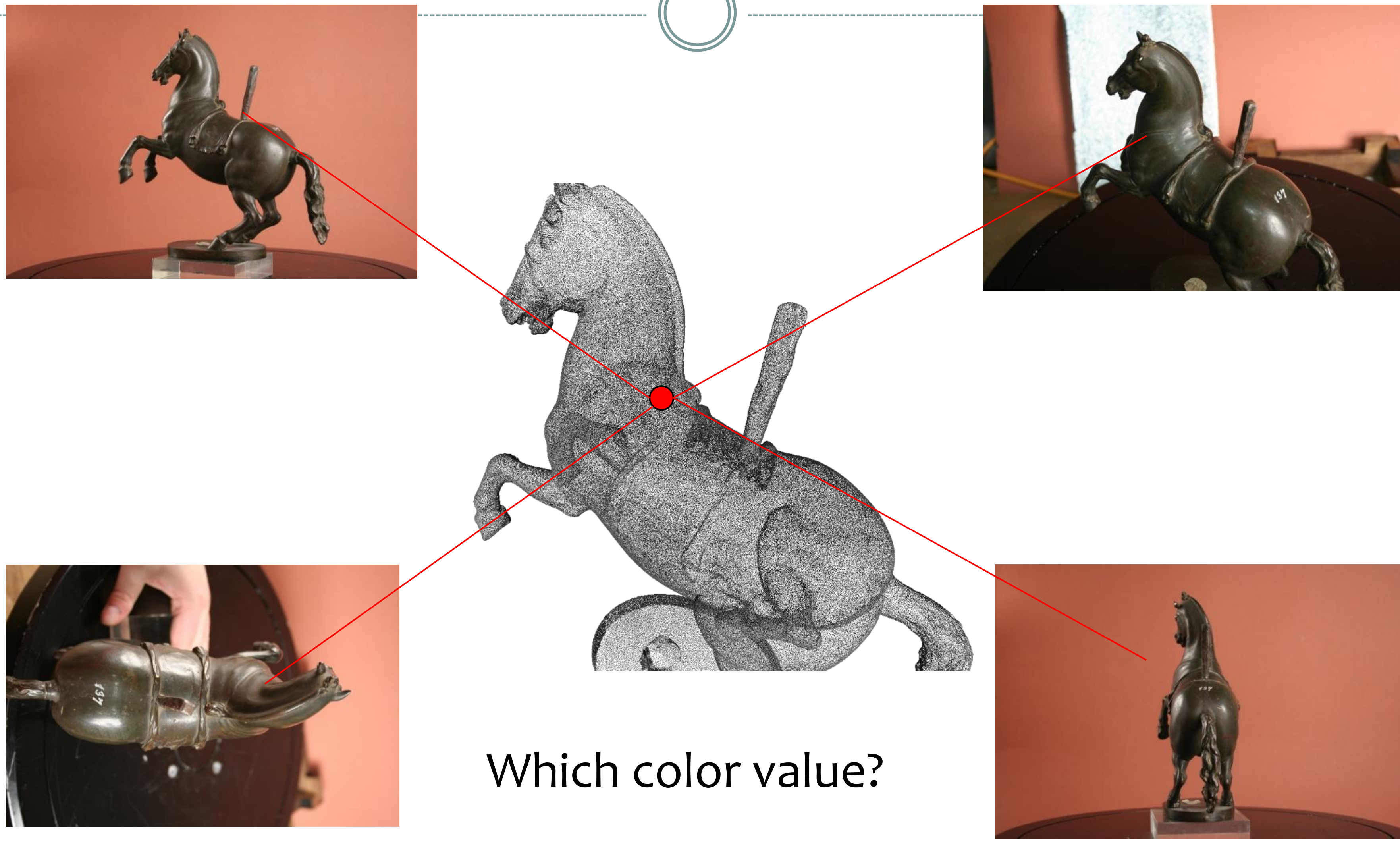
vs

Color per vertex

- Easy structure
- Compatible with “multiresolution” or adaptive structures
- No need for parametrization

- Very dependent on geometric density
- Harder to access or “boost”
- Texture mapping is more widely used

Mapping the color information



Which color value?

Mapping the color: automatic texture mapping

Weaver, a tool for the generation of texture maps

For each area, the better (orthogonal) photo is chosen

Mesh is splitted according to the photo allocation and parametrized using perspective projection from photos, the used area is cut and packed in the texture color discordances on borders are corrected

Reconstructing Textured Meshes

From Multiple Range RGB Maps

M. Callieri, P. Cignoni, and R. Scopigno 2002

Texure mapping

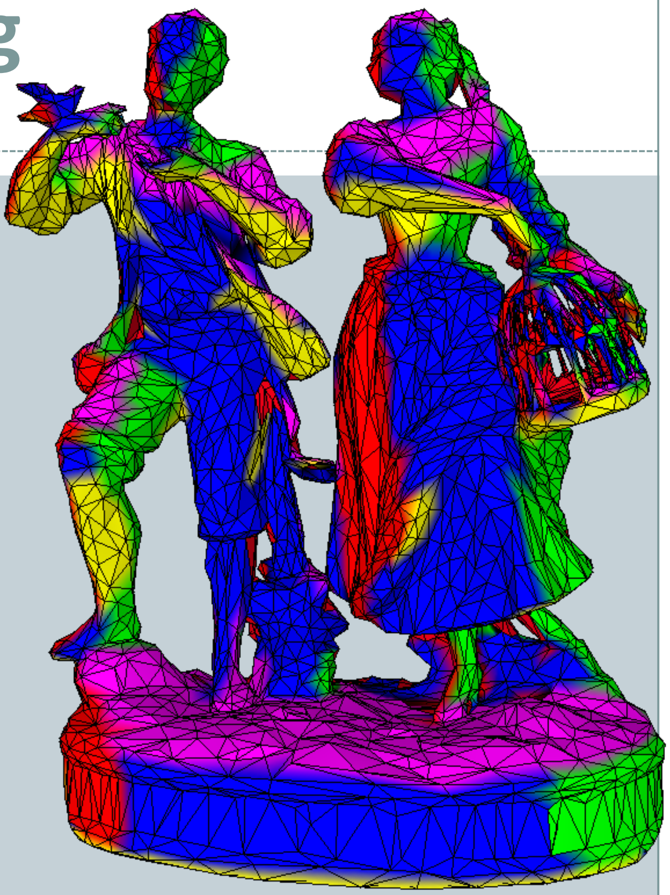
Texture parameterization:

1) *Vertex - to - Image* Binding:
   - For each vertex, find images which see it
   - Select best mapping

2) Patch Growing
   - Minimize fragmentation of texture patches

3) Patch Boundary Smoothing (reduce aliasing…)

4) Texture Patches Packing



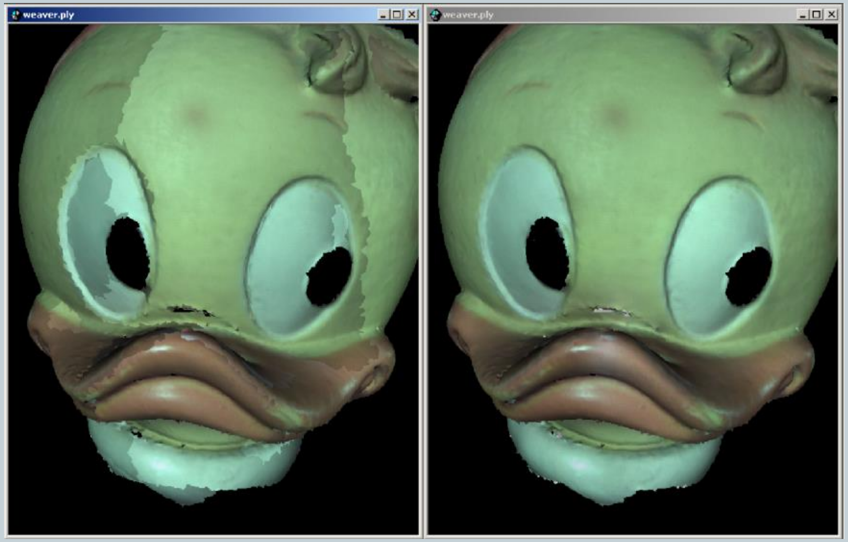
Reducing aliasing

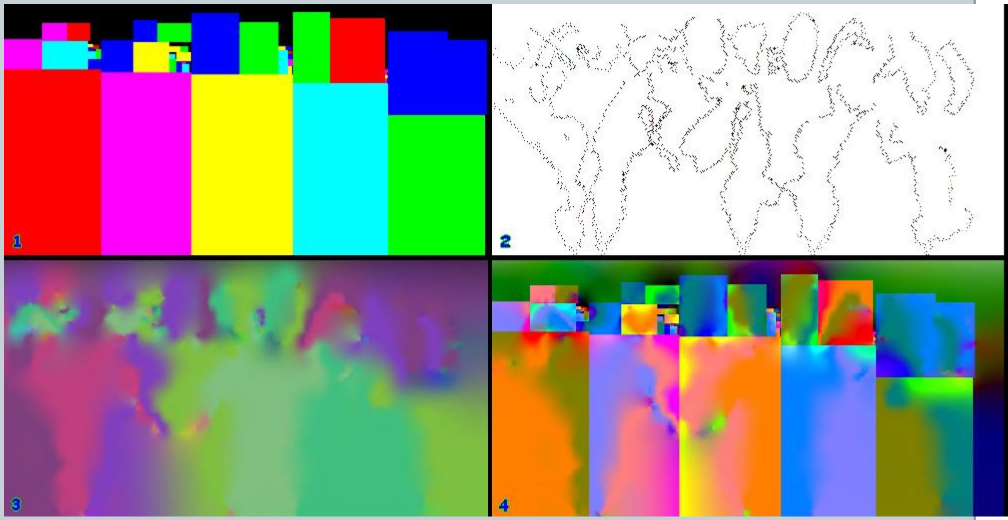
- Reducing aliasing on the boundary of texture patches:
  - **Miss-aligment**
    - Unsufficient image-to-geometry accuracy in aligning photos
    - Appears as "ghosting" effects or discontinuities of the drawing
  - Improve locally the match (local modifications of the UV mapping, guided by texture content)

Reducing aliasing

- Reducing aliasing on the boundary of texture patches:
  - **Color discontinuities** (due to different illumination condition in color sampling)

  - Detect differences on line of frontier of texture shift
  - Compute a color correction factor for each pair of corresponding texels on this frontier line (average color)
  - Smootly interpolate of those factors (push-pull algorithm)





Mapping the color: masked photo blending

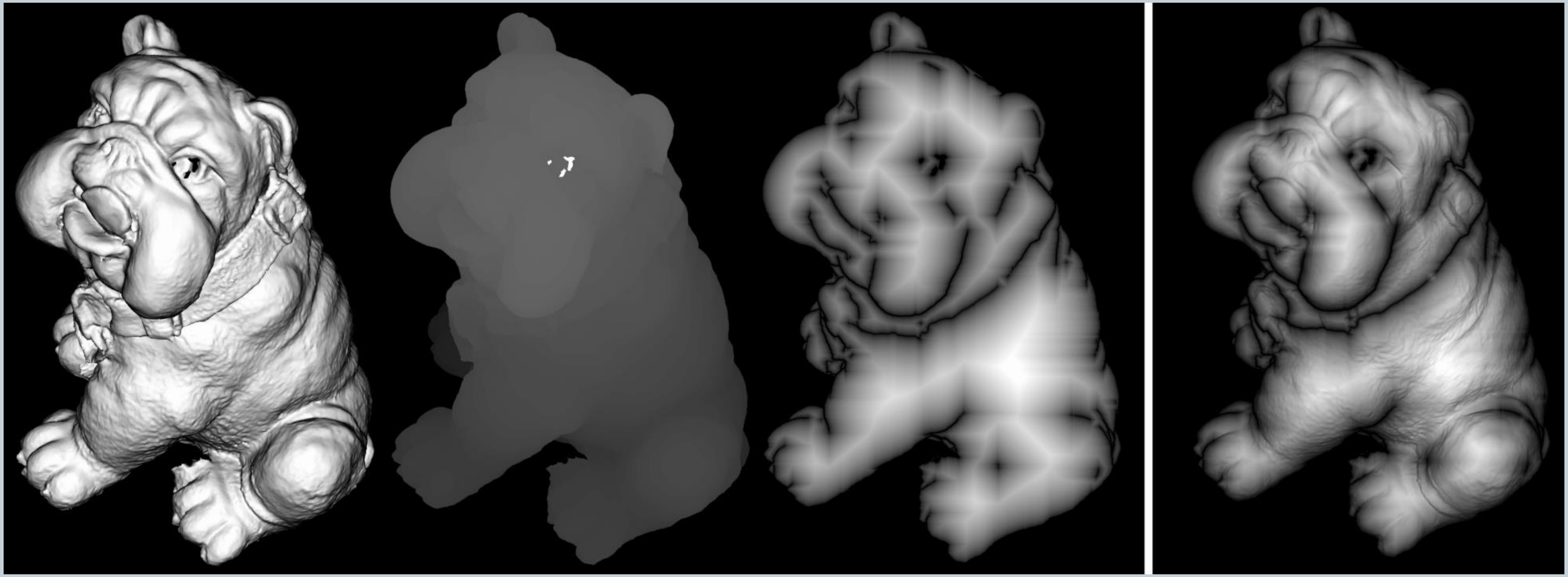
For each image, a set of quality masks is calculated.

Angle Mask

Depth Mask

Border Mask

Final Mask



Mapping the color: masked photo blending

Load Project

Calculate Masks

For each vertex:
-Load all the images (with masks) which map on it
- Assign a color value as a weighted sum of all contributions

Save Model!

This can be done Out of Core!

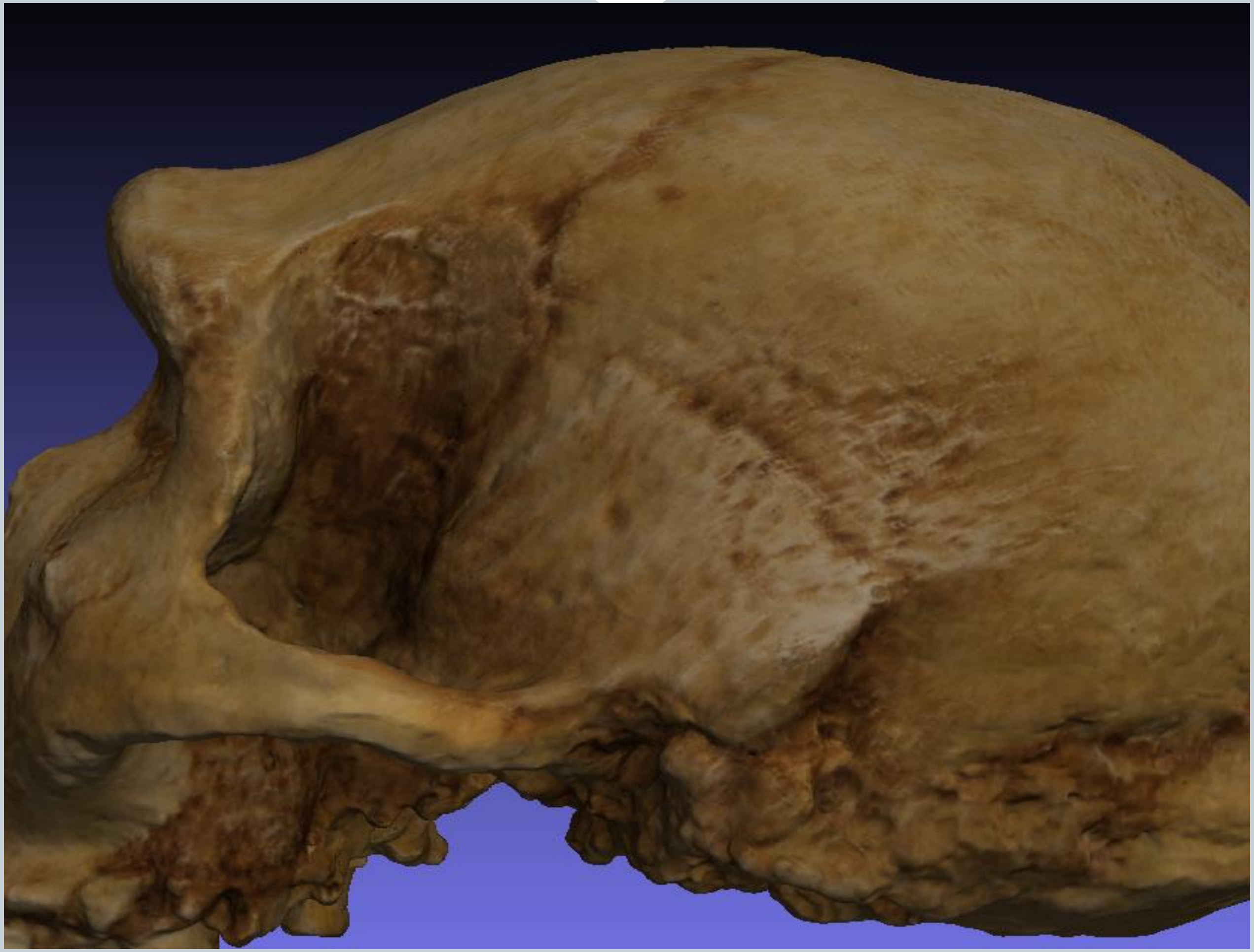
Masked photo blending
M. Callieri, P. Cignoni, M. Corsini and R. Scopigno

Blending – Needs accurate alignment!

- If the *img-to-3D* alignment is not very accurate → severe **blurring** introduced by blending!
- Alignment – error may depend on:
  - Inaccurate alignment (Manual? Automatic?)
  - Inaccurate digital 3D model (NOT the real object)
    - Data from TOF?
    - Bad scanning or raw data processing?
    - Excessive simplification?
  - Some examples…

Blending – Needs accurate alignment!

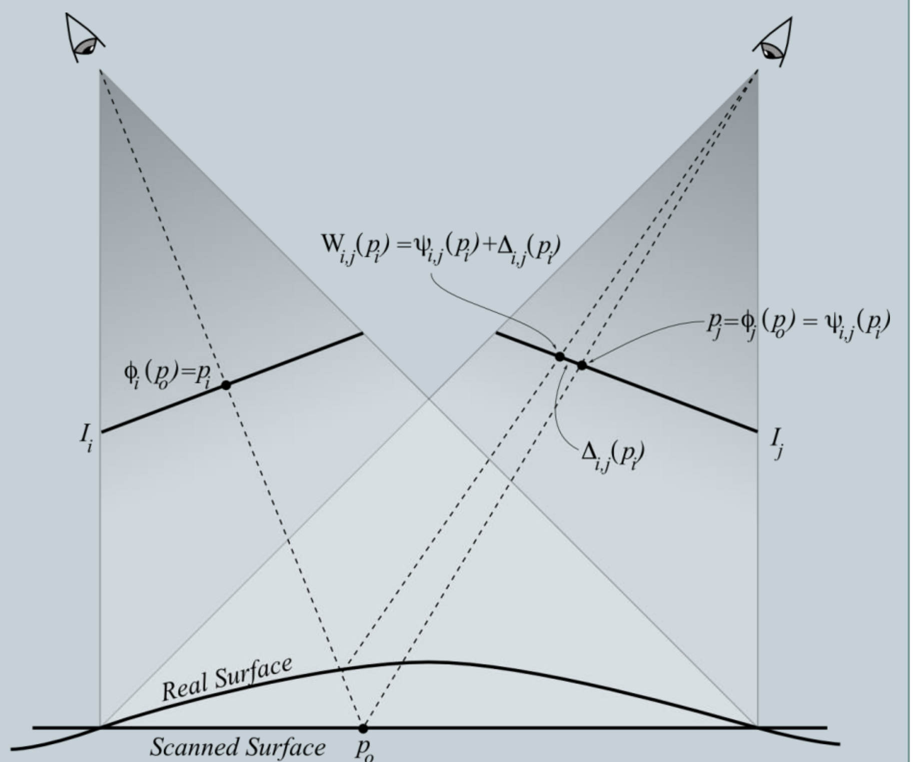
Blending – Needs accurate alignment!



Blending – Needs accurate alignment!

Improving mapping quality

- Assume alignment is the best you might produce, but still small errors (e.g. <1-2 mm)
- Improve mapping by **slightly locally warping** the input images:
  - Run **optical (pixel) flow** over each pair of overlapping images
  - Check how features correspond
  - Correct locally the misalignment: try to improve matching by **slightly warping** the images



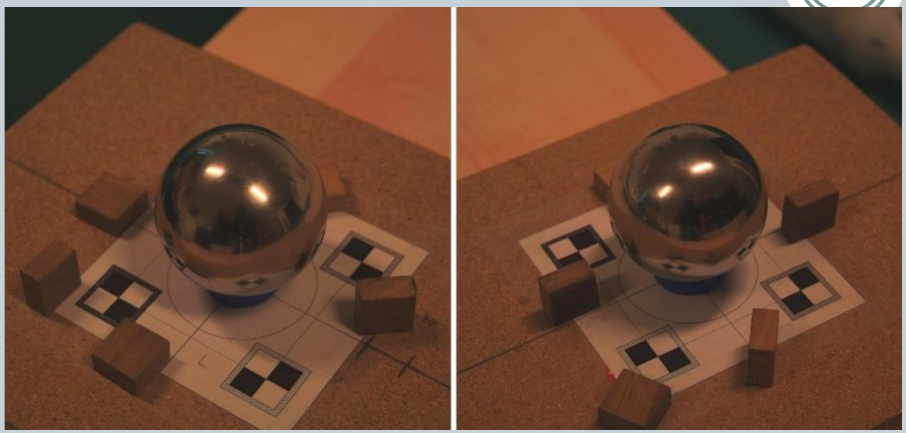
Improving mapping quality

M. Dellepiane, R. Marroquim et al, “Flow-based local optimization for image-to-geometry projection”, IEEE TVCG 2011

Color projection: open issues

The quality of color depends mainly on:
- the original photo set (shadows, highlights, uneven lighting, bad coverage)
- the quality of image registration

Color projection: controlling the light environment

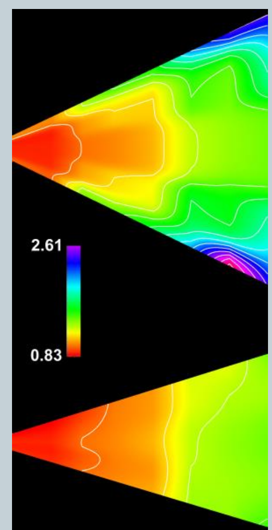


Use an acquisition device to estimate the lights in the scene.

Stereo light probe
M. Corsini et al. 2008

“Calibrate” a light source to correct image artifacts before and during projection

Flash lighting space sampling
M. Dellepiane et al. 2009



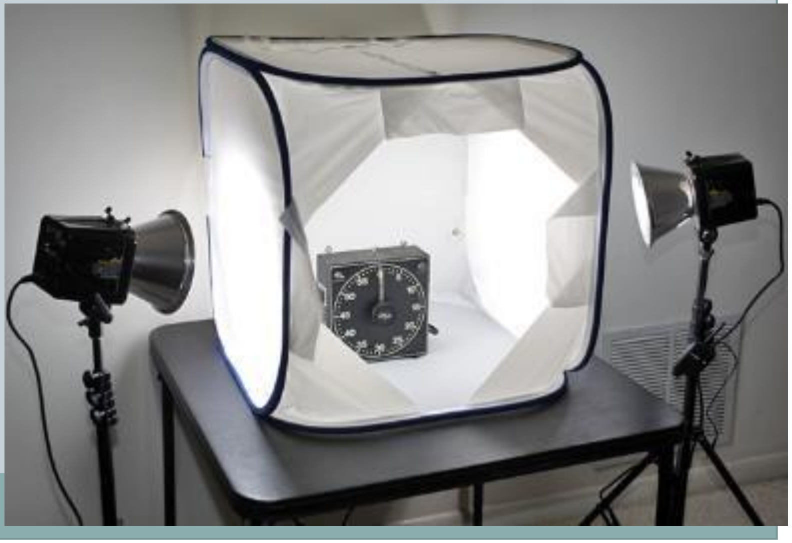
Acquisition of albedo

- Adopts a loosely-controlled illumination environment
- Shot multiple (**6**) images for each view, with different lighting conditions (SW-controlled lights)
- For each set of multi-illumination images, pixel per pixel:
  - remove highlights (intensity peaks)
  - remove shadows (low intensity pixel values)
  - de-shading using 3D geometry (simple Lambertian model) and the remaining pixel values (min 3)
  - **Output**: a single texture for each view, with de-shaded and void pixels

In practice...

- Use light diffusers, or a light tent
- Beauty dish + flash is your friend on smaller objects
- Use may lights, and make light bouce around
- For outdoor scenes, wait for an overcast day





MeshLab in full color

Image Alignment

Filters->Camera->Image registration: Mutual information

Usage:

1) Get Shot
2) Apply

Note: Focal length issue

Coming soon:

Use of correspondences/hybrid method (Sottile et al 2010)

MeshLab in full color

Color projection

Filters->Camera->Project active rasters to current mesh

Usage:
1) Apply

Color per vertex
or
Texture (if you already have a parametrization)

MeshLab in full color

Parameterization

Filters->Texture-> Parameterization + Texturingfrom images

Usage:

1) Define texture name and resolution
2) Apply
3) Save model with texture

Color projection: wrap up

Big issues in color projection

- Photo shooting (lights setup, surface coverage)
- Material estimation
- Image registration (semi-automatic)
- Color encoding
- Color projection
- Visualization

Pseudo-conclusion: the approach depends mainly on the object and the application