

電腦視覺與應用

Computer Vision and Applications

Lecture02-3-Pinhole camera-Visual Hull case study

Tzung-Han Lin

National Taiwan University of Science and Technology
Graduate Institute of Color and Illumination Technology

e-mail: thl@mail.ntust.edu.tw





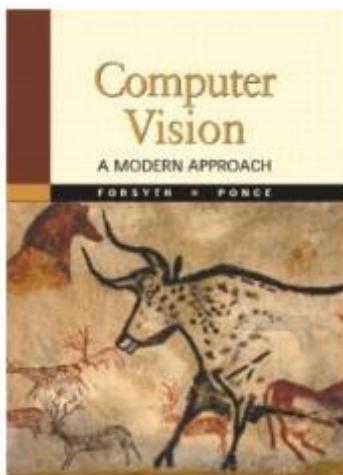
Pinhole Camera

- Case study
 - A simple visual hull algorithm
 - Shape from Silhouette



Pinhole Camera

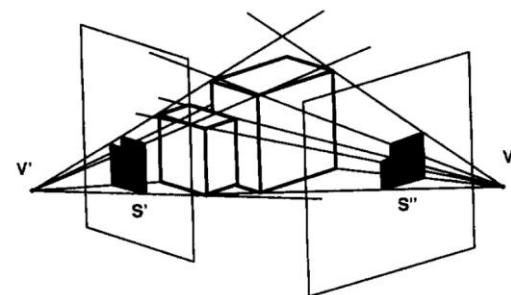
- Lecture reference coverage in following:
 - Computer Vision A Modern Approach, Chapter 26 (partly).
 - And miscellaneous papers from internet resource.





Visual Hull (sentences from Wiki)

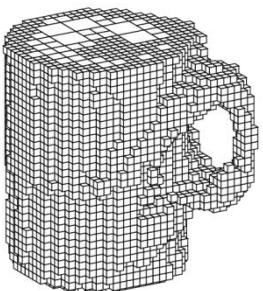
- “A **visual hull** is a geometric entity created by **shape-from-silhouette** 3D reconstruction technique introduced by A. Laurentini. This technique assumes the foreground object in an image can be separated from the background. Under this assumption, the original image can be thresholded into a **foreground/background** binary image, which we call a silhouette image. The foreground mask, known as a silhouette, is the 2D projection of the corresponding 3D foreground object. Along with the camera viewing parameters, the silhouette defines a back-projected generalized cone that contains the actual object. This cone is called a silhouette cone. The upper right thumbnail shows two such cones produced from two silhouette images taken from different viewpoints. The intersection of the two cones is called a visual hull, which is a bounding geometry of the actual 3D object (see the bottom right thumbnail).”



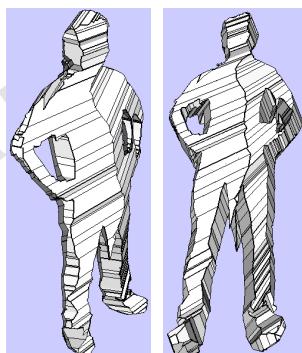


Visual Hull → Shape from Silhouette

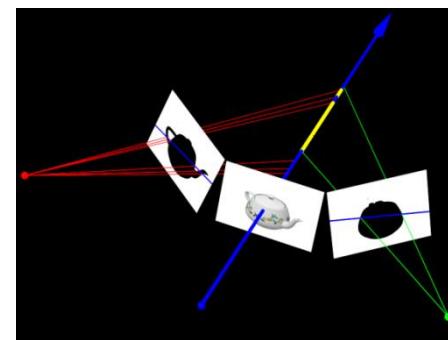
- Three categories from methods
 - 1. Voxel based extraction
 - 2. Exact Polyhedral visual hull
 - 3. Image-based visual hull



1



2



3

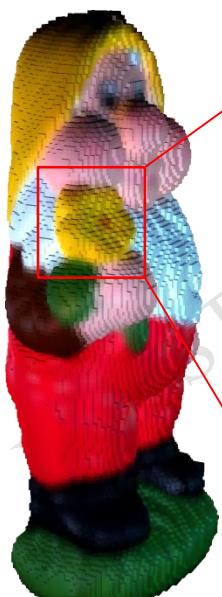


Voxel (Definition)

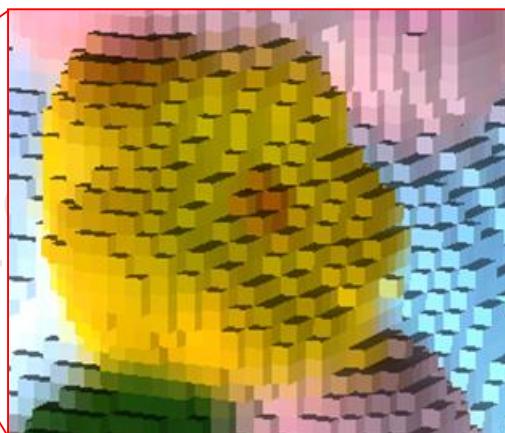
- “A voxel (volumetric pixel or, more correctly, Volumetric Picture Element) is a volume element, representing a value on a **regular grid** in three dimensional space.” (sentence from wikipedia)



Meshes



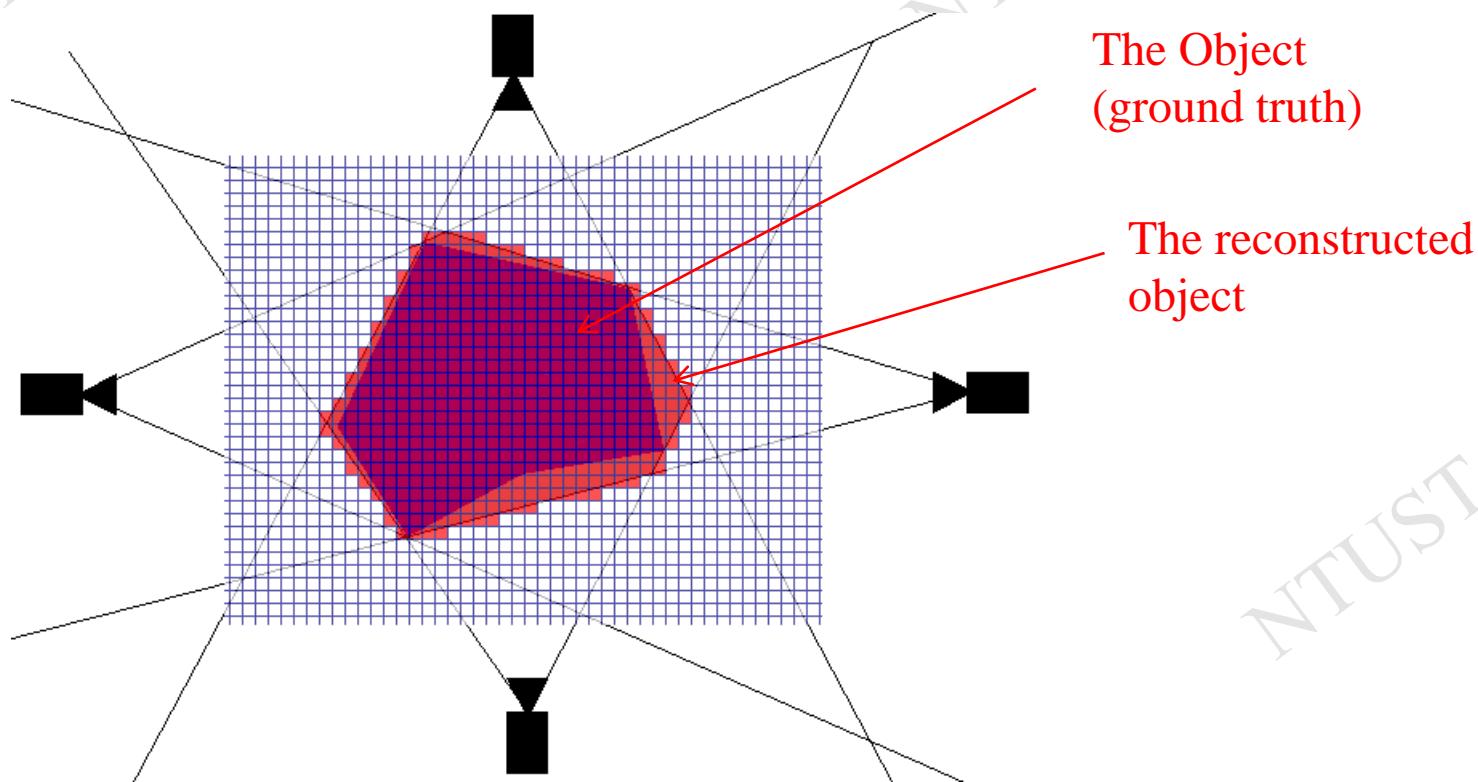
Voxels





Shape from Silhouette

- Basic idea...



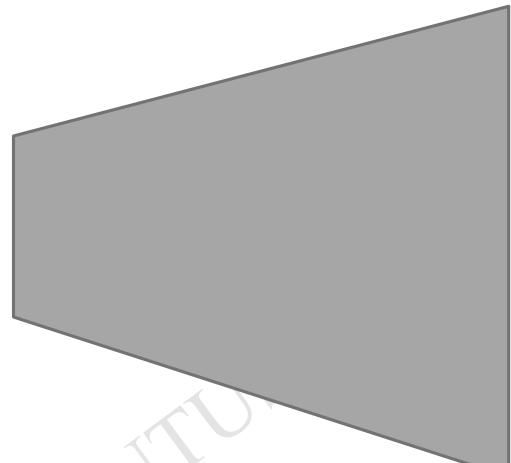
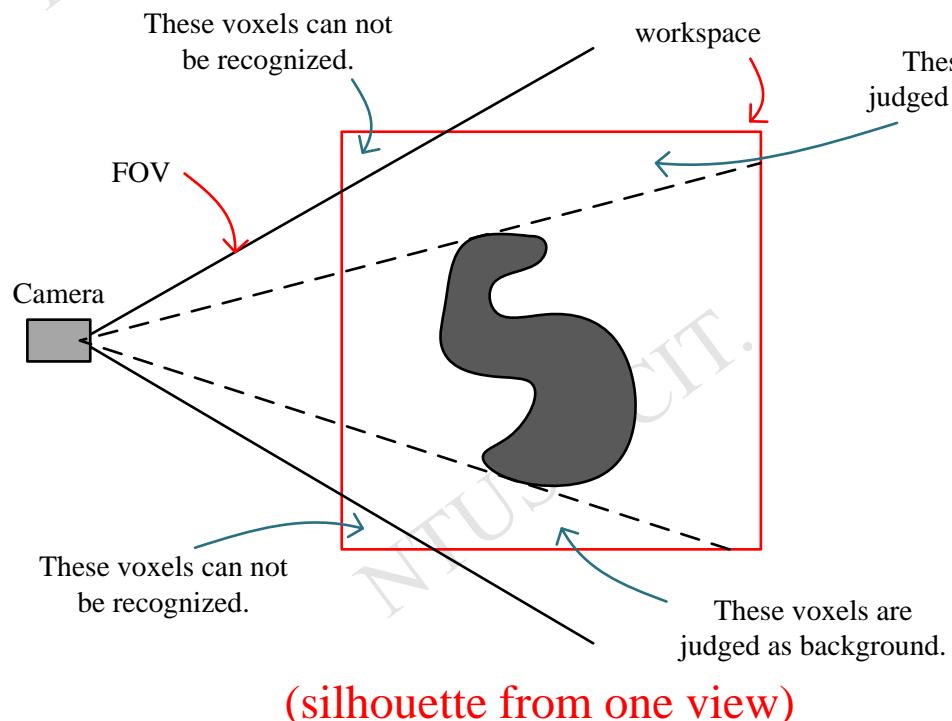
The Object
(ground truth)

The reconstructed
object



Shape from Silhouette

- Example for 2D Case in One view image

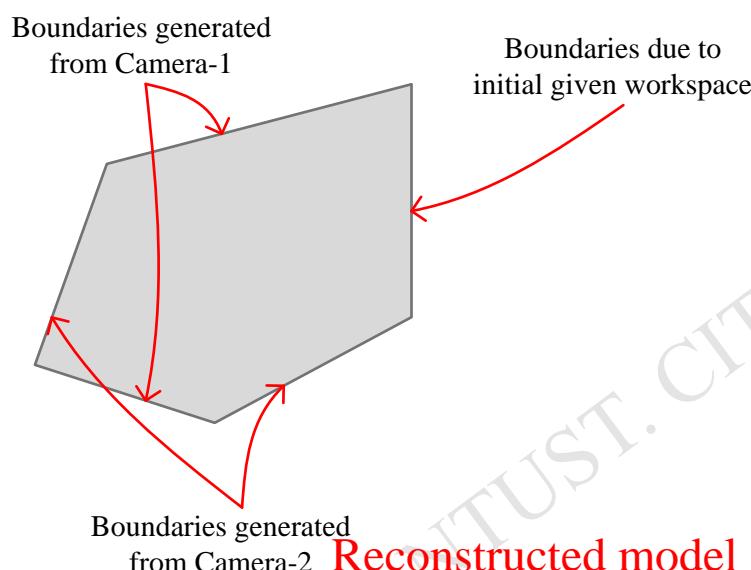
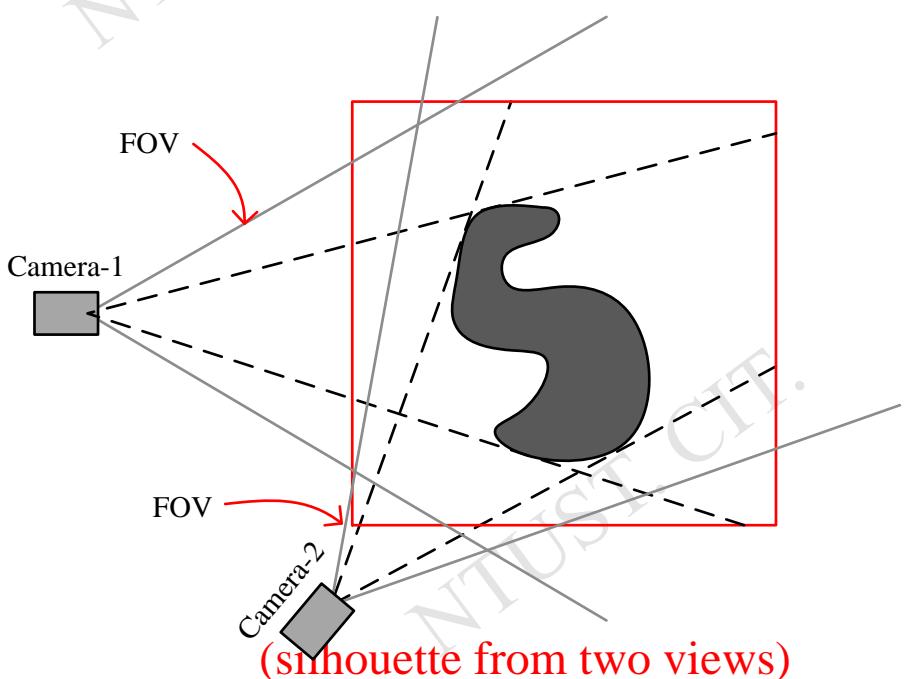


Reconstructed model



Shape from Silhouette

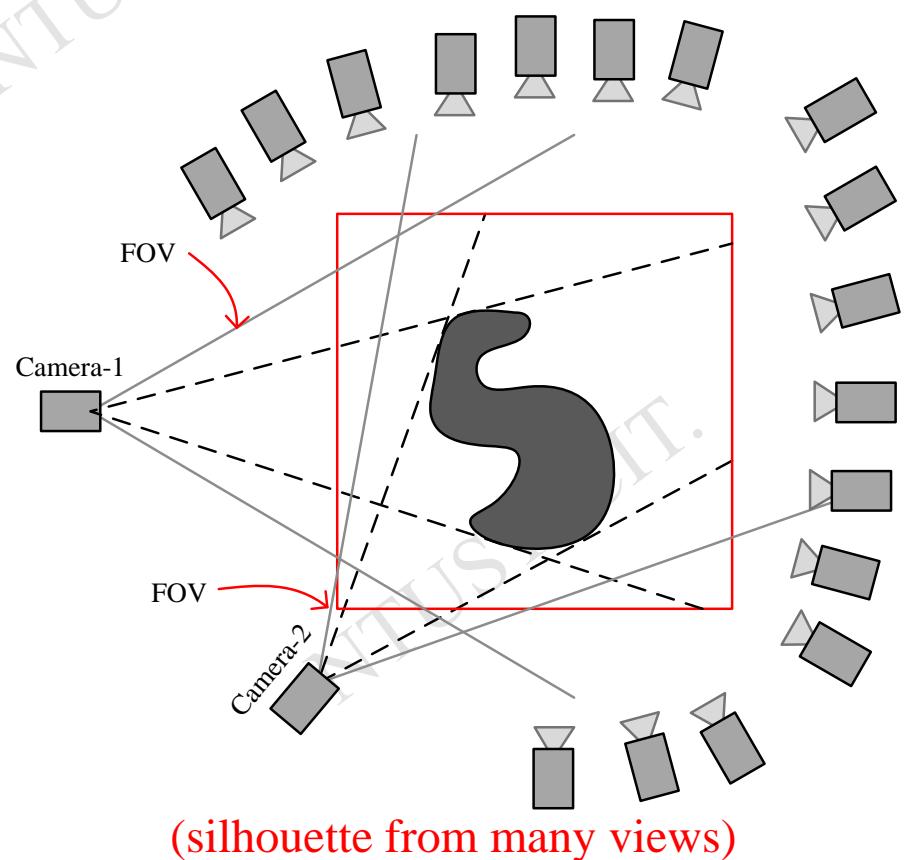
- Example for 2D Case in Two view images



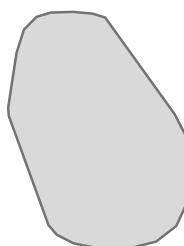


Shape from Silhouette

- Example for 2D Case in Many view images



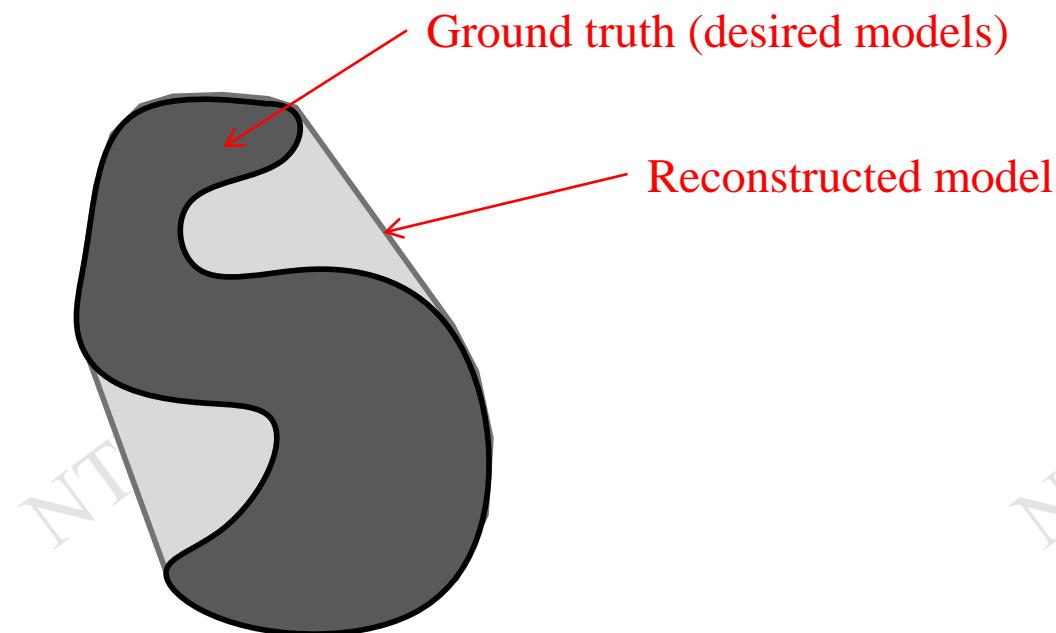
Reconstructed model
(convex hull)





Shape from Silhouette

- Note: the reconstruction model from visual hull is **NOT** necessary to be always convex (or says convex hull). Why?

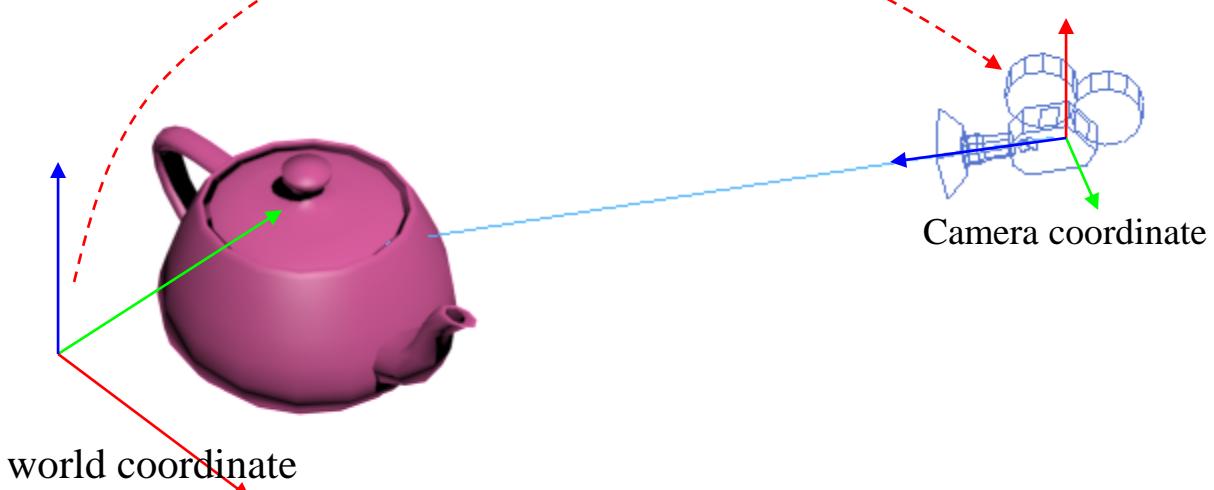




Shape from Silhouette

- What do intrinsic and extrinsic parameters mean?

Extrinsic parameter
(for this camera position)



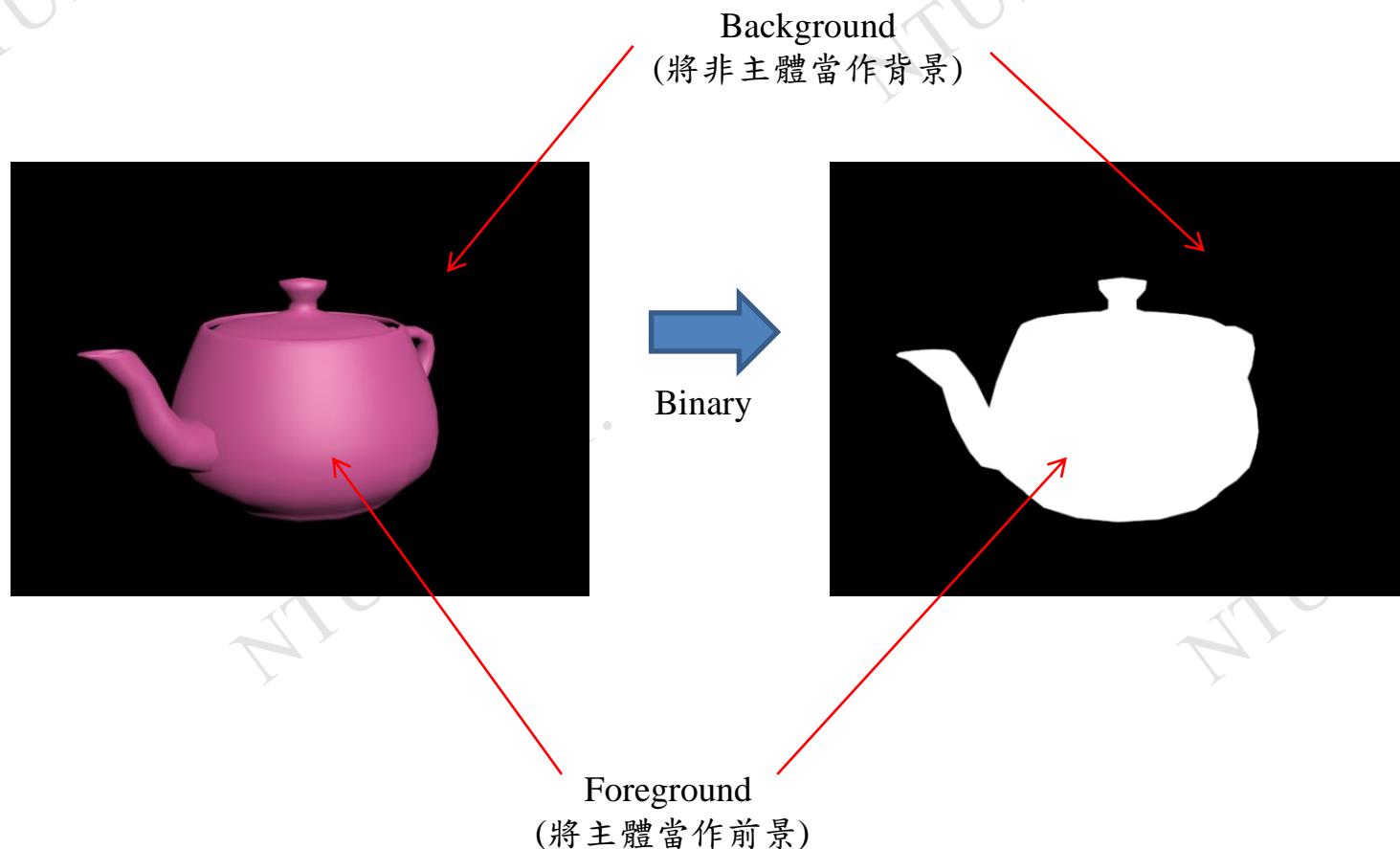
Applying intrinsic parameter





Shape from Silhouette

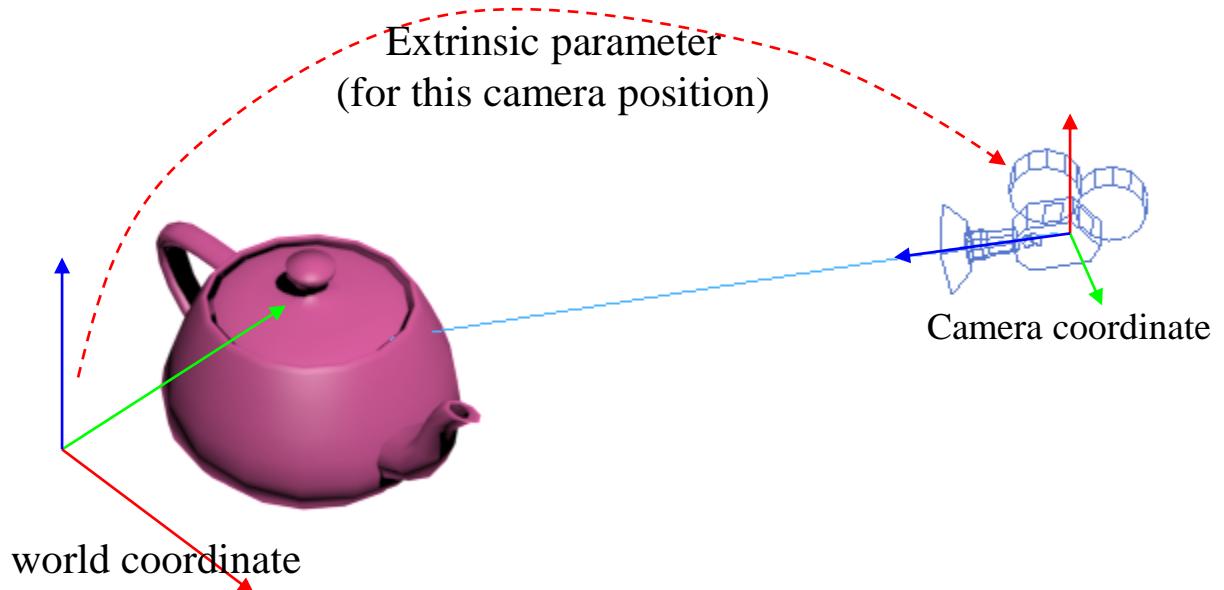
- What does background subtraction do?





Shape from Silhouette

■ Summary



Applying intrinsic parameter



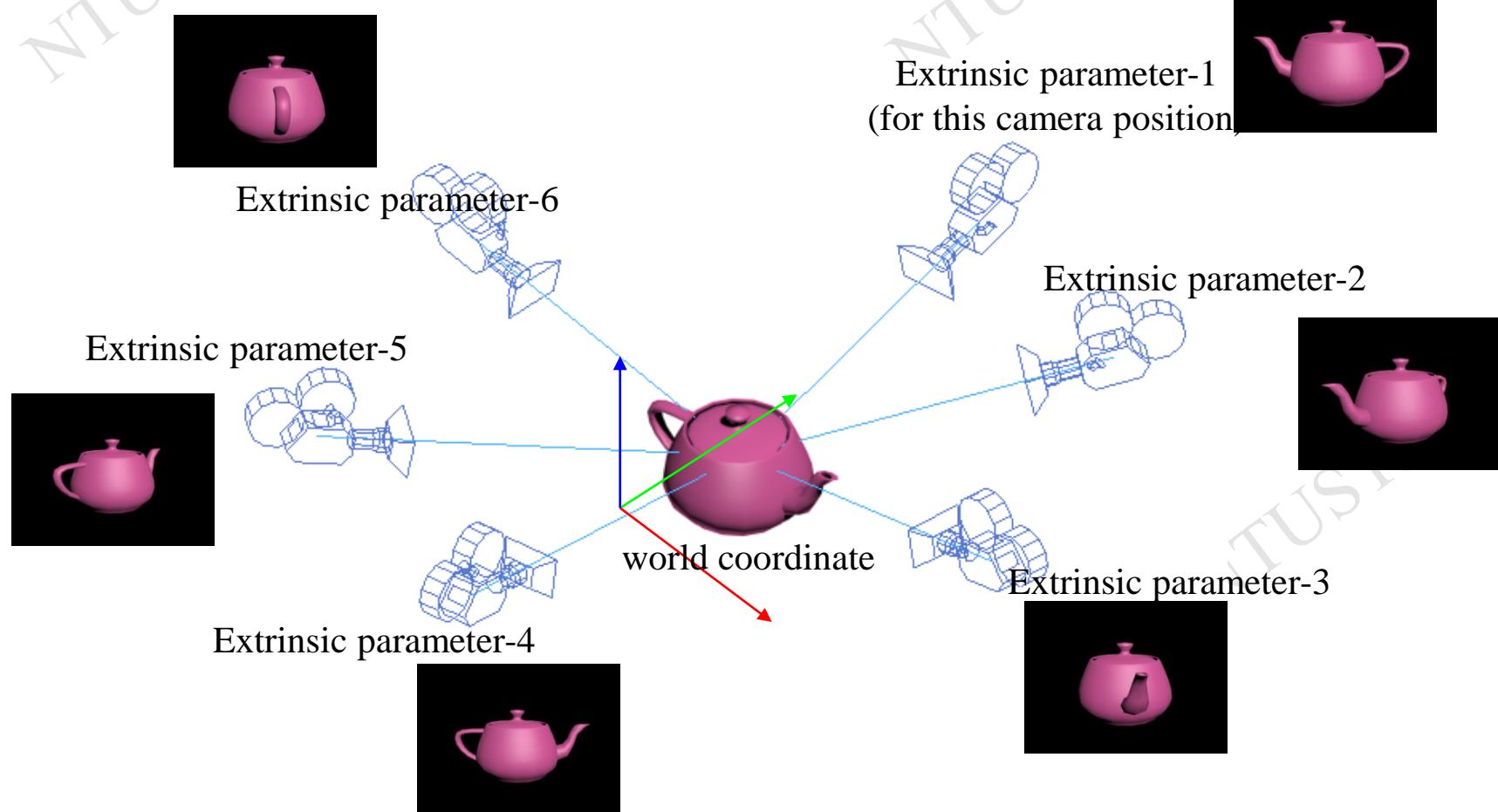
Binary map





Shape from Silhouette

■ Example

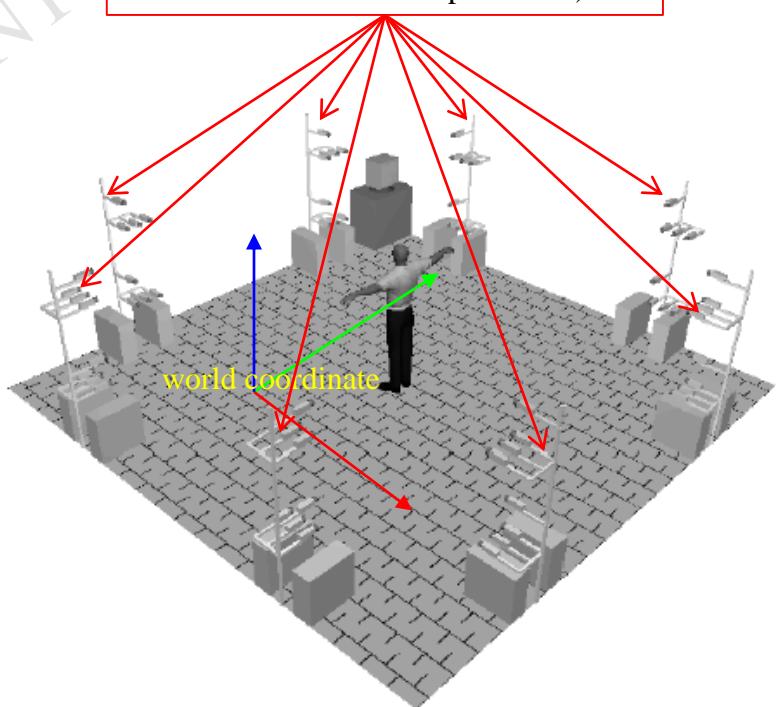




Shape from Silhouette (voxel based)

- To be a dynamic system. 8 sync. cameras

Cameras (calibrated and known
intrinsic & extrinsic parameter)

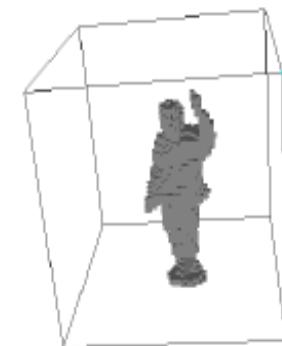


Solution:

When cameras' parameters are known (here, for example, eight cameras)

1. Create finite voxels in world coordinate
2. Projection for every voxel to each camera, and judge their existences.

3D reconstruction

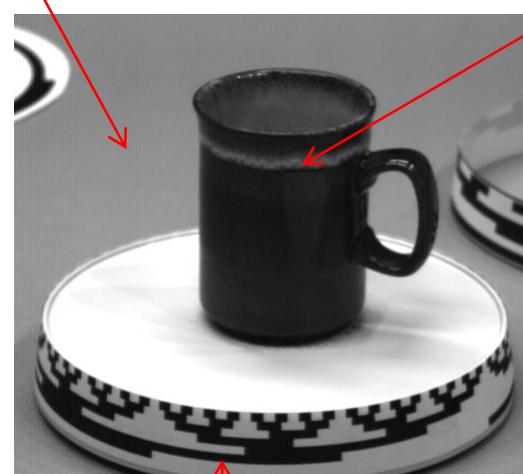




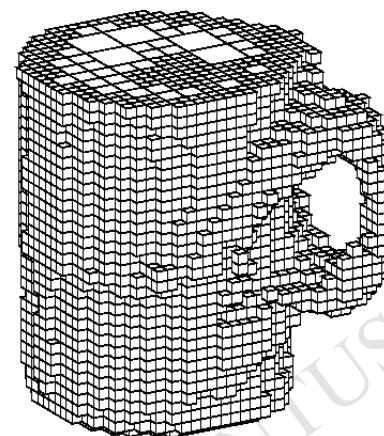
Shape from Silhouette (voxel based)

- One camera + turntable (static object).

Background
(除了想建構的3D model之外
的所有pixel都視為背景)



Foreground
(想建構的3D model)



One kind of Checkerboard
(for determining Extrinsic parameter)

Image by Richard Szeliski



Shape from Silhouette (voxel based)

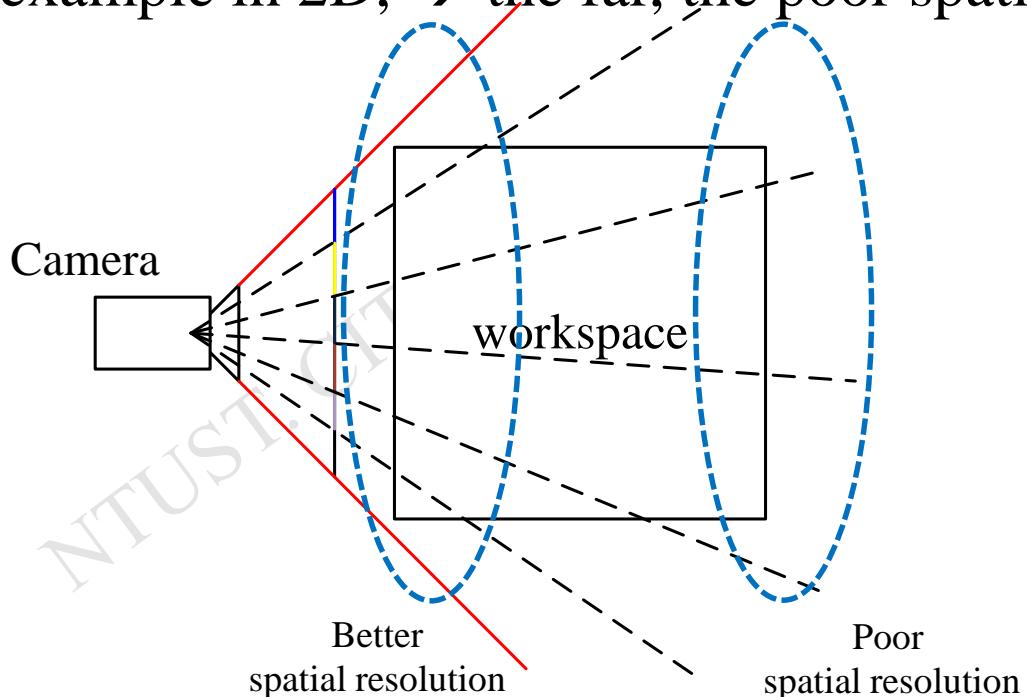
- Summary for the requirement of algorithm
 1. At least one view (known intrinsic & extrinsic parameters). → **Camera Calibration**.
 2. In each view, background and foreground pixels are needed to be identified. → In practice, “**Background subtraction**” algorithms may be used.
 3. The workspace in world coordinate should be defined. → define the volume of one voxel.





Shape from Silhouette (voxel based)

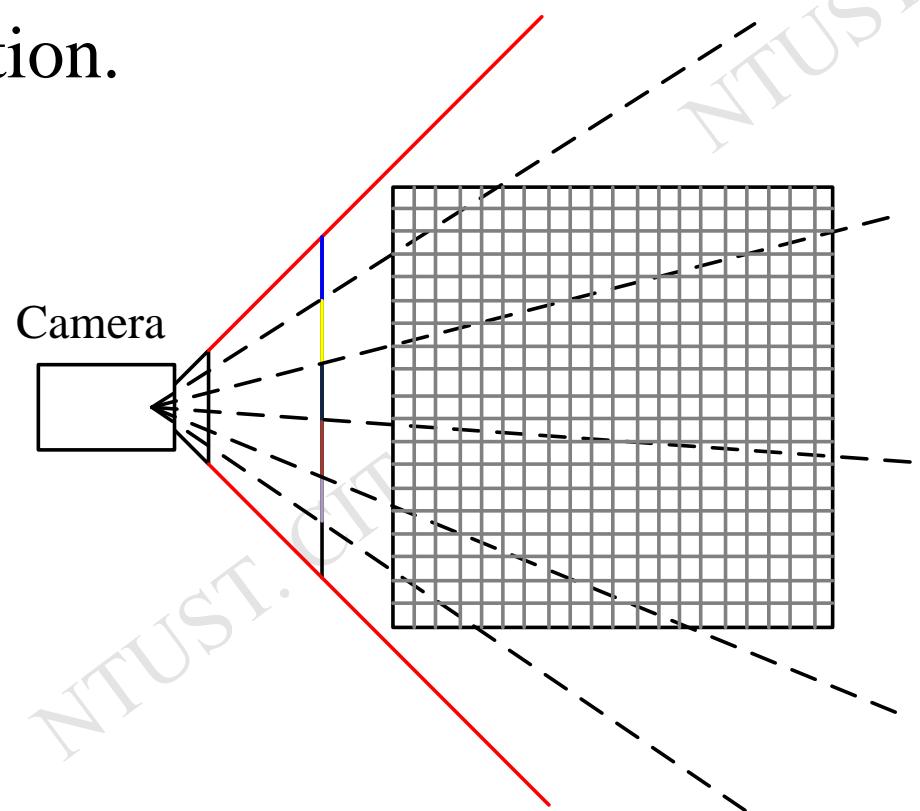
- The spatial resolution issue:
 - What is the resolution for better quality?
 - for example in 2D, → the far, the poor spatial resolution.





Shape from Silhouette (voxel based)

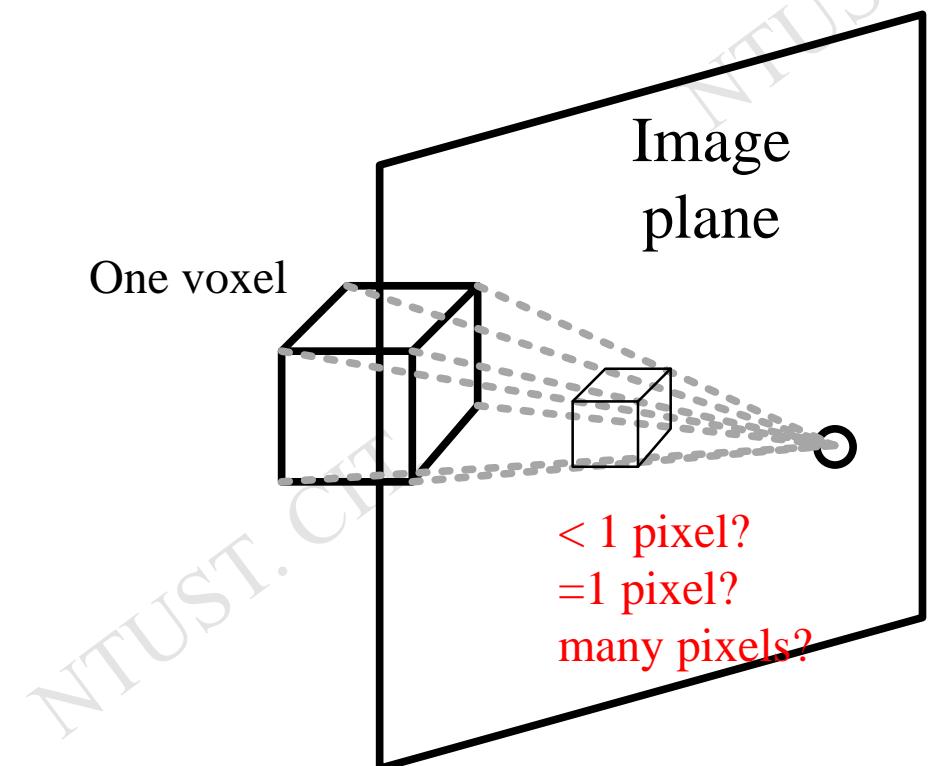
- The voxel size has a limitation due to spatial resolution.





Shape from Silhouette (voxel based)

- Voxel size and the projection size issue.

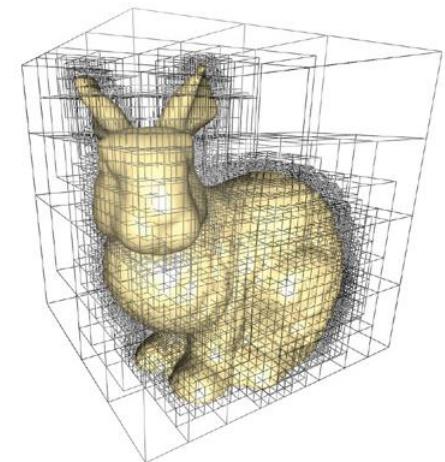
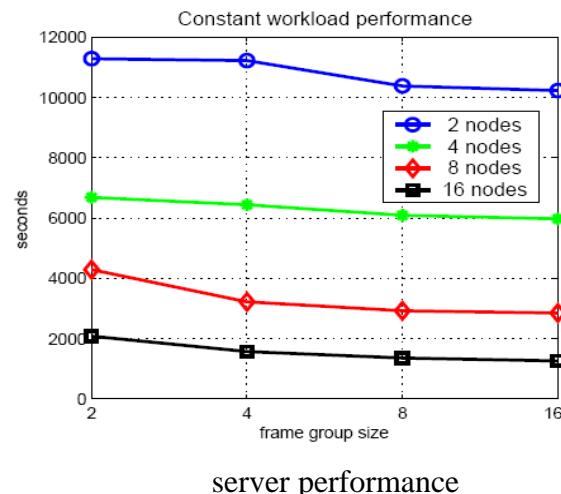
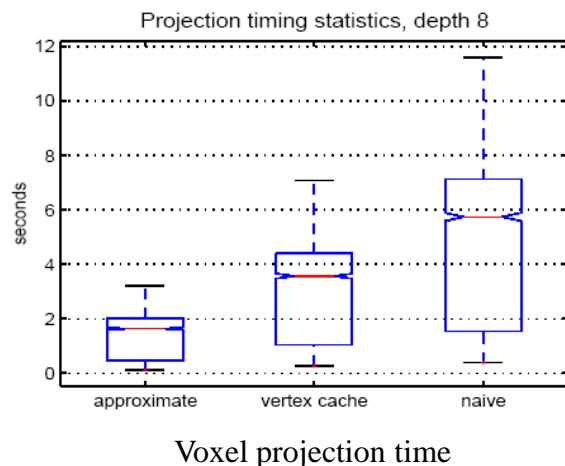




Shape from Silhouette (voxel based)

■ The speed-up strategy

1. Use **Octree** for hierarchically encoding voxels.
2. Approximate the “cube” projection
3. Parallel processing (clustering)

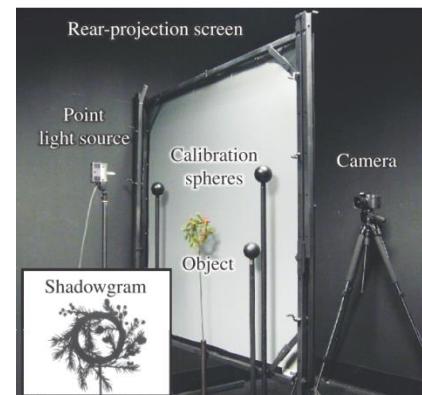
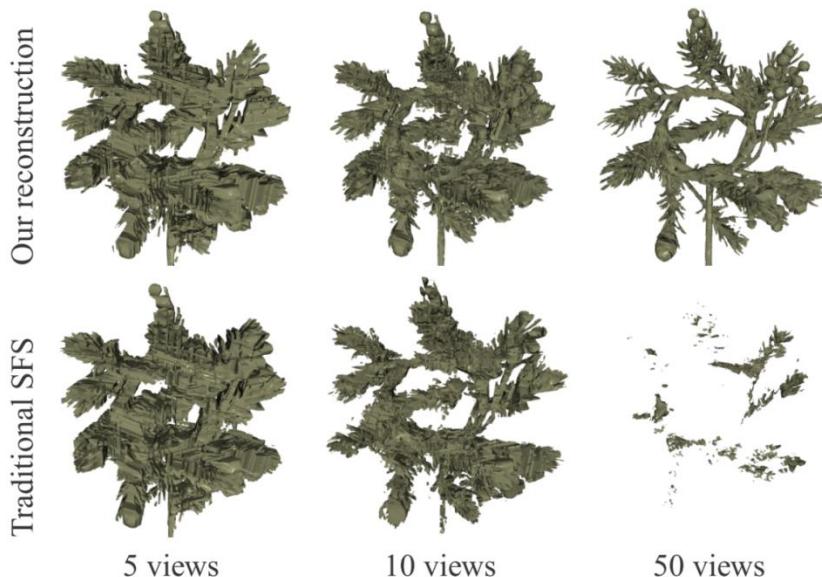


16 PCs (P2-450MHz)
100MBit networks(TCP/IP)



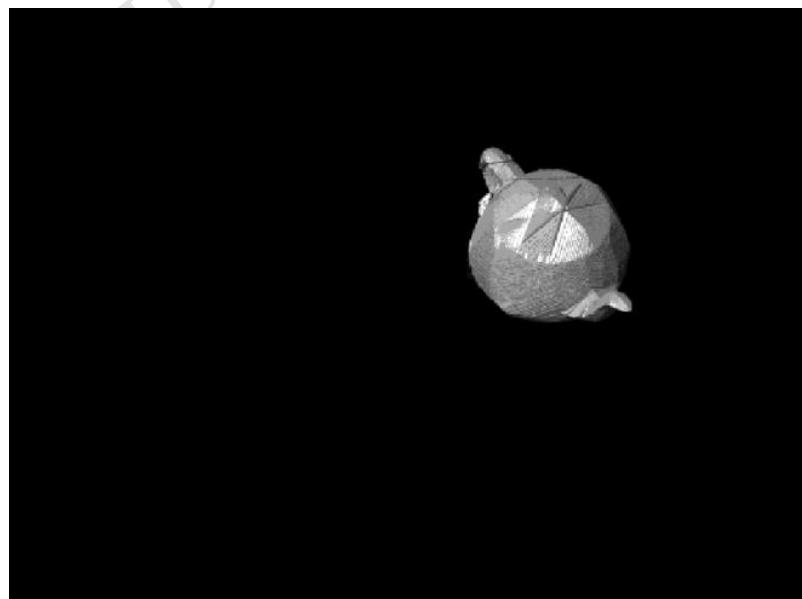
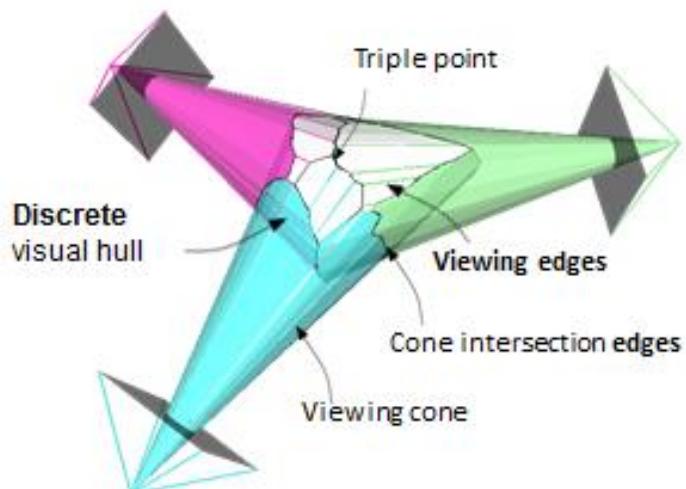
Shape from Silhouette

- The number of views issue:
 - The more views, the better reconstruction ?
 - NOT exactly, for example:





Exact Polyhedral Visual Hulls (EPVH) Algorithm



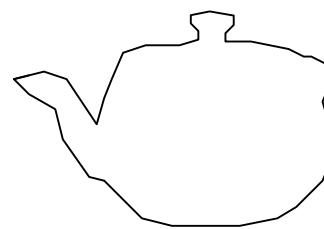
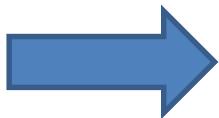


Exact Polyhedral Visual Hulls (EPVH) Algorithm

- Pre-processing: Before processing EPVH algorithm, you may need to “vectorize” your foreground object.



n pixels → foreground

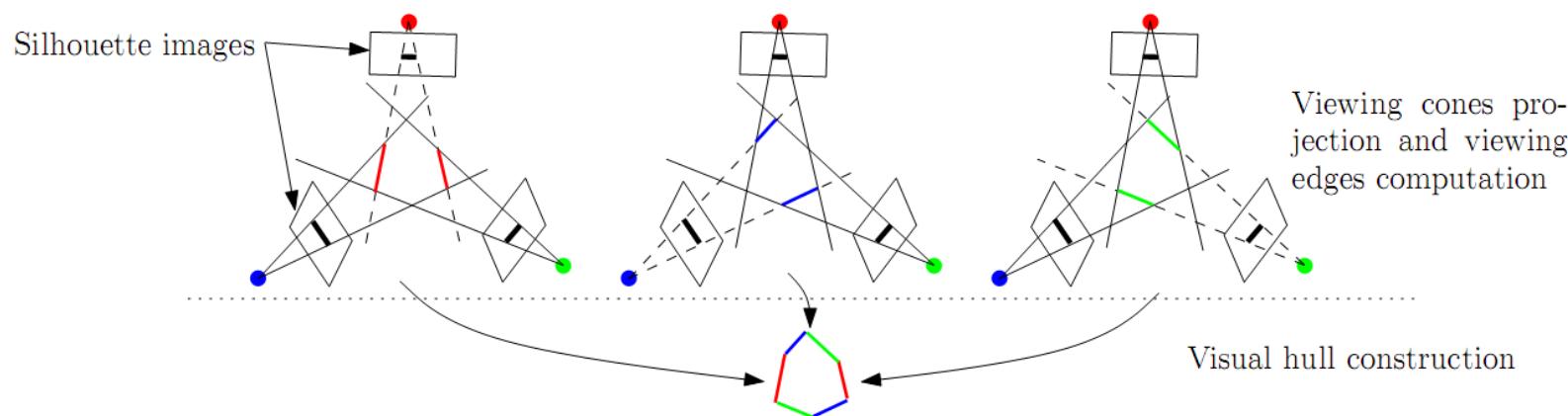
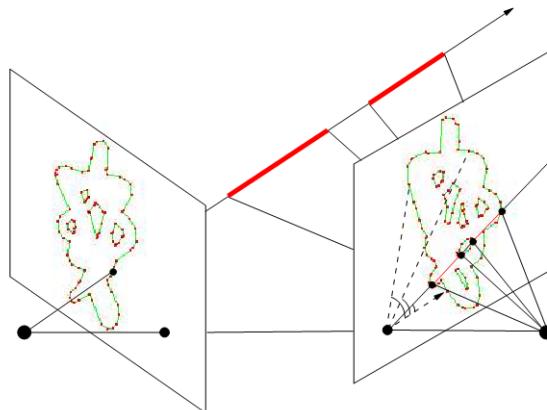


m vertexes for bounding a shape



Exact Polyhedral Visual Hulls (EPVH) Algorithm

- Determination for viewing edges (epipolar geometry)

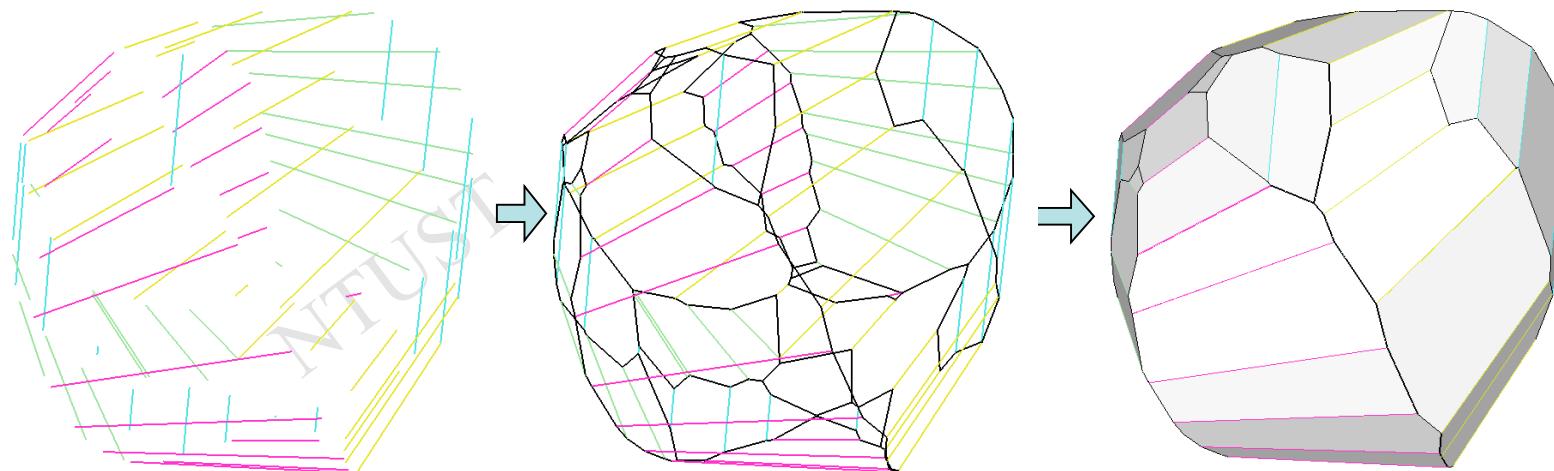




Exact Polyhedral Visual Hulls (EPVH) Algorithm

■ 3 Steps to EPVH

- 1. Compute viewing edges
- 2. Cone intersection edges and triple points
- 3. Faces.





Exact Polyhedral Visual Hulls (EPVH) Algorithm

- First, silhouette images are converted to polygons.
(convex or non-convex, with holes allowed)
- Each edge is back projected to form a 3D polygon.
- Then each polygon is projected onto each image, and intersected with each silhouette in 2D.
- The resulting polygons are assembled to form the polyhedral visual hull



Exact Polyhedral Visual Hulls (EPVH) Algorithm

■ Speed v.s. Completeness

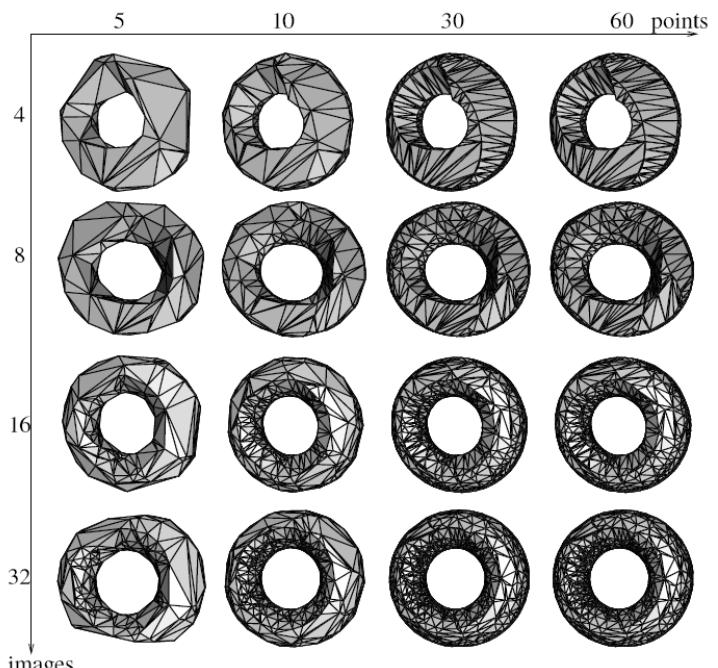
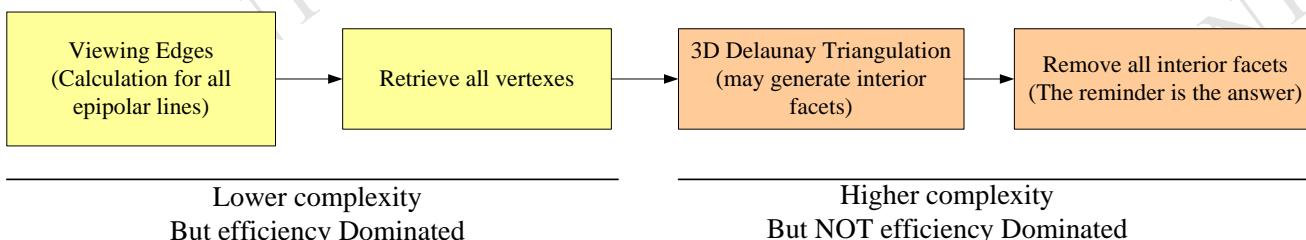
Algorithm 1 Visual hull surface points

```

1: for all contours  $O_j^i$  in all images: do
2:   for all images  $k$  such that  $k \neq i$ : do
3:     for all points  $p_j^i$  in  $O_j^i$  : do
4:       compute the epipolar line  $l$  of  $p_j^i$  in image  $k$ ,
5:       for all contours  $O_l^k$  in image  $k$ : do
6:         compute the intersections of  $l$  with  $O_l^k$ ,
7:         update depth intervals along the viewing line
8:           of  $p_j^i$ ,
9:         end for
10:      end for
11:      compute the 3D points delimiting intervals along
12:        the viewing line of  $p_j^i$ .
13:    end for
14:  end for

```

Hybrid algorithm





Exact Polyhedral Visual Hulls (EPVH) Algorithm

- Parallel processing to real-time

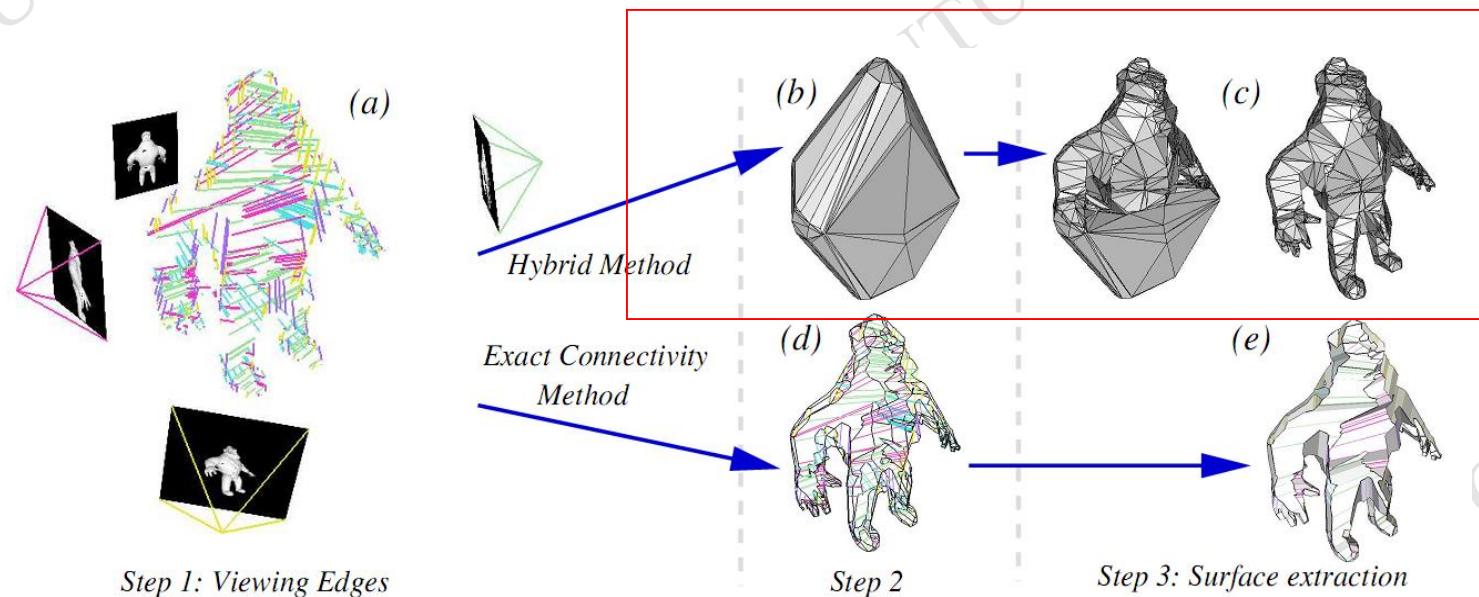


Figure 3: Outline of the visual hull modeling techniques chosen for parallelization. (a) Images of an object are taken, silhouettes are identified, their contours vectorized, and viewing edges are computed for each point of the discretization. (b) The hybrid method computes the Delaunay tetrahedron decomposition of space based on viewing edge vertices. (c) Each tetrahedron is carved according to silhouette consistency, and the final visual hull model is obtained. (d) The exact connectivity method computes the cone intersection components belonging to the visual hull, to complete the entire visual hull polyhedron mesh. (e) Faces are extracted from the mesh representation and the final polyhedron model of the visual hull is obtained.



Exact Polyhedral Visual Hulls (EPVH) Algorithm

■ More examples...

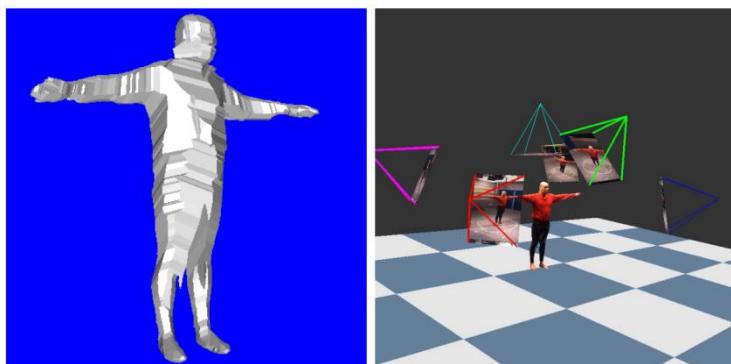
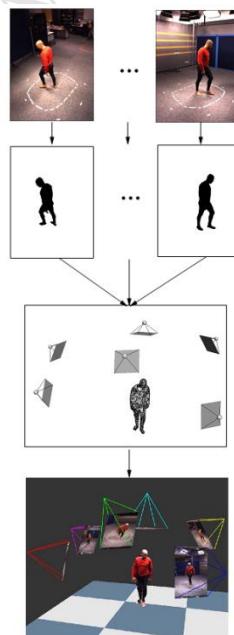
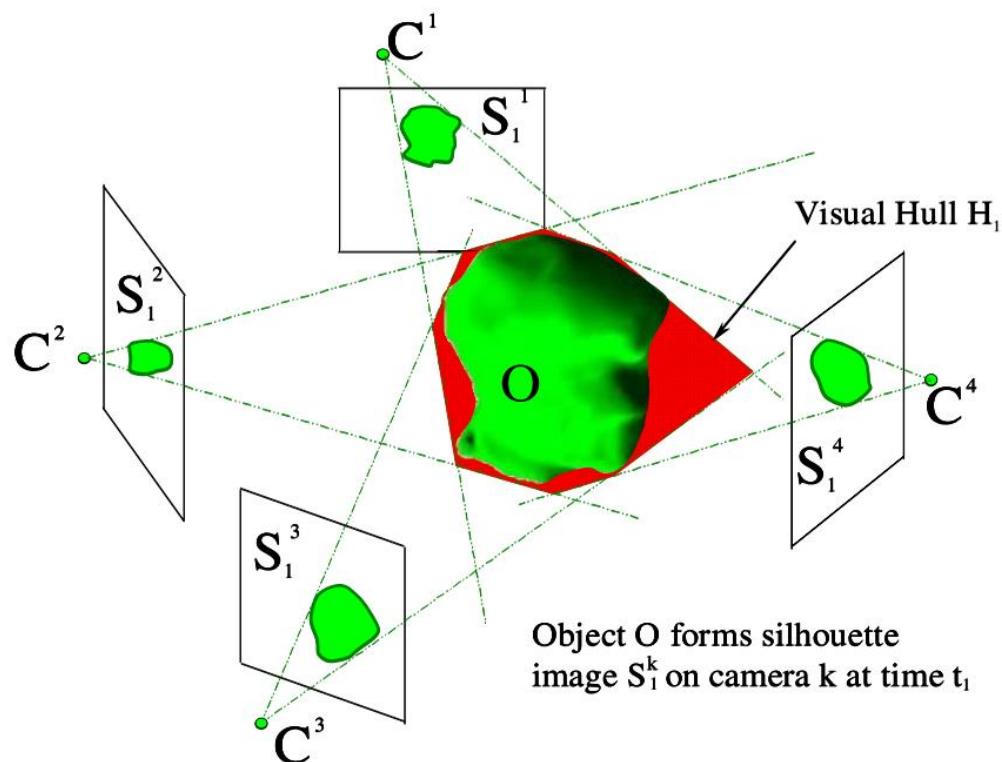


Figure 1. From multi-camera videos to dynamic textured 3D models



Exact Polyhedral Visual Hulls (EPVH) Algorithm

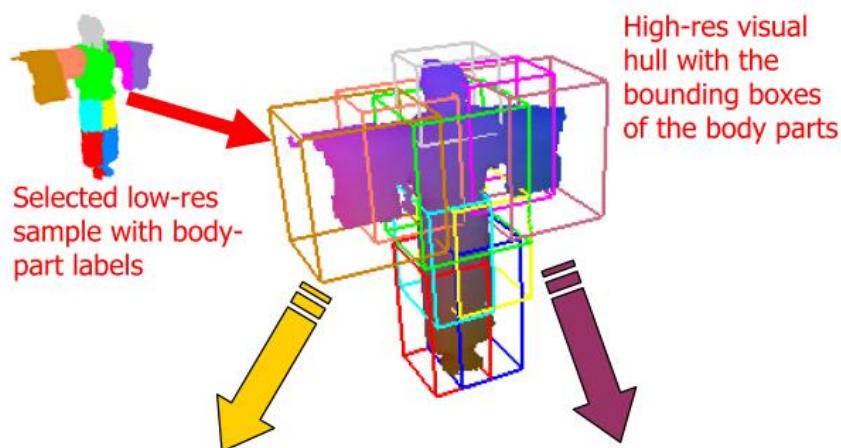
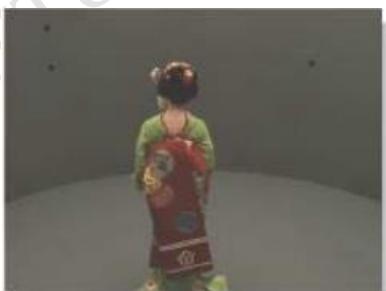
■ More examples...





Exact Polyhedral Visual Hulls (EPVH) Algorithm

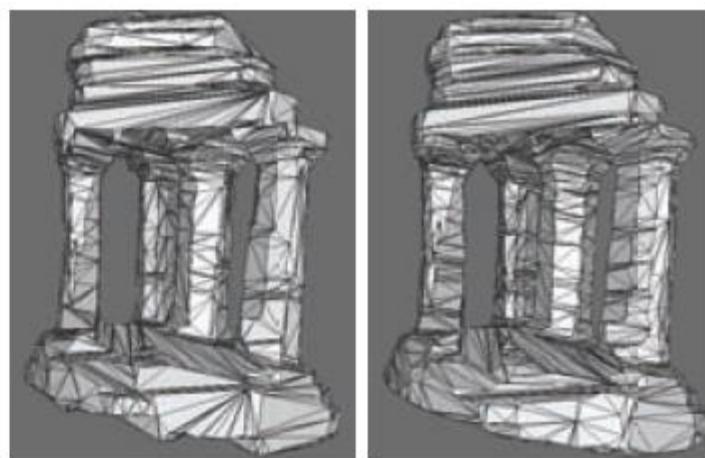
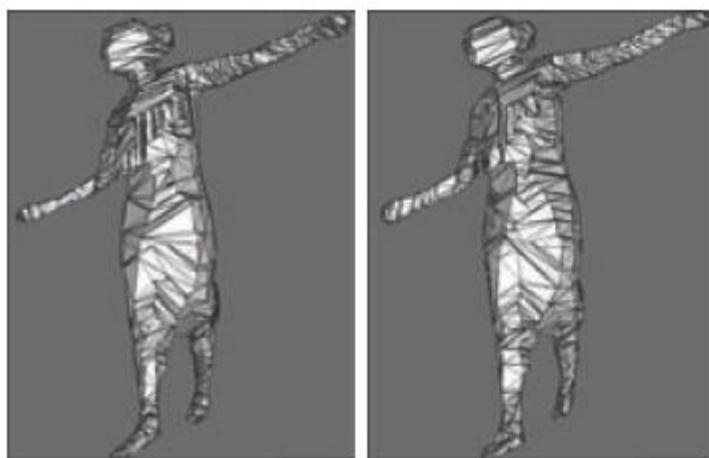
■ More examples...





Exact Polyhedral Visual Hulls (EPVH) Algorithm

■ More examples...





Exact Polyhedral Visual Hulls (EPVH) Algorithm

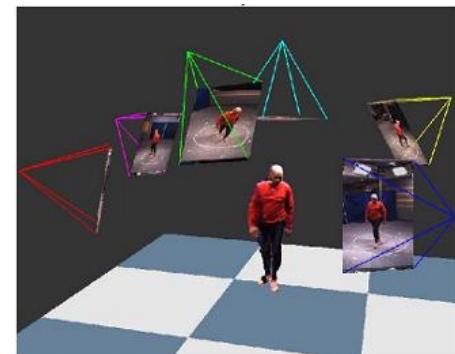
- Famous team
 - 2003 VVG, **Real-Time Capture, Reconstruction and Insertion into Virtual World of Human Actors** (INRIA)
 - 2007 SIGGRAPH **Grimage Markerless 3D Interactions**





Exact Polyhedral Visual Hulls (EPVH) Algorithm

- Famous team
 - 2006 ICVS **A Mixed Reality Environment for Interactions**



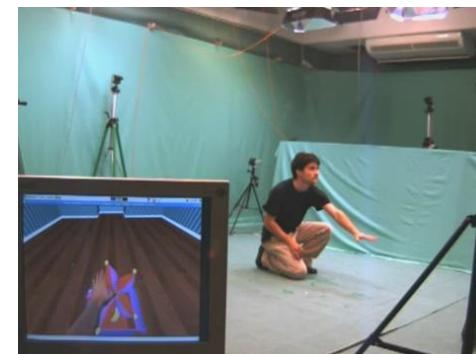
- 2004 IPT **Marker-less Real Time 3D Modeling for Virtual Reality**





Exact Polyhedral Visual Hulls (EPVH) Algorithm

■ Video demo example



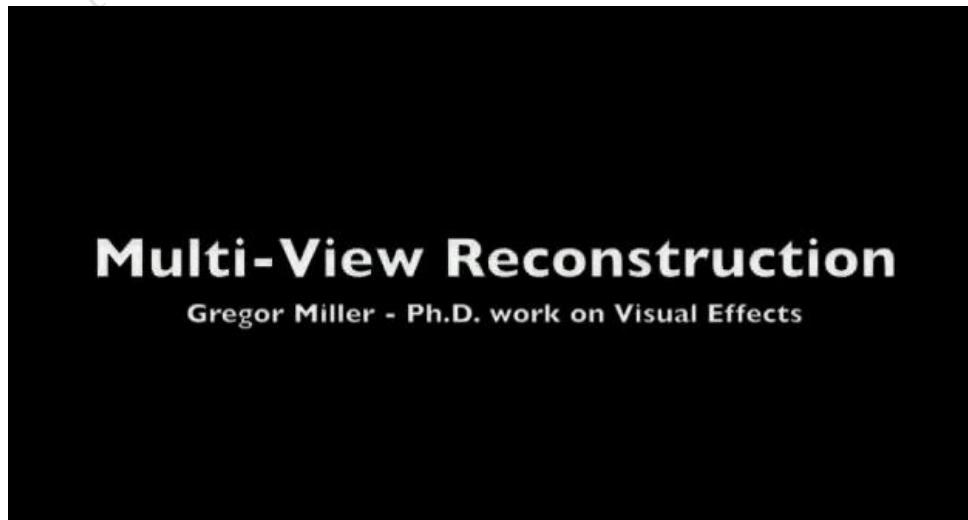
<http://www.youtube.com/watch?v=ZG53YDVW1Ik>

http://www.youtube.com/watch?v=YsbIx_B_KU8E



Application and video demo

■ Motion analysis





Application and video demo

■ Motion analysis—cont.





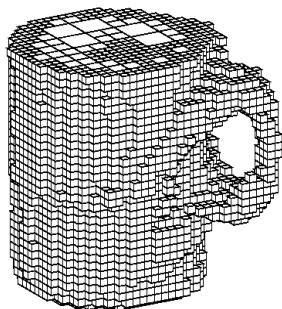
Background: limitation of modern 3D scanner

- Modern 3D scanners have difficult on obtaining intricate objects, particularly for translucent material, , thin and tiny structures.
- The surface properties, such as reflectance, color and roughness, usually affect the scanning accuracy.
- For conventional 3D scanners, the post-production, including noise removal, hole-filling, merging and integration, are time-consuming and unpredictable.

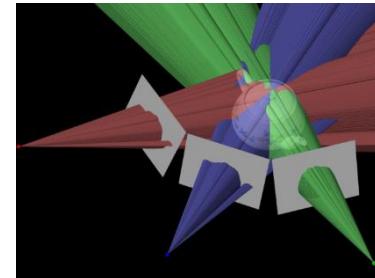


Why “shape from silhouette(SFS)” works

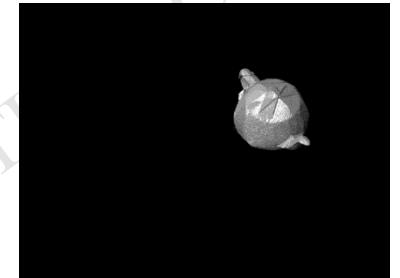
- A famous algorithm called “visual hull” is widely used for carving the 3D shapes from their silhouettes.
- To avoid inconsistent 3D estimation due to surface properties.
- No systematic error from merging procedures



Voxel based visual hull



Exact polyhedral visual hull
(EPVH)





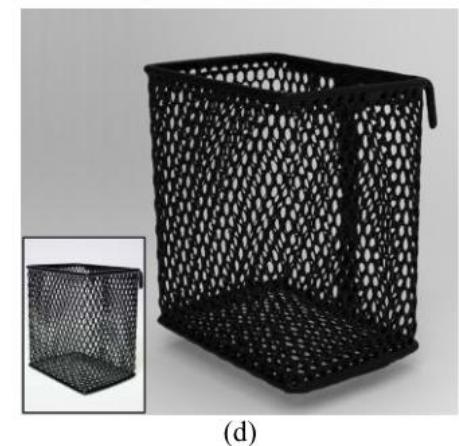
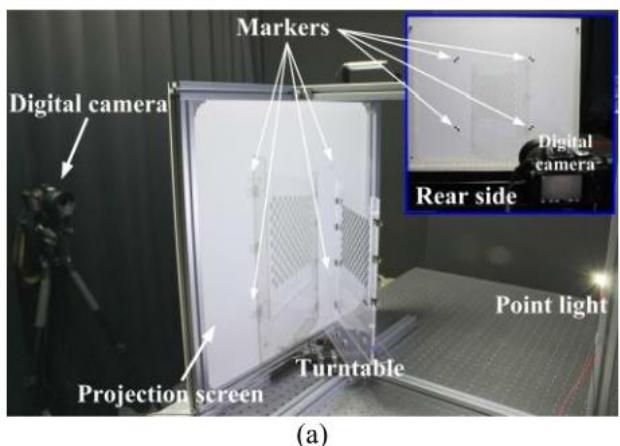
Why we use shadows for SFS?

- To extend to the limitation of DOF in cameras.
- To obtain distinct silhouettes rather than those from image post processing.
- Much easier to identify the silhouette boundary, particularly for translucent or intricate objects.
- Based on off the shelf components, affordable, and no optical design is required.



Visual hull for 3D reconstruction

- Translucent object and intricate object





Visual hull for 3D reconstruction

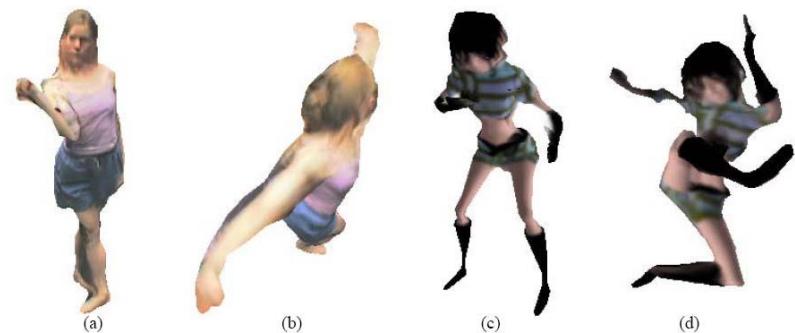
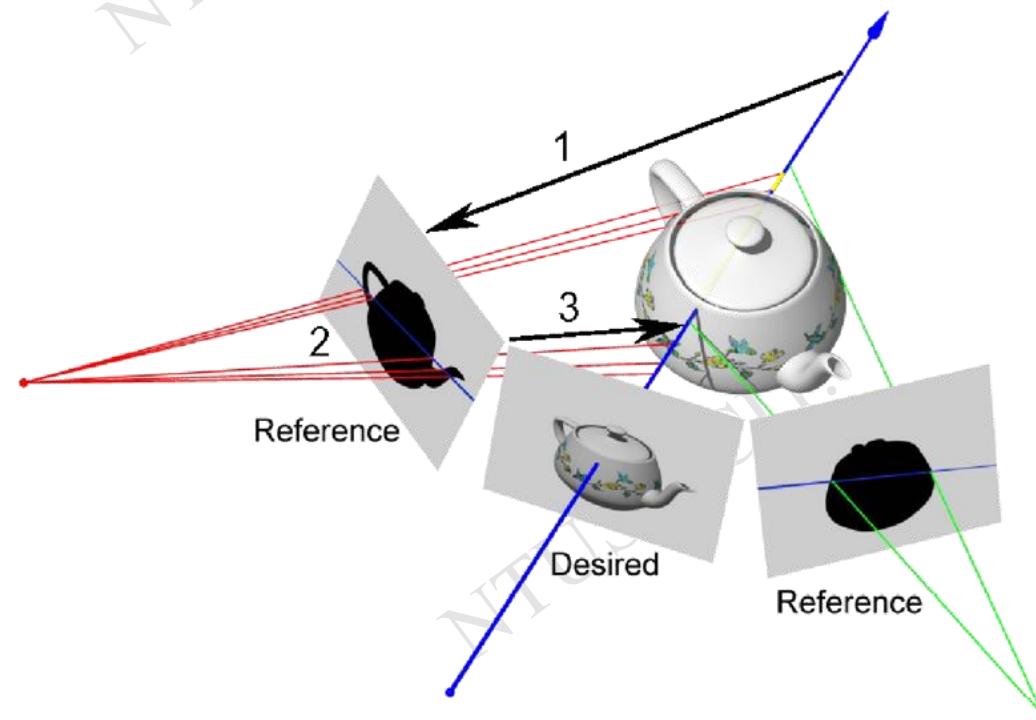


(72 shadow images and
72 color images)



Image-based rendering

■ View-dependent multi-view texture mapping

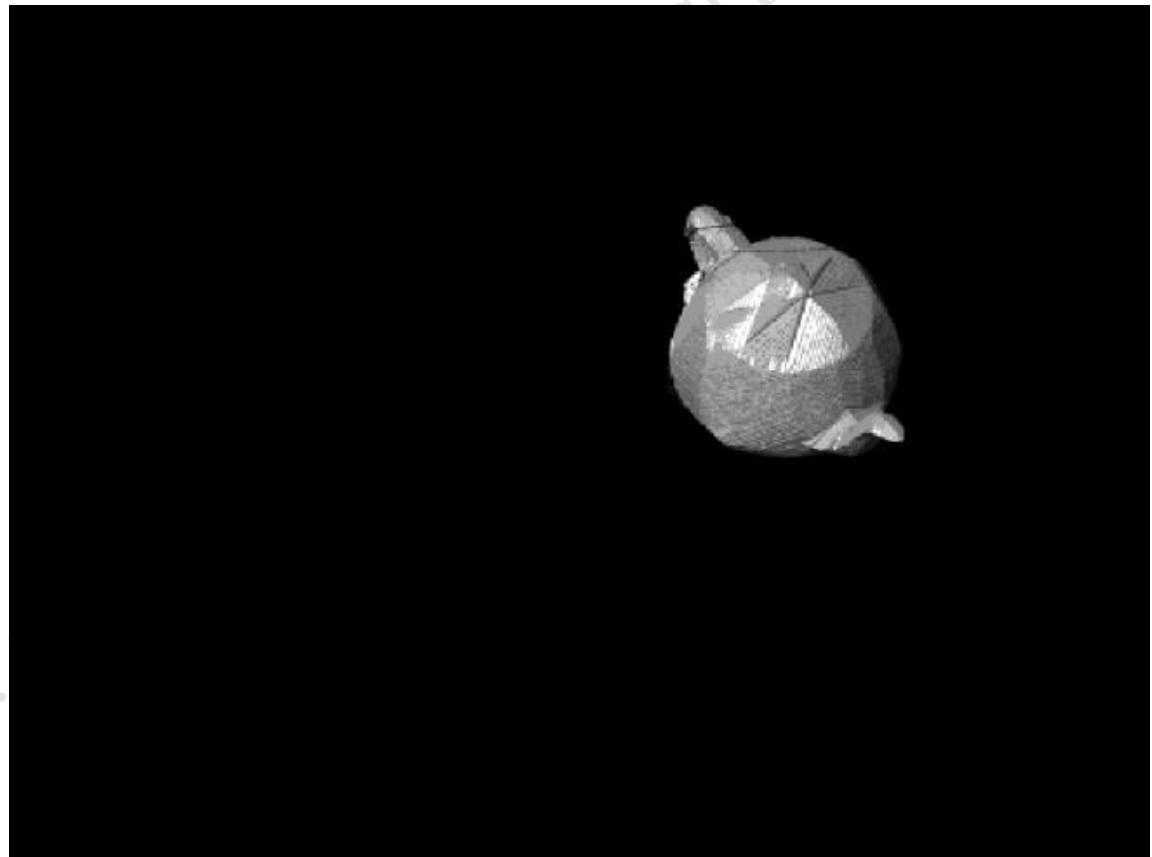


W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan, “Image-based visual hulls,” *Proc. 27th Annu. Conf. Comput. Graph. Interact. Tech. - SIGGRAPH ’00*, pp. 369–374, 2000.
<http://people.csail.mit.edu/wojciech/IBVH/index.html>



Image-based rendering

■ Example



W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan, “Image-based visual hulls,” *Proc. 27th Annu. Conf. Comput. Graph. Interact. Tech. - SIGGRAPH '00*, pp. 369–374, 2000.
<http://people.csail.mit.edu/wojciech/IBVH/index.html>



Commercial product (based on Visual Hull)

- Strata 3D
- 3D SOM

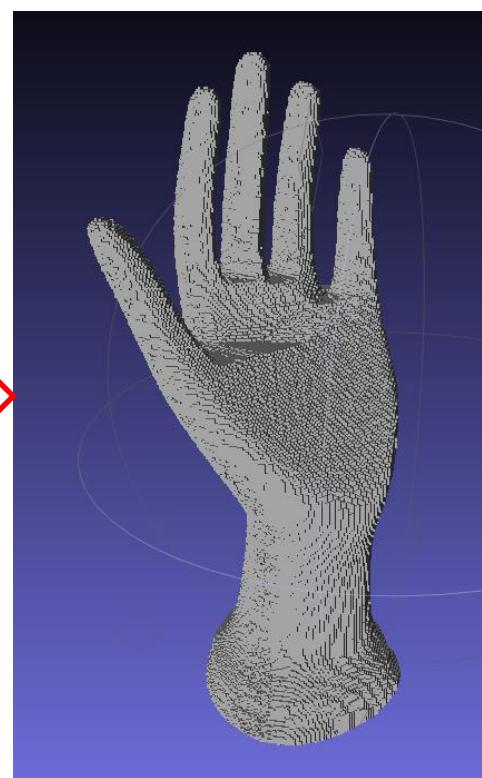
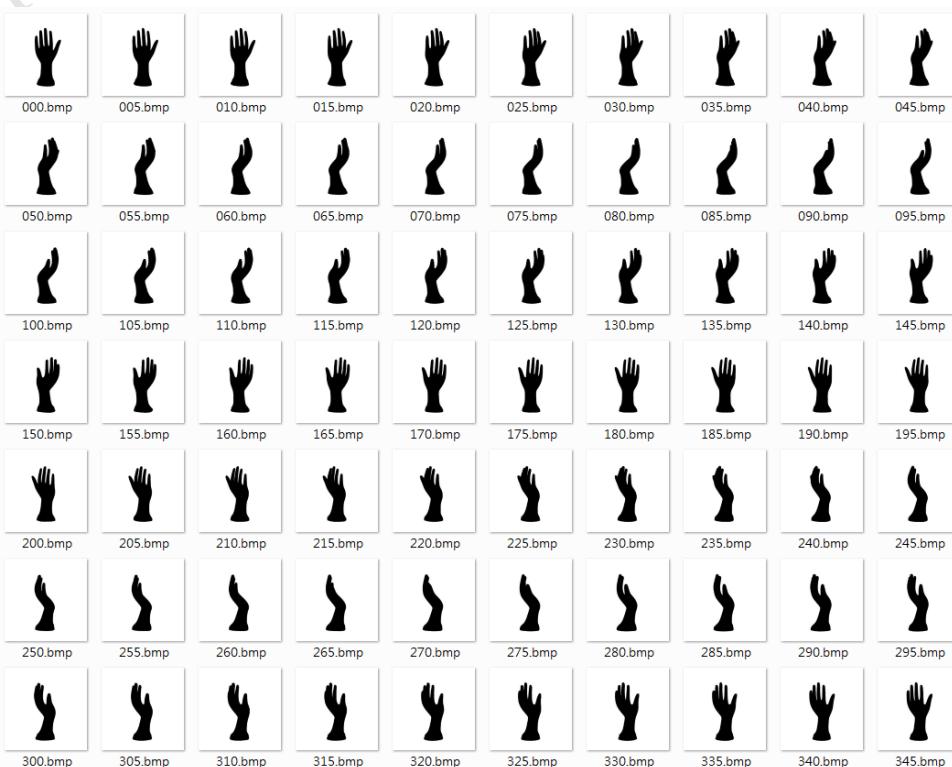
The screenshot shows the Strata 3D Software and Services page. It features a video player showing a camera lens, a "Buy Strata Foto 3D CX 2 Online Now!" button, and sections for "Features & Benefits" (Sophisticated Camera Support, Automated 3D Space Orientation, Feature Pinning - Allows the Use of Any Photo Set), "Reviews & Awards", and "Case Studies & Videos". On the left, there's a sidebar with links like "Solutions", "Get Info", "Live Chat (Offline)", "Email a Question", "Give Us a Call", "Get the Strata Bulletin", and "Email Address". A "Send" button is also present.

The screenshot shows the Welcome to 3D Software Object Modeller Pro page. It includes a "Version 3" section with a "BUY NOW pricing and sales" button, a "FREE trial" section with a "Download a 14 day free trial..." button, and a "3DSOM News" section with a "Sign up for our free newsletter..." button. Below these are sections for "View Showreel", "What's New" (listing recent releases like a new version with WebGL and material channels, and the launch of a 3D anatomy site), and a "3DSOM Pro Software" sidebar. A large 3D model of a shell is displayed in the center, with various interaction tools at the bottom. A quote from a user is visible on the left.



Multi-view in 3D reconstruction (silhouette)

■ Shape from silhouette 理論 (voxel)

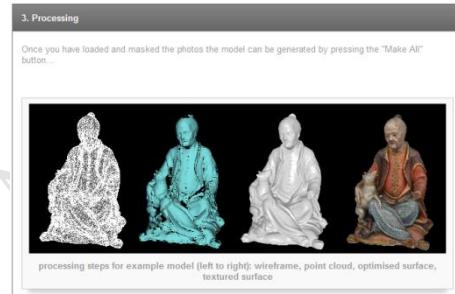
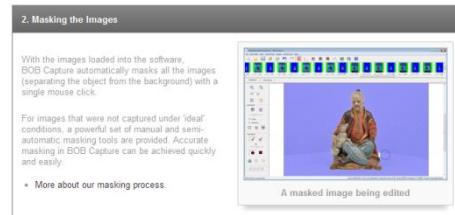
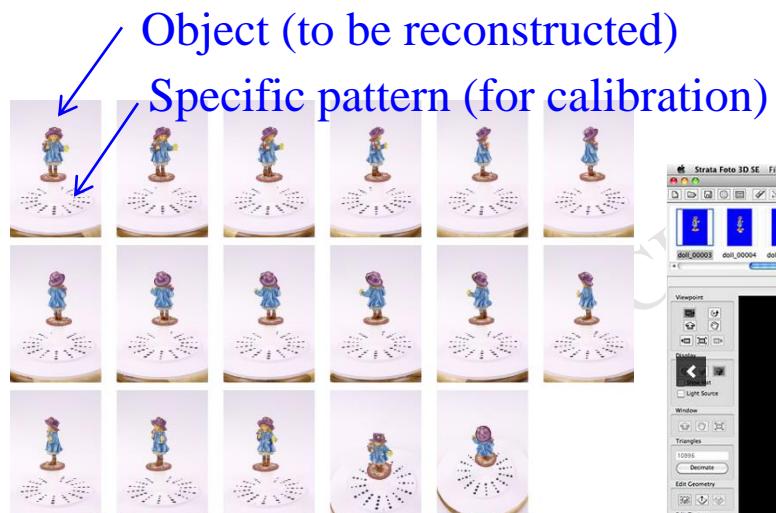




Multi-view in 3D reconstruction (silhouette)

Multiview Visual-Hull based algorithm

- Commercial production: Strata, 3DSOM.
- Put objects on a pattern then take multiple-view pictures





Multi-view in 3D reconstruction (silhouette)

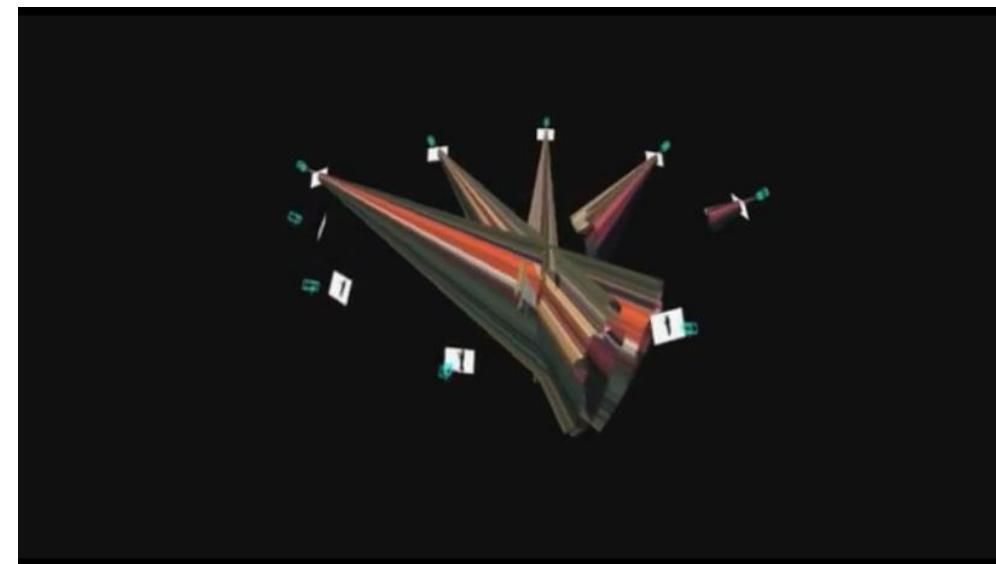
■ Reconstruction Example (by Strata 3D)





Multi-view in 3D reconstruction (silhouette)

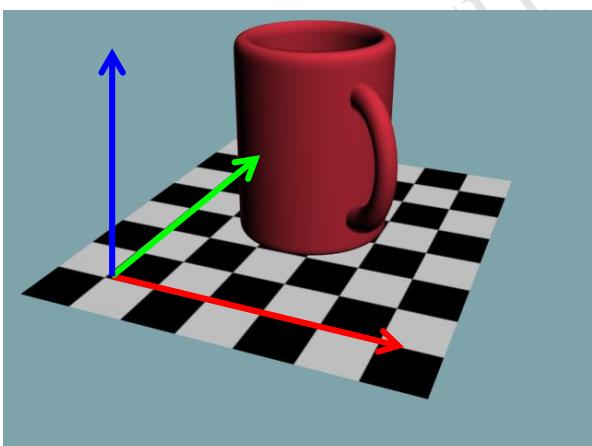
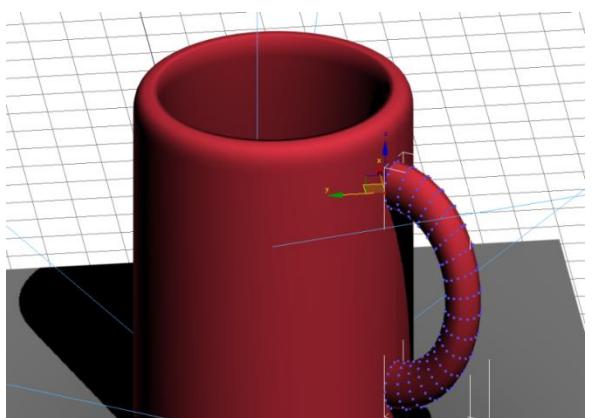
- 4D data (action replay)
- Image based rendering





Take practice

■ Excise



For Example

Camera1 : extrinsic parameter (3x4)

0.9244	0.3815	-0.0002	-84.4407
0.1780	-0.4319	-0.8842	19.6768
-0.3374	0.8173	-0.4671	322.7263

Camera1 : intrinsic parameter (3x3)

798.921754	0.000000	319.074703
0.000000	799.027372	241.213600
0.000000	0.000000	1.000000

