

# Lecture 8

## Active stereo & Volumetric stereo



- Active stereo
  - Structured lighting
  - Depth sensing
- Volumetric stereo:
  - Space carving
  - Shadow carving
  - Voxel coloring

**Reading:**

[Szeliski] Chapter 11 "Multi-view stereo"  
S. Savarese, M. Andreetto, H. Rushmeier, F. Bernardini and P. Perona,  
3D Reconstruction by Shadow Carving: Theory and Practical Evaluation,  
International Journal of Computer Vision (IJCV) , 71(3), 305-336, 2006  
Seitz, S. M., & Dyer, C. R. (1999). Photorealistic scene reconstruction by  
voxel coloring. *International Journal of Computer Vision*, 35(2), 151-173.

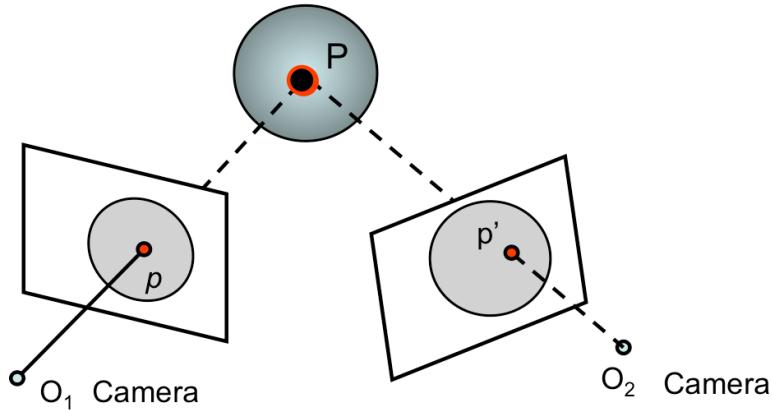
Silvio Savarese

Lecture 7 -

20-Apr-16

In this lecture, we'll first discuss another framework for describing stereo systems called active stereo, and then introduce the problem of volumetric stereo, along with three popular methods for solving this problem.

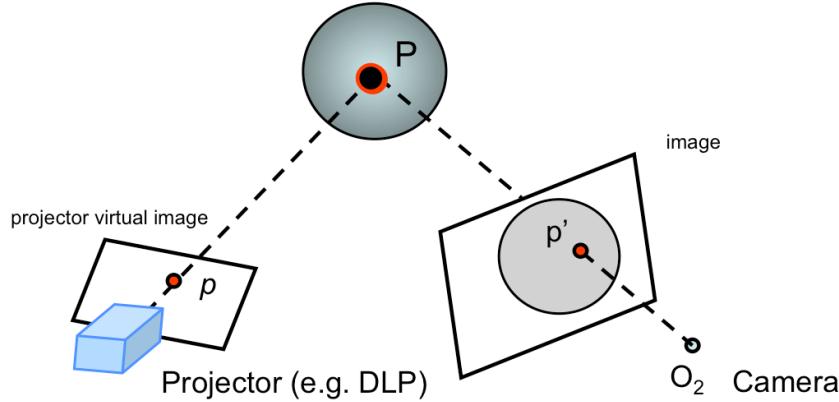
## Traditional stereo



What's the main problem in traditional stereo?  
We need to find correspondences!

In traditional stereo, the main idea is to use corresponding points  $p$  and  $p'$  to estimate the location of  $P$  in 3D by triangulation. A key challenge here, however, is to solve the correspondence problem in the first place – how do we know that  $p$  corresponds to  $p'$ , if multiple 3D points are present in the scene?

## Active stereo (point)



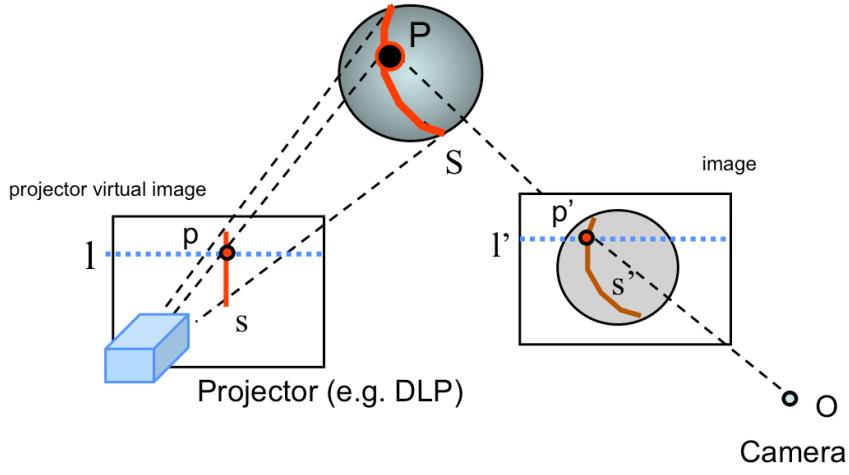
Replace one of the two cameras by a projector

- Single camera
- Projector geometry calibrated
- What's the advantage of having the projector? Correspondence problem solved!

We now introduce a technique which can help mitigate this problem. The main idea is to replace one of the two cameras with a device that is capable of projecting a “mark” into the object that can be easily identified from the second camera. The projector-camera pair defines the same epipolar geometry that we introduced for camera pairs, whereby the image plane of the camera 1 is now replaced by a “projector virtual plane”. In the example above, the projector is used to project a point (indicated by  $p$ ) into the object which produces  $P$ .  $P$  is then observed from the camera 2. Because we know what we are projecting (e.g., we know the position of  $p$  in the projector virtual image as well as its color and the intensity value), we can easily discover the corresponding observation in the second camera  $p'$ . Because we are using a projector and a camera (instead two cameras), this system is called **active stereo**.

Notice that in order for this to work we need to have both camera and projector calibrated (more on this in a few slides).

## Active stereo (stripe)

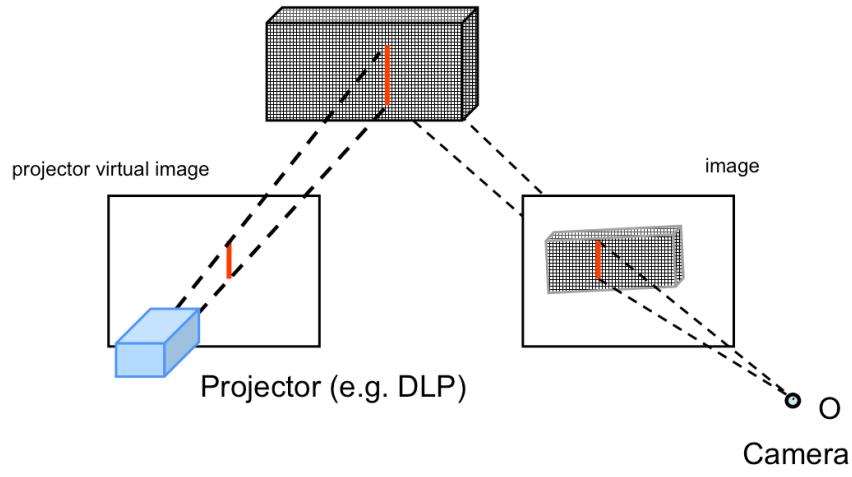


- Projector and camera are parallel
- Correspondence problem is solved!

A common strategy is to project a vertical stripe  $s$  instead of a single point (e.g., the red line in the projector virtual plane). The projection of  $s$  into the object is  $S$  and its observation in the camera is  $s'$ . If camera and projector are parallel (or get rectified), we can discover corresponding points very easily. (e.g. a point  $p$  that lies on  $s$  and the corresponding point  $p'$  on  $s'$ ) by simply intersecting  $s'$  with the horizontal epipolar line (blue dashed line  $l'$ ) defined by  $p$ . In this example, the result of this intersection is indeed the point  $p'$ . By triangulating corresponding points, we can reconstruct all the 3D points (e.g.  $P$ ) on  $S$ .

The entire shape (visible) of the object can be recovered (scanned) by projecting a sequence of stripes (e.g., at different locations along the x-axis of the projector virtual image) and repeating the operation described above for each of such stripes.

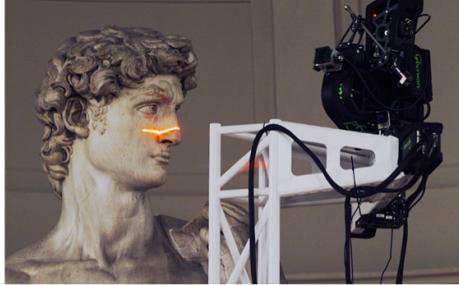
## Calibrating the system



- Use calibration rig to calibrate camera and localize rig in 3D
- Project stripes on rig and calibrate projector

An active stereo system can be easily calibrated by using the same techniques discussed in lecture 3: We can first calibrate the camera using a calibration rig. Then, by projecting (known) stripes or patterns into the calibration rig, and by using the corresponding observations in the camera, we can setup constraints for estimating the projector intrinsic and extrinsic parameters (that is, its location and pose w.r.t. the camera).

## Laser scanning



Digital Michelangelo Project  
<http://graphics.stanford.edu/projects/mich/>

- Optical triangulation
  - Project a single stripe of laser light
  - Scan it across the surface of the object
  - This is a very precise version of structured light scanning

Source: S. Seitz

In the next few slides, we show examples of active lighting methods that can be used to successfully recover the shape of unknown objects. The approach proposed by M. Levoy and his students at Stanford back in 2000 uses a laser scanner instead of a projector. The method was used to recover the shape of the Michelangelo's Pieta' with sub-millimeter accuracy.

## Laser scanning



The Digital Michelangelo Project, Levoy et al.

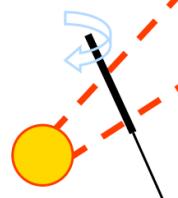
Source: S. Seitz

An example of reconstruction.

## Active stereo (shadows)



J. Bouguet & P. Perona, 99



Light source

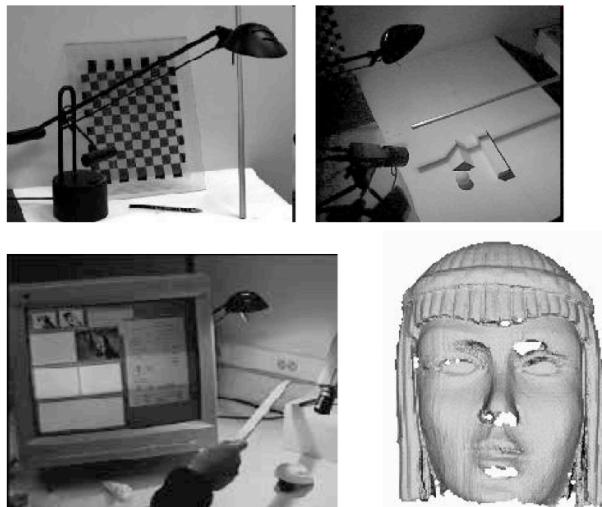


O

- 1 camera, 1 light source
- very cheap setup
- calibrated the light source

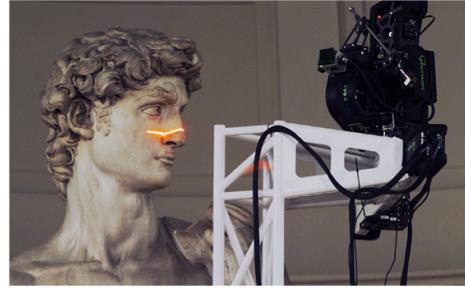
An alternative approach, which uses a much cheaper setup (but it produces much less accurate results), leverages shadows to produce active patterns to the object we want to recover. The shadow is cast by moving a stick located between a light source and the object. In order to make this to work, one needs to “calibrate” the light source (i.e. estimate its location in 3D w.r.t. the camera).

## Active stereo (shadows)



Here we see some snapshots of the system.

## Limitations of Laser scanning



- Slow
- Cannot capture deformations in time

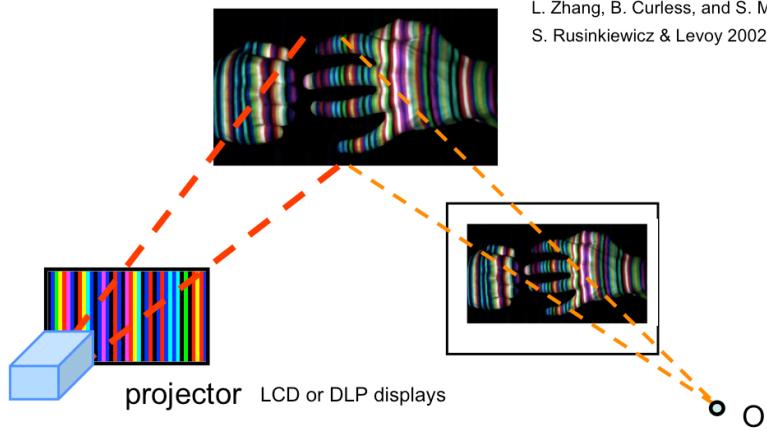
Source: S. Seitz

What are the main limitations of techniques based on laser scanning?

They are slow and cannot capture deformation in time

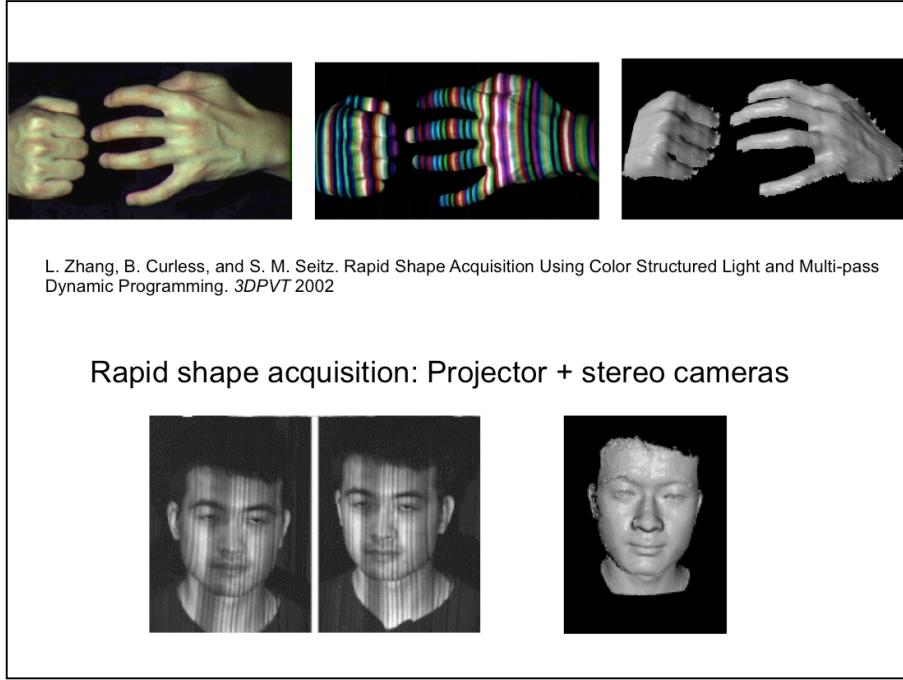
## Active stereo (color-coded stripes)

L. Zhang, B. Curless, and S. M. Seitz 2002  
S. Rusinkiewicz & Levoy 2002



- Dense reconstruction
- Correspondence problem again
- Get around it by using color codes

Finally, a successful extension was introduced to obtain a dense reconstruction of the object from just one single image or frame. The idea is to project a pattern of stripes to the entire visible surface of the object, instead of projecting a single stripe. Each stripe is associated to a different color code. Color codes are designed such that they stripes can be uniquely identified from the image.



The advantage of this approach is that it can be used for recovering the 3D structure of an object from a video sequence (and thus for capturing moving objects – a facial expression in this example).

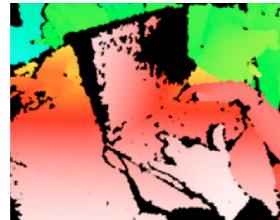
## Active stereo – the Kinect sensor



- Infrared laser projector combined with a CMOS sensor
- Captures video data in 3D under any ambient light conditions.



Pattern of projected infrared points to generate a dense 3D image



Depth map

Source: wikipedia

The concept behind active stereo is used in modern depth sensors such as the Kinect.

Such sensor consists of an infrared laser projector combined with a monochrome CMOS sensor, which captures video data in 3D under any ambient light conditions.

Some results are shown in the bottom figure.

# Lecture 8

## Active stereo & Volumetric stereo



- Active stereo
  - Structured lighting
  - Depth sensing
- Volumetric stereo:
  - Space carving
  - Shadow carving
  - Voxel coloring

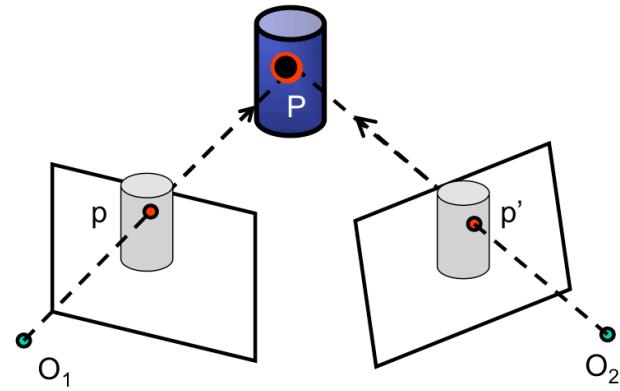
Silvio Savarese

Lecture 7 -

20-Apr-16

We now introduce and study the volumetric stereo problem. We start with the traditional stereo setup and then contrast the volumetric stereo problem with it. We then describe three popular methods for implementing a system based on volumetric stereo: Space carving, Shadow carving and Voxel carving

## “Traditional” Stereo

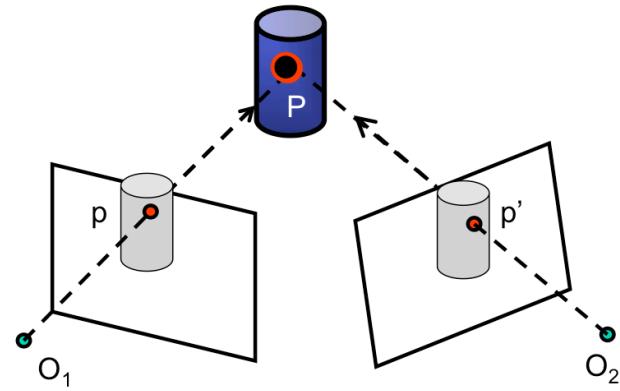


Goal: estimate the position of  $P$  given the observation of  $P$  from two view points

Assumptions: known camera parameters and position ( $K, R, T$ )

In the traditional stereo setup, we seek to estimate the 3D world coordinate of a point  $P$ , given its observations  $p$  and  $p'$  as imaged from two cameras with centers at  $O_1$  and  $O_2$ . We assume that the camera matrices and their relative position and pose are known.

## “Traditional” Stereo

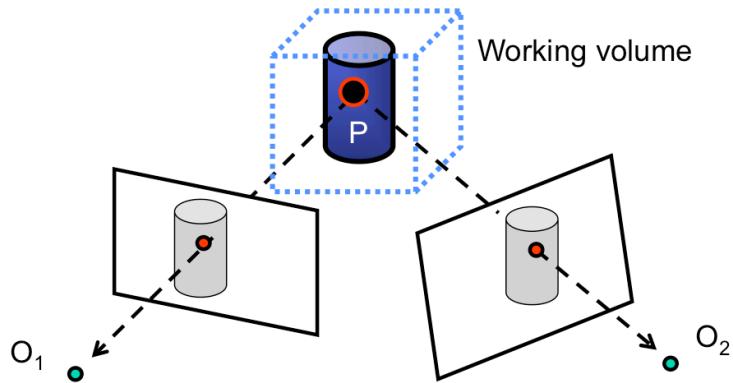


Subgoals:

1. Solve the correspondence problem
2. Use corresponding observations to triangulate

In the traditional stereo, we seek to solve two key sub-problems: i) The correspondence problem – establishing correspondences between points in the two views; ii) Using the so established correspondences to perform triangulation— to find the 3D coordinate of the points under consideration.

## Volumetric stereo



1. Hypothesis: pick up a point within the volume
2. Project this point into 2 (or more) images
3. Validation: are the observations **consistent?**

**Assumptions:** known camera parameters and position ( $K, R, T$ )

The volumetric stereo approach follows a reverse logic through 3 major steps:  
1) We **hypothesize** that a point exists in 3D within a given working volume (e.g., the dashed volume in the figure). 2) We project the hypothesized 3D point into the multiple images. 3) We **validate** (check) if such projections are “consistent” across views and with respect to what we actually observe in the images for all the images. Again, we make the assumption that cameras are calibrated ( $K, R$ , and  $T$  are known) and, thus, the projection of 3D points to images can be computed.

Because it is typical for such techniques to assume that the scene we want to reconstruct is contained by a working volume, these techniques are mostly used for recovering 3D models of objects as opposed for recovering models of entire scenes.

A key question now is: what does it mean “consistent”?

## Consistency based on cues such as:

- Contours/silhouettes → Space carving
- Shadows → Shadow carving
- Colors → Voxel coloring

Depending on the definition of the concept of consistent observations, different techniques can be introduced. For instance, as we shall see next, consistency can be computed using object contours or silhouettes, object self-shadows or object colors. These would lead to 3 major techniques called respectively:

- Space carving
- Shadow carving
- Voxel coloring

# Lecture 8

## Active stereo & Volumetric stereo



- Active stereo
  - Structured lighting
  - Depth sensing
- Volumetric stereo:
  - Space carving
  - Shadow carving
  - Voxel coloring

**Reading:**  
[\[Szeliski\]](#) Chapter 11 "Multi-view stereo"

Silvio Savarese

Lecture 7 -

20-Apr-16

We start by discussing the cues that come from contours and silhouettes and the related space carving approach.

## Contours/silhouettes

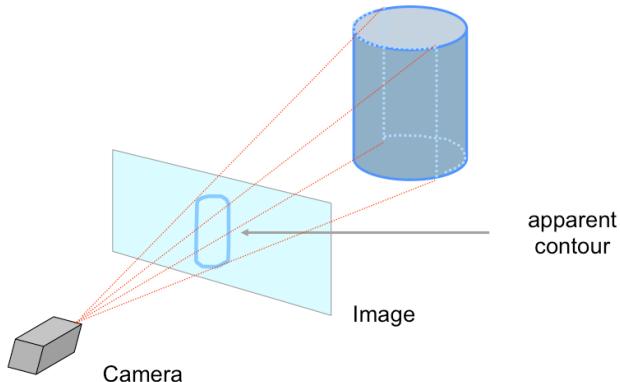
- Contours are a rich source of geometric information



By looking at this scene (city skyline), even if we almost only see the contours of these buildings, we can still get an idea of their shape; analyzing how we can use contours for implementing a volumetric stereo approach is the focus in this part of the lecture

## Apparent Contours

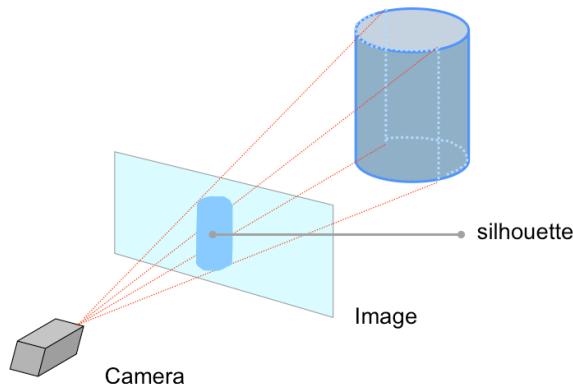
- Projection of the set of points on the object surface which separate the visible and occluded parts on the surface  
[sato & cipolla]



Let's start with some definitions. An apparent contour is defined as the projection of the set of all points that separate the visible and occluded parts of a surface (for instance, see dashed cyan lines on the cylinder). An example of apparent contour is the bold cyan curve in the image plane.

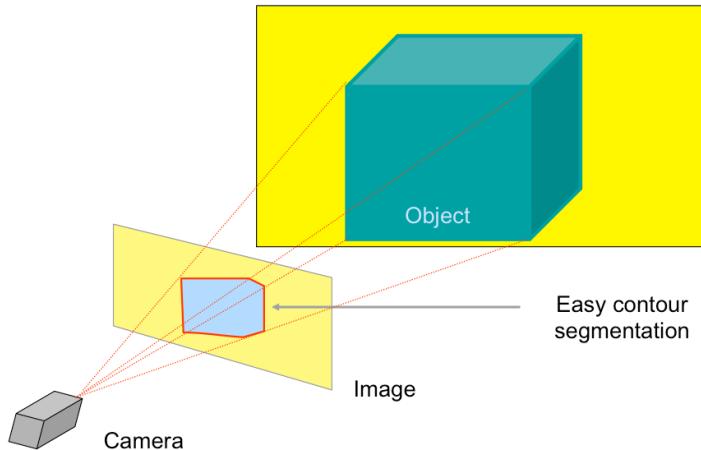
## Silhouettes

A silhouette is defined as the area enclosed by the apparent contours



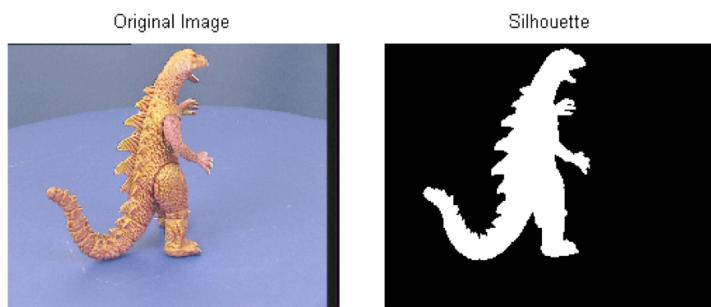
A silhouette is defined as the area enclosed by the apparent contours (cyan area in the image).

## Detecting silhouettes



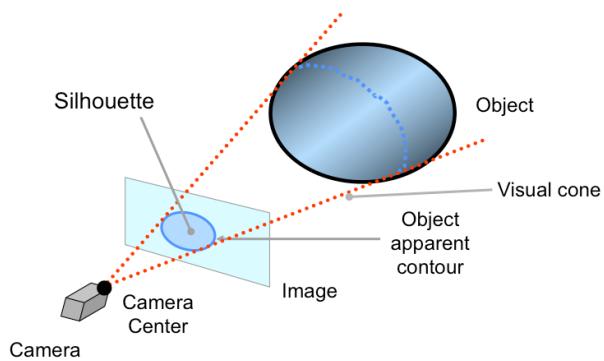
One practical advantage of working with silhouettes (or apparent contours) is that they can be easily detected in images if we have control of the luminance values of the background (e.g., the yellow rectangular region) that is behind the object we want to reconstruct. For instance, one can imagine to put the object we want to reconstruct in front of a blue or yellow screen which can help segment the object from the background.

## Detecting silhouettes



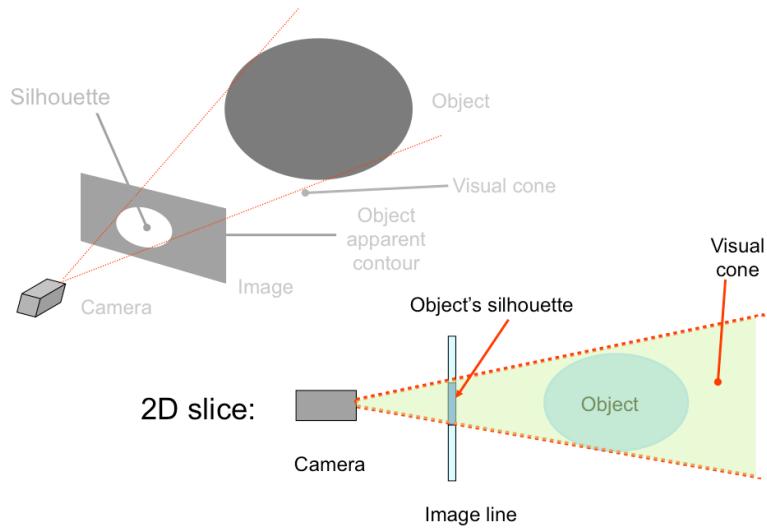
As this example shows, segmenting the object (dinosaur) from this blue background is easy and leads to the results on the right hand side.

## How can we use contours?



In the next few slides, we will study how contours can be used to obtain a measurement of consistency to solve the volumetric stereo problem. The slide summarizes the main components and definitions of the volumetric stereo systems. We highlight here the concept of visual cone which is the envelope surface defined by the center of the camera and the object contour in the image plane. Note that, by construction, it is guaranteed that the object will lie completely inside this visual cone.

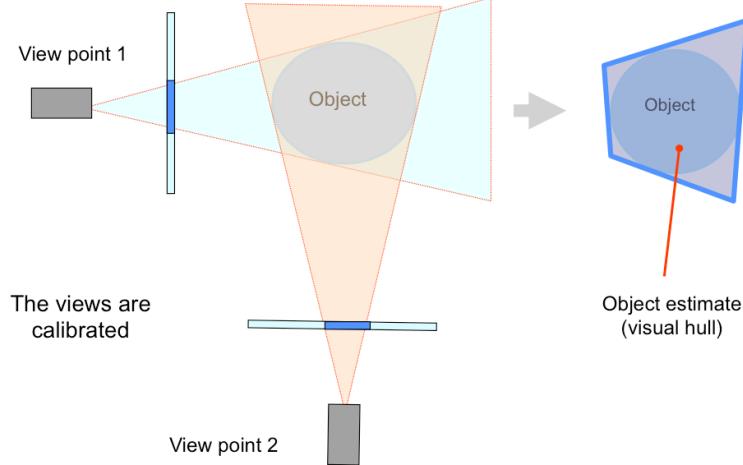
## How can we use contours?



To simplify our discussion, we consider a 2D version of the setup – that is, a 2D slice that traverses the 3D volume and passes through the camera center. In the 2D slide model an image becomes a 1D image line, a silhouette a 1D segment and the visual cone becomes a sector (red dashed lines) as shown in the lower part of the slide. We will discuss the space carving approach by reasoning in the 2D slice; the generalization to the 3D case is obvious by considering the fact that the 2D slice can sweep the entire volume.

If the camera is calibrated and we can measure the silhouette or object apparent contour, then we can also calculate the visual cone. Even if we don't know the shape of the object and where it is located yet, we know that it must be enclosed by the visual cone.

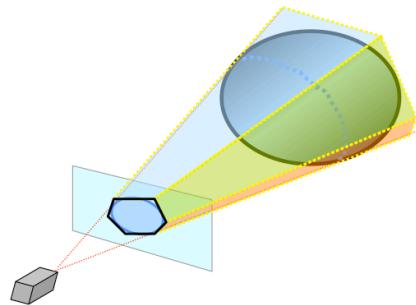
## How can we use contours?



Assume now that the same object is observed from an view point (in general, from multiple views). Assume that for each view we can compute the silhouette and that the cameras are all calibrated (we know the rotation and translation transformation relating all the cameras). This allows us to compute a visual cone for each view. Since we know that the object must be contained by the visual cone, an estimate of the shape of the object can be obtained by intersecting all the visual cones. Such intersection is also called **visual hull**.

## How to perform visual cones intersection?

- Decompose visual cone in polygonal surfaces  
(among others: Reed and Allen '99)

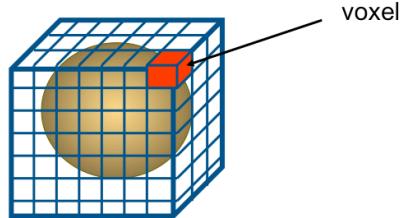


How do we implement this operation of intersecting visual cones in practice? One way to perform an intersection of visual cones is to decompose each cone into polygonal surfaces. Intersecting polygonal surfaces is easier in general but not trivial (computationally wise) and can be very sensitive to measurement noise for small baselines. Next we introduce an alternative approach which is much more robust and can be computationally very efficient: **space carving**.

## Space carving

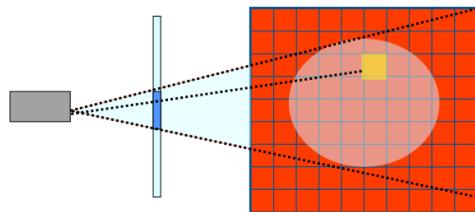
[ Martin and Aggarwal (1983) ]

- Using contours/silhouettes in volumetric stereo, also called **space carving**



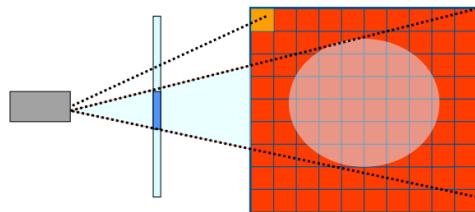
Space carving attempts to find the 3D shape of an unknown arbitrary object given multiple images of the object from different viewpoints. In space carving, the working volume (cube) is decompose in sub volumes (sub-cubes) also called **voxels**. The main idea of space carving is to explore all the voxels in the working cube and evaluate whether a voxel belongs to the actual object (in this case, it is labeled as full) or not (in this case, it is labeled as empty). The process of deciding whether a voxel is full or empty is based on the consistency test which is computed for all the observations (views) as we explain next.

## Computing Visual Hull in 2D



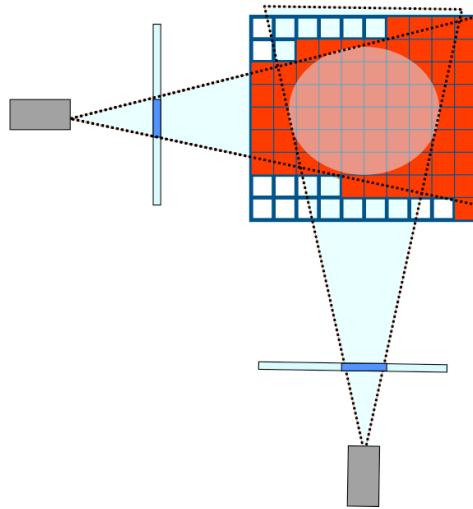
The intuition we follow here is that voxels that fell in the visual cone (and thus are projected into the silhouette of the object in the image) are candidates for being full. In this figure, the orange voxel is one of such voxels.

## Computing Visual Hull in 2D



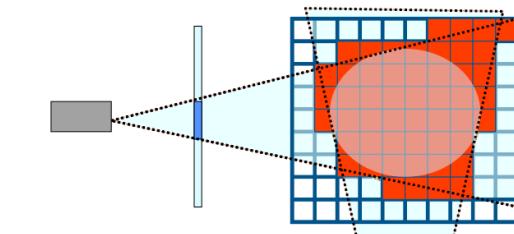
Voxels that do not fall in the visual cone (and thus are not projected into the silhouette of the object in the image) are guaranteed to be empty. In this figure, the orange voxel is one of such voxels.

## Computing Visual Hull in 2D



If we have more than one camera observing the same object and the cameras are calibrated (with respect to each other), we can generalize this operation for all the cameras: In order for a voxel to be full, it must be projected into the object silhouette for all the cameras (images). A voxel is guaranteed to be empty if there exists at least one view for which the voxel gets projected outside the corresponding object silhouette.

## Computing Visual Hull in 2D



**Visual hull:**  
an upper bound estimate

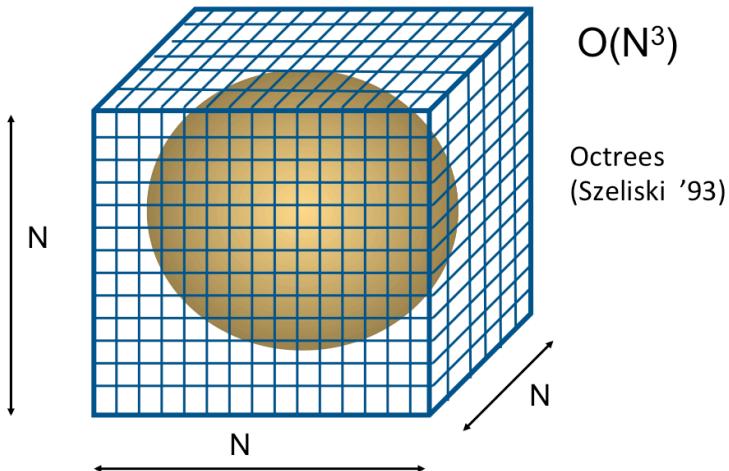
Consistency:

A voxel must be projected into a silhouette in each image

We can repeat this operation for all the voxels in the working volume which produces, in practice, an approximation of the intersection of all of the visual cones associated to the cameras. All the voxels colored in red are labeled as “full” and do satisfy the property that they are projected into an object silhouette for all the cameras (images). The final result is the object visual hull which is an upper bound estimate of the actual object shape (i.e., while no part of the object can lie outside this hull, there are parts of the hull that may not contain any part of the object).

The consistency check that we perform here is whether or not the voxel is projected into a silhouette in each of the images.

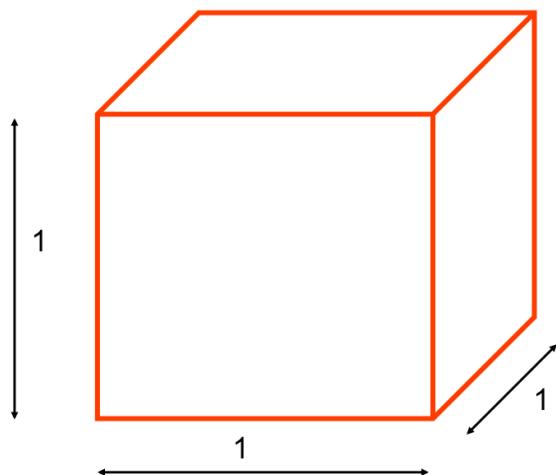
Space carving has complexity ...



If we choose to work with a grid size of  $N \times N \times N$  for the 3D space, then space carving has a complexity of  $O(N^3)$ . This is because we need to evaluate the consistency check at each voxel in space (and there are  $N^3$  voxels in total).

We can employ octrees to reduce the complexity of space carving. In the next few slides, we study how octrees can help reduce the complexity.

## Complexity Reduction: Octrees

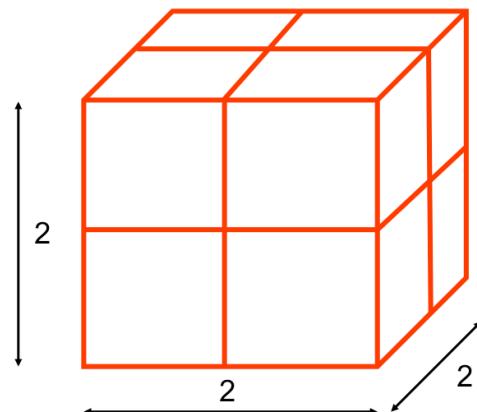


An octree is a tree-based data structure, in which there are 8 children for each internal node. They are often used to partition the 3D space by recursively subdividing it into eight octants. Octrees are the three-dimensional analog of quadtrees.

The slide shows the original working cube (with no octree decomposition yet).

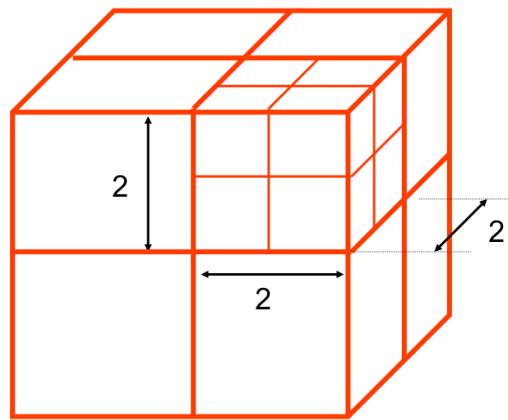
## Complexity Reduction: Octrees

- Subdividing volume in sub-volumes of progressive smaller size



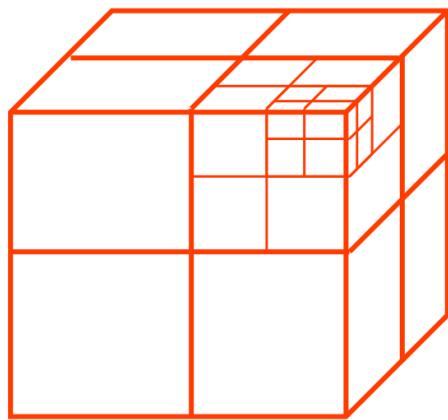
The figure shows the working cube subdivided into eight octants.

## Complexity Reduction: Octrees



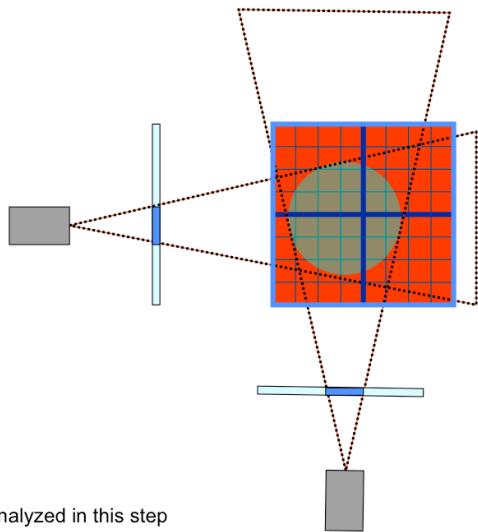
The figure shows the upper-front-right octant that is further subdivided into eight octants.

## Complexity Reduction: Octrees



The idea is to progressively sub-divide the volume into octans (sub-volumes) using the octree decomposition scheme. We will now apply space carving to such octree structure.

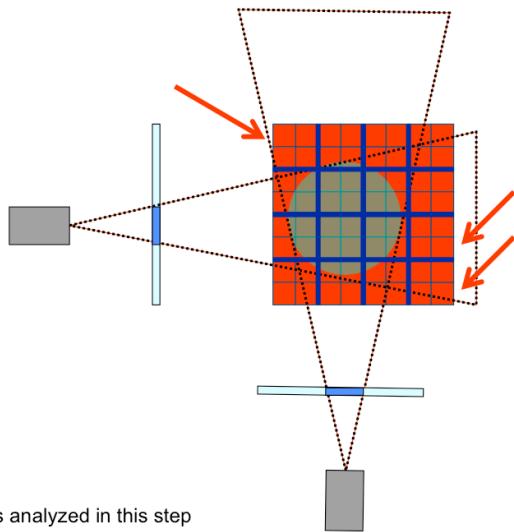
## Complexity reduction: 2D example



Again, let's consider an example in 2D.

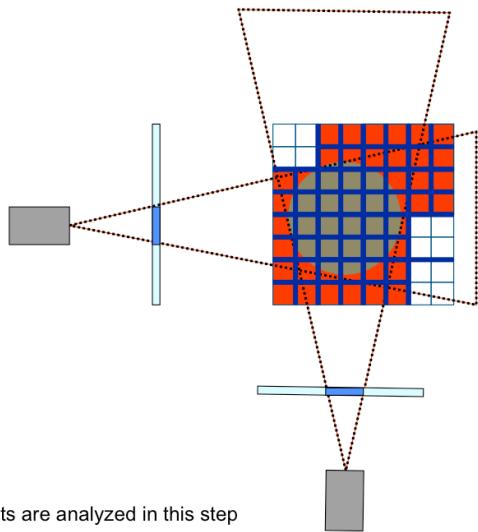
In the first step, we decompose the initial working cube into 4 sub-volumes (blue bold squares) and check if the projections of each of these sub-volumes are consistent. In this example, they are all consistent and all the sub-volumes are labeled as “full”. Notice that 4 elements are analyzed in this step.

## Complexity reduction: 2D example



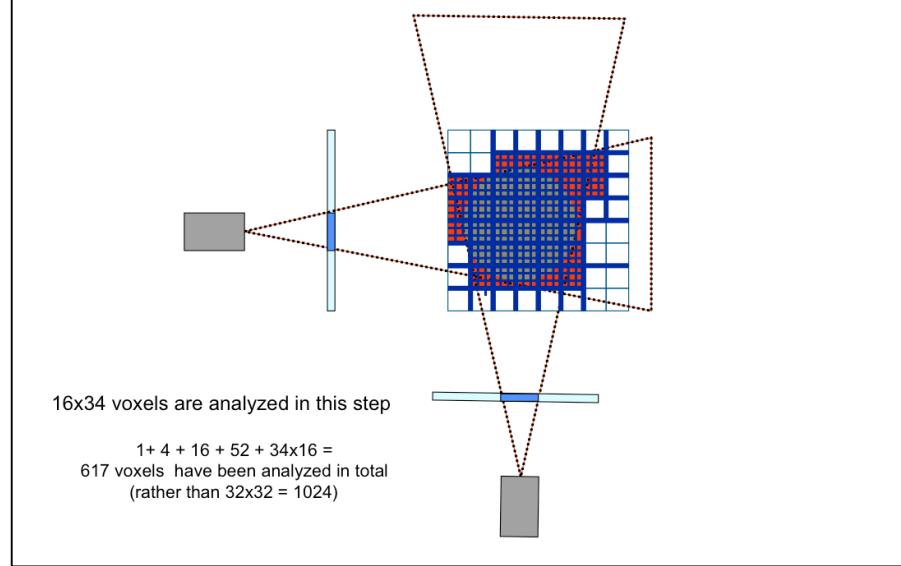
In the next step, we further decompose each sub-volume that is marked as “full” in the previous steps into 4 smaller sub-volumes (bold blue squares) and check if the projections of each of these sub-volumes are consistent. In this case they are all consistent except for the 3 ones marked by the red arrows. These 3 sub-volumes are labeled as “empty”. Notice that we have analyzed 16 elements in this step.

## Complexity reduction: 2D example



Similarly, in the next step, we further decompose each sub-volume that is marked as “full” in the previous steps into 4 smaller sub-volumes (bold blue squares) and check if the projection of each of these sub-volumes are consistent. Those that are not consistent are marked “empty” and eliminated from our analysis. Notice that we have analyzed 52 elements in this step.

## Complexity reduction: 2D example



In the next step, all of the sub-volumes that are labeled as “full” in the previous are further decomposed and processed as we have seen so far. In this case 16x34 elements are analyzed. We assume this is the final step and we have reached the desired resolution.

As this example shows, by using the octree decomposition, the total number of voxels that is analyzed is  $1+4+16+52+34\times16$  (step-wise), which is a total of 617. In comparison, the brute-force search in the 3D voxels space would have analyzed  $32\times32 = 1024$  voxels. The reconstruction results obtained using octrees is identical to the one that uses the brute-force search.

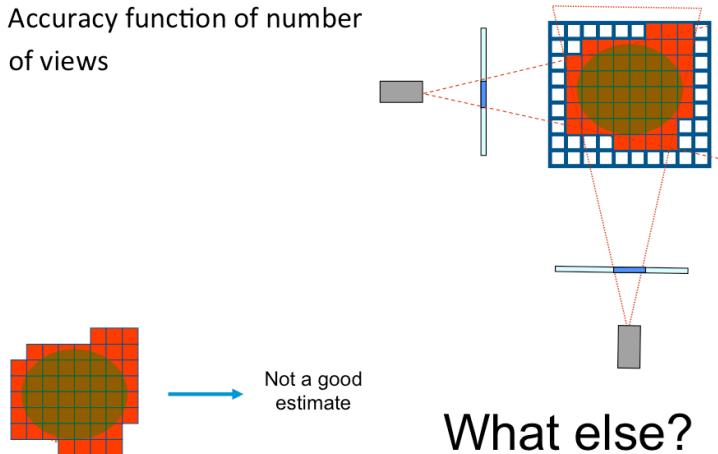
## Advantages of Space carving

- Robust and simple
- No need to solve for correspondences

The main positives of space carving are that it is a robust method that is also simple. There is no need to solve the correspondence problem (the problem of establishing correspondences between unknown points across multiple views).

## Limitations of Space carving

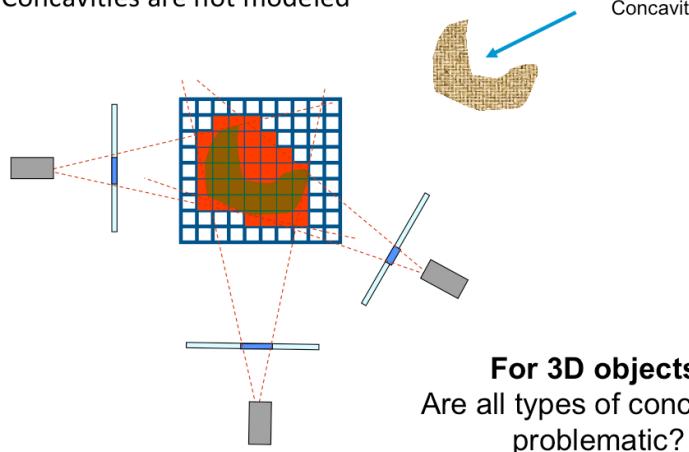
- Accuracy function of number of views



However, there are also a few disadvantages to using space carving. If the number of views is too low, then we end up with a very loose upper bound estimate for the visual hull of the object. So, in a sense, the accuracy is a function of the number of distinct views (so that more voxels are likely to be eliminated during the consistency checks).

## Limitations of Space carving

- Concavities are not modeled

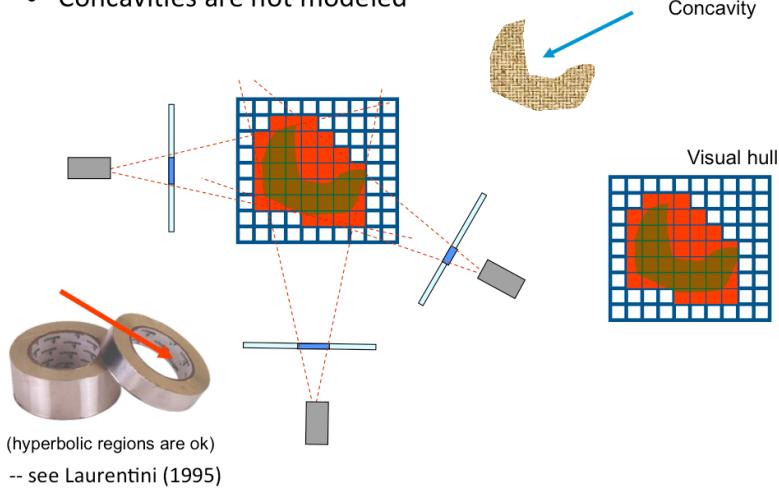


Another major drawback with using space carving is that it is incapable of modeling concavities in the object. Suppose the object is of the shape given in the slide – notice the large concavity on the right hand side. In this case, none of the consistency checks will eliminate any of the voxels in the concavity.

**Are all types of concavities  
problematic?**

## Limitations of Space carving

- Concavities are not modeled



Interestingly, in 3D, concavities associated to hyperbolic regions do not suffer from this drawback in that there is always a view point from which the voxels in the concavity may violate the consistency test.

An extensive study of all the properties and limitations of the space carving technique can be found in Laurentini (1995).

## Space carving: A Classic Setup



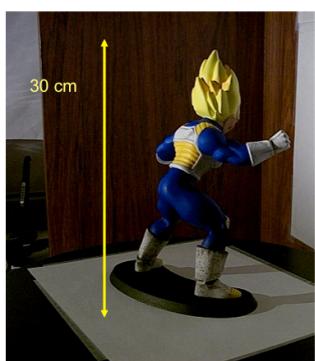
The figure depicts a standard setup that can be used to implement a space carving technique in practice. The object is placed on a turntable and observed from a camera fixed on a tripod. The turntable rotates  $X$  degrees about the vertical axis so as to simulate an equivalent motion of the camera, where  $X$  is a known (controllable) quantity. This way, we obtain multiple views of the object without moving the camera.

## Space carving: A Classic Setup

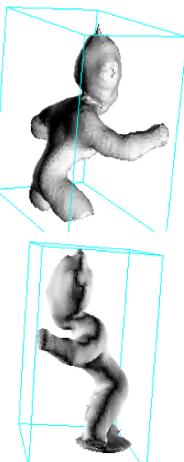


The camera and the object are shown separately here.

## Space carving: Experiments

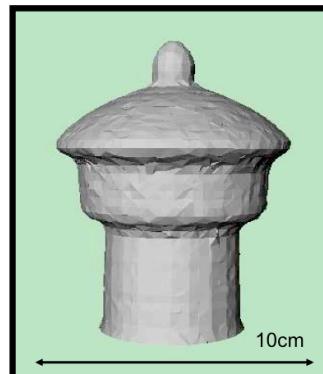


24 poses ( $15^\circ$ )  
voxel size = 1mm



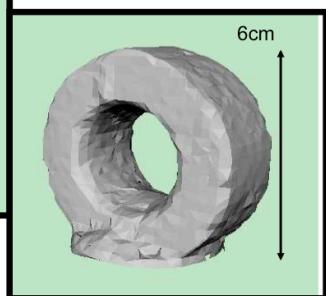
The slide shows examples of reconstruction on the right; here 24 view points (i.e., 24 rotation steps of 15 degrees each) were used and the voxel size was 1mm.

## Space carving: Experiments



24 poses ( $15^\circ$ )

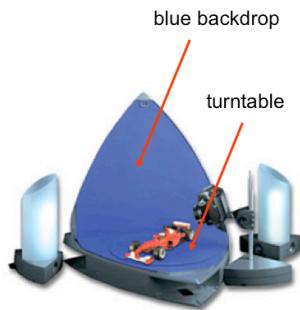
voxel size = 2mm



Other reconstruction results.

## Space carving: Conclusions

- Robust
- Produce conservative estimates
- Concavities can be a problem
- Low-end commercial 3D scanners



In summary, we have seen a few advantages (such as simplicity and robustness) and disadvantages of space carving (concavities, loose upper bound). We have also studied how an octree data structure can be employed to good effect, in reducing the number of operations needed to be performed. Because of these properties, space carving is often used in low-end commercial 3D scanners.

# Lecture 8

## Active stereo & Volumetric stereo



- Active stereo
  - Structured lighting
  - Depth sensing
- Volumetric stereo:
  - Space carving
  - Shadow carving
  - Voxel coloring

Silvio Savarese

Lecture 7 -

20-Apr-16

We now move to the shadow carving technique.

## Shape from Shadows

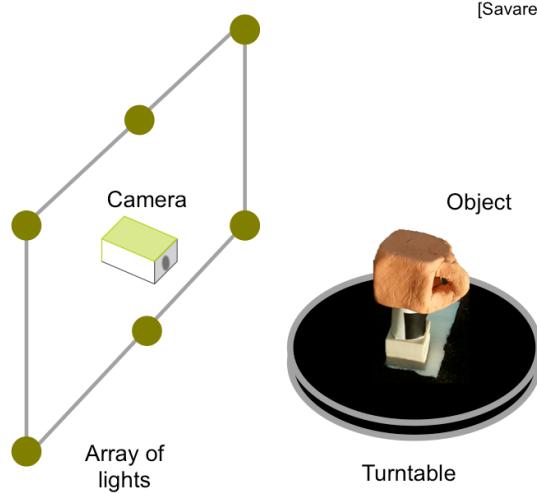
- Self-shadows are visual cues for shape recovery



Another important cue for determining the 3D shape of objects are self-shadows. The presence of such self-shadows indicates the existence of concavities (which cannot be inferred using the space carving approach).

## Shadow carving: The Setup

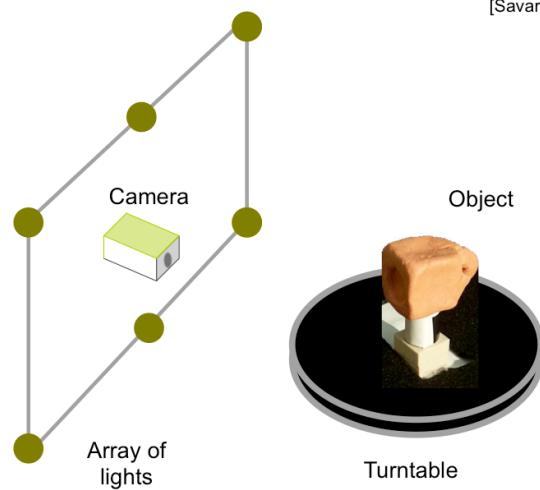
[Savarese et al. 2001]



Shadow carving uses a setup that is very similar to the one used for space carving. In this setup too, an object is placed on a turntable, so that the camera can be stationary and relative motion between the camera and the object can be achieved using the turntable.

## Shadow carving: The Setup

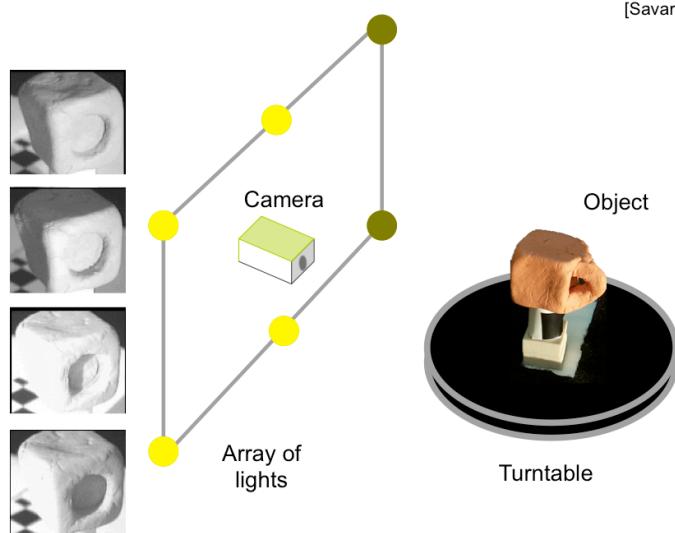
[Savarese et al '01]



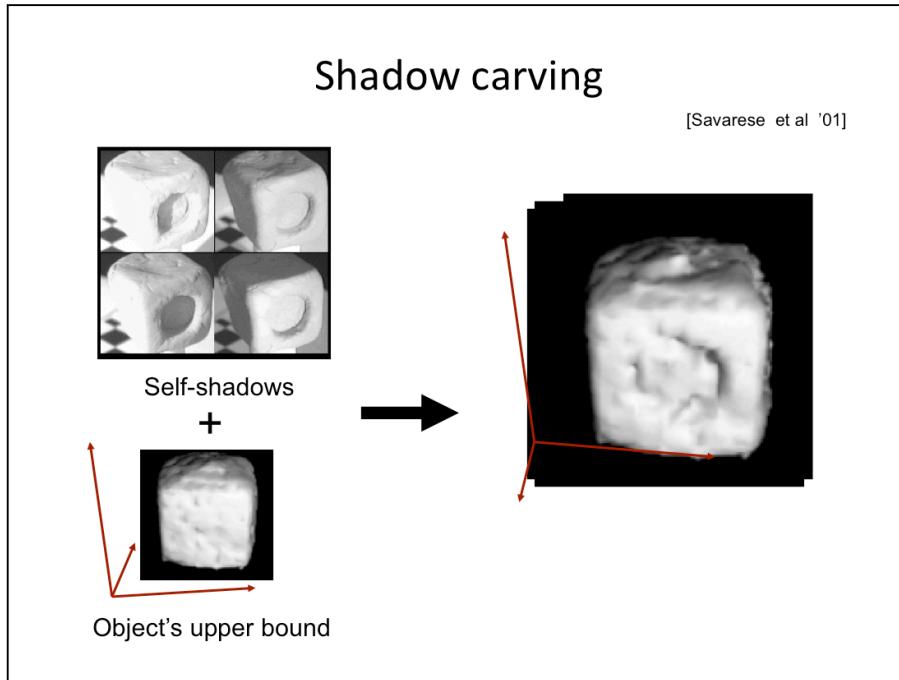
Moreover, there is an array of lights around the camera, whose states are appropriately controlled as on/off, that are used to make the object cast shadows.

## Shadow carving: The Setup

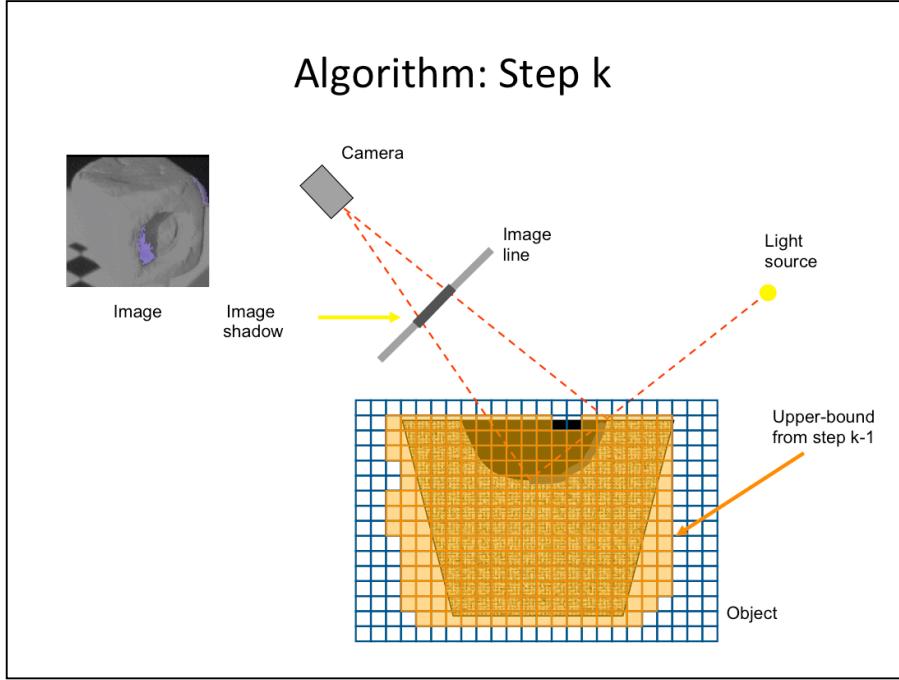
[Savarese et al '01]



Examples of images of the object illuminated by one of the light sources are shown on the left.

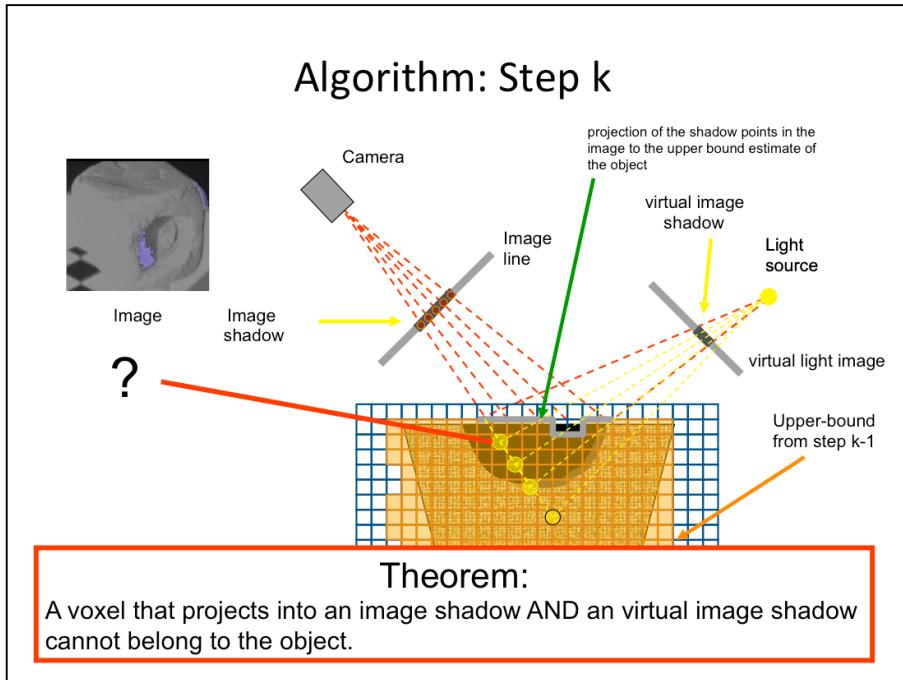


Shadow carving uses these images to refine the estimate of the object shape which is initially recovered using a space carving approach. As seen in this slide, self-shadows do reveal important cues that allow to infer the concavities on the object. Thus, any object with an arbitrary topology is suitable for applying this method.

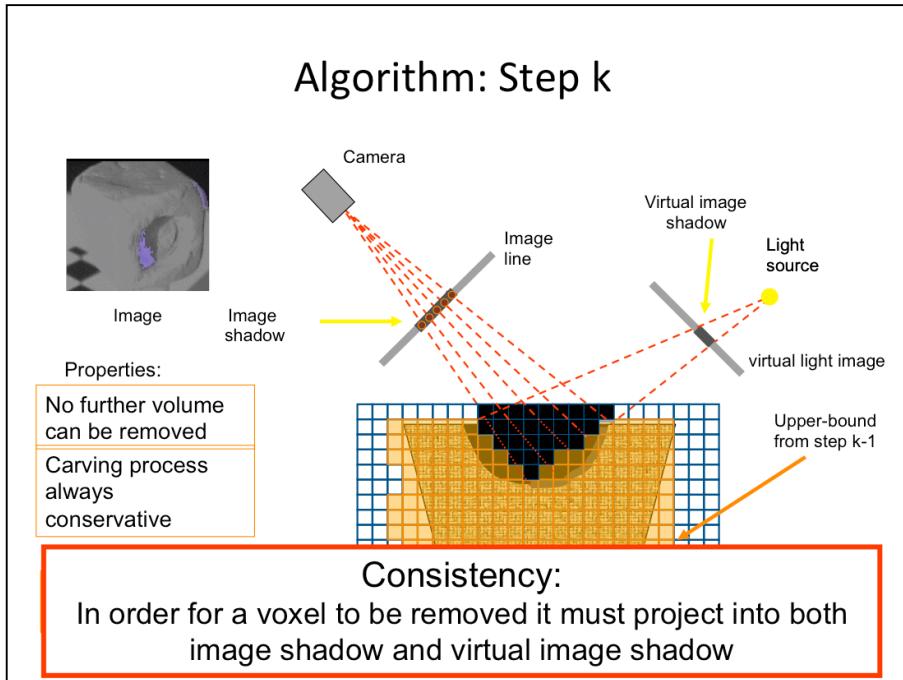


Here are a few more details about the shadow carving approach. A working volume encloses the object which do feature a concavity. The working cube is decomposed into voxels. We assume that in the step k-1, some of these voxels are marked as empty (white color) and others as full (orange color); the latter voxels form an upper bound estimate of the object shape.

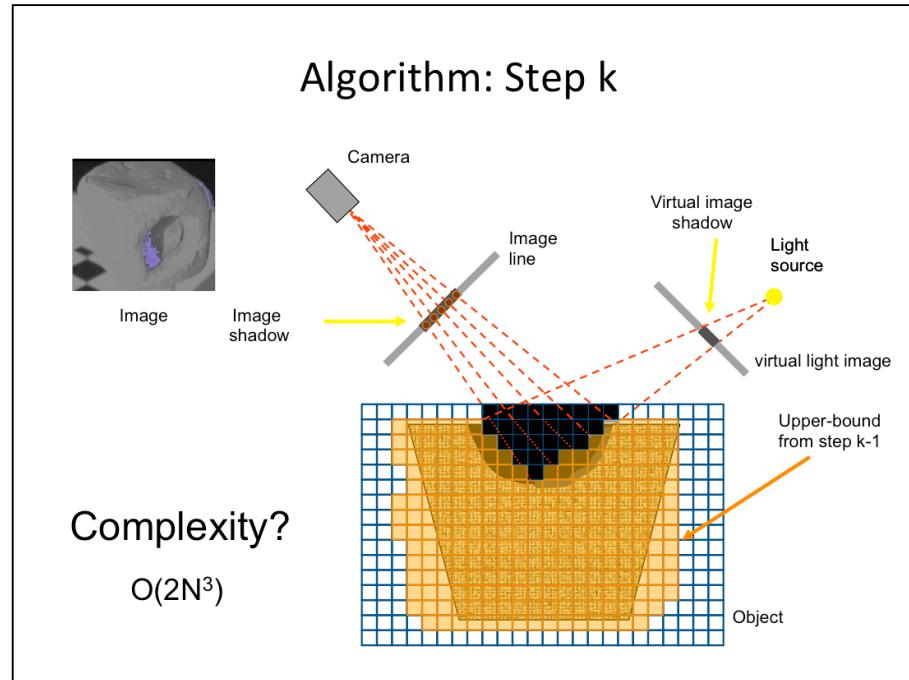
At step k, we do observe the object from either a different view point or illuminated from a different light source. This produces a new image with an associated shadow (dark gray line in the image line).



The projection of the shadow line in the image line (which is measurable) to the upper bound estimate of the object (which is known since it has been estimated in the previous step) yields the gray line which is marked by the green arrow. The projection of such gray region to the virtual light image (which is known, since the setup is calibrated) generates the “virtual image shadow” shown in dark gray. The virtual light image is a virtual plane that is used to model the light source similarly to what have seen for the pinhole camera model. The shadow carving algorithm leverages a theorem that claims that “A voxel that projects into an image shadow AND to a virtual image shadow cannot belong to the object”.



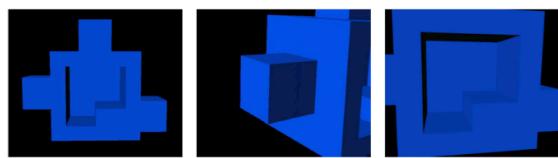
We use this as a consistency criterion: A voxel that does project into both an image shadow AND a virtual image shadow, is marked as occupied. A voxel that projects outside either the image shadow OR the virtual image shadow, is marked as empty. It can be shown that this method always produces a conservative volume estimate and that no further voxels can be marked empty given the current set of observations (i.e., the number of sub-voxels we carve out because of shadow carving is the largest possible given the current set of observations).



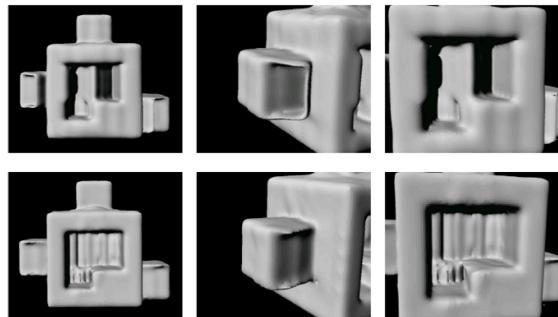
If we consider an  $N \times N \times N$  volume in the 3D space, it follows that the complexity of the shadow carving algorithm is  $O(2N^3)$  in that for every voxel (and there are  $N^3$  of them) we need to project to both camera and light (=two operations).

## Simulating the System

- 24 positions  
- 4 lights



- 72 positions  
- 8 lights



This slide illustrates the results obtained with different configurations of the number of lights and positions. The top row shows synthetic images of an artificial scene illuminated by different light sources.

Notice that certain portions in the concave regions cannot be recovered because of an insufficient number of views or light sources.

## Results

- 16 positions  
- 4 lights

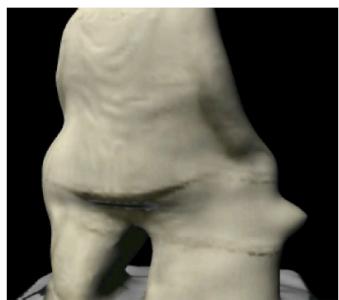


Space carving

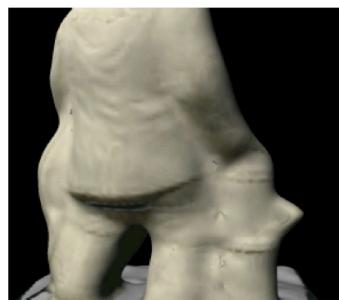
Shadow carving

More results using real world data. The top shows images of the object illuminated by different light sources. The central row indicates reconstruction results obtained using space carving. The bottom row indicates reconstruction results obtained using shadow carving.

## Results



Space carving



Shadow carving

Details of the reconstruction results. Notice that most of the concavities have been recovered by shadow carving.

## Shadow carving: Summary

- Produces a conservative volume estimate
- Accuracy depending on view point and light source number
- Limitations with reflective & low albedo regions

In summary, shadow carving always produces a conservative volume estimate. The quality of the results depends on the number of light sources and views. Some disadvantages of this approach are that it cannot handle cases where the object contains reflective and low albedo (i.e., dark) regions. This is because shadows cannot be detected accurately in such conditions.

# Lecture 8

## Active stereo & Volumetric stereo



- Active stereo
  - Structured lighting
  - Depth sensing
- Volumetric stereo:
  - Space carving
  - Shadow carving
  - Voxel coloring

Silvio Savarese

Lecture 7 -

20-Apr-16

We now move to the last technique – voxel coloring.

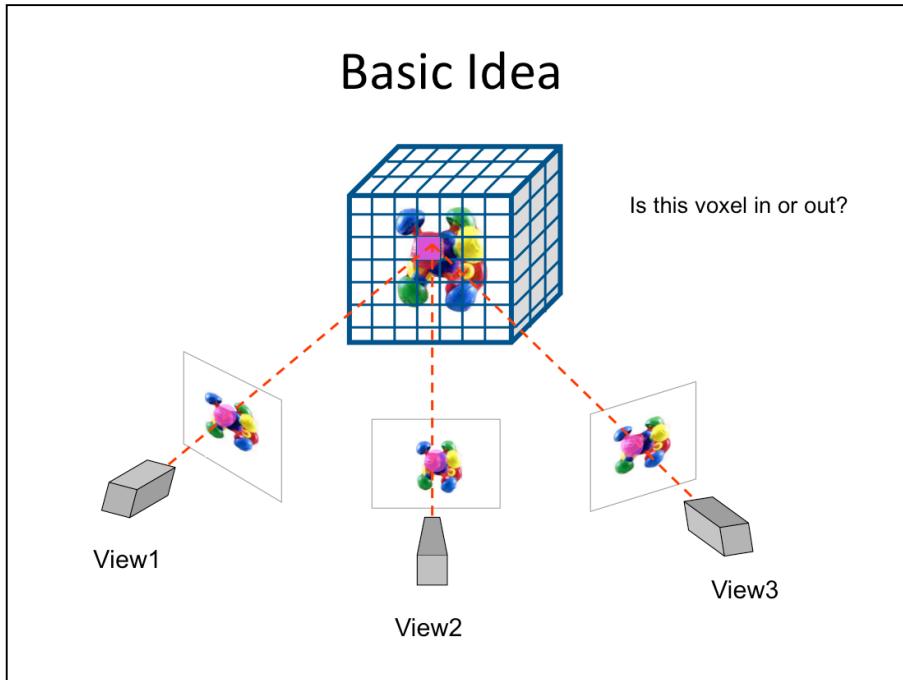
## Voxel Coloring

[Seitz & Dyer ('97)]  
[R. Collins (Space Sweep, '96)]



- Color/photo-consistency
- Jointly model structure and appearance

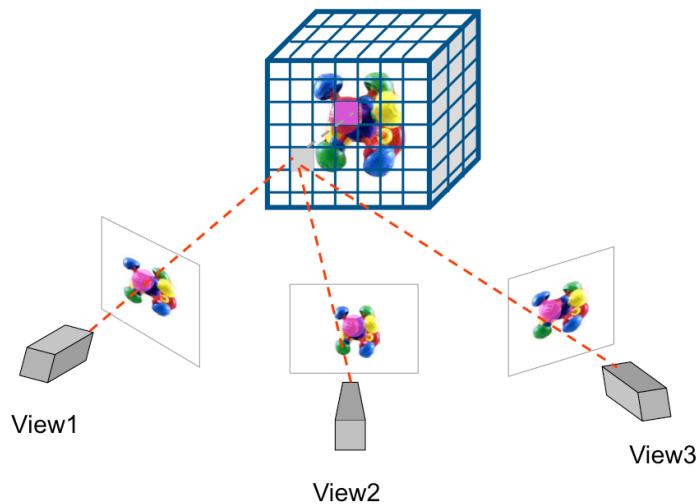
The preliminary idea of voxel coloring was introduced by Robert Collins in 1996 and by Seitz and Dyer in 1997. In this method, the color is used as a cue for carving out voxels.



Suppose that we are given images of the object we want reconstruct from several view points (three different views, in this example). For each voxel, we look at its corresponding projections in each of the images and decide whether or not the voxel belongs to the object (i.e. whether it is marked as full or empty). We do this by comparing the appearance (color) of each of these projections, verifying if they bear the same appearance. This is the consistency test for voxel coloring. The color associated to the projections of the voxel in the image can be used to transfer albedo information to the reconstructed voxel (this was not possible in space or shadow carving, where only a “full” or “empty” assignment is performed).

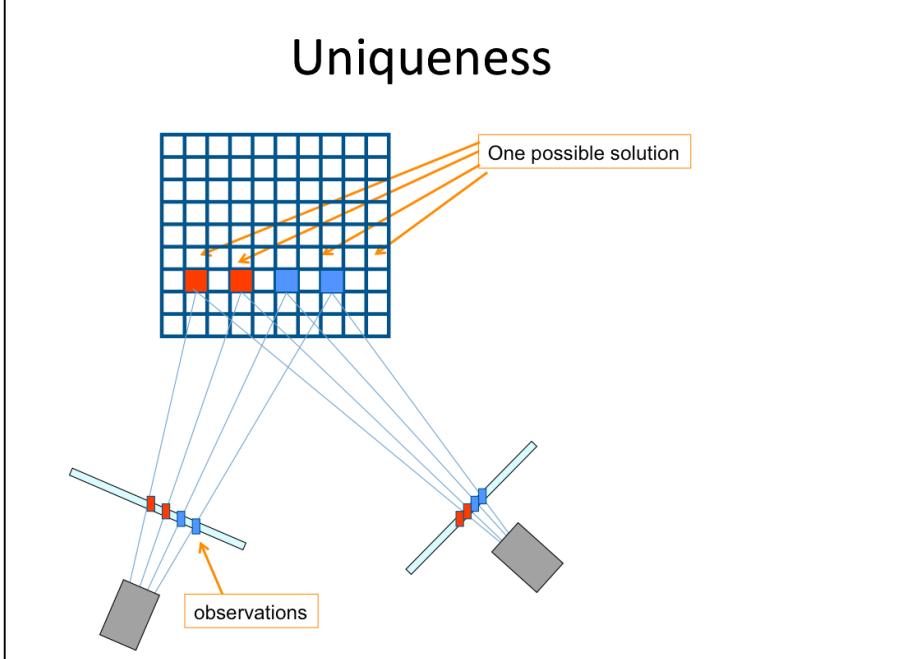
In this example, the voxel under consideration is projected to the images in a region whose color is consistent across images and therefore is label as full. The color “purple” can be transferred to the voxel so as to color it.

## Basic Idea



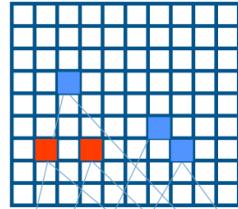
Here we see another examples. In this case the appearance of its projections in each of the views is not be consistent and the voxel is labeled as empty.

## Uniqueness



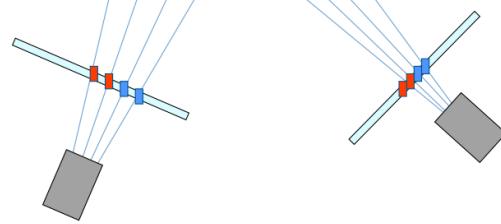
One drawback of voxel coloring is that it produces a solution that is not necessarily unique as the example here shows. Given the observations in the two image lines (red and blue pixels), the corresponding configuration of estimated voxels is not unique.

## Uniqueness



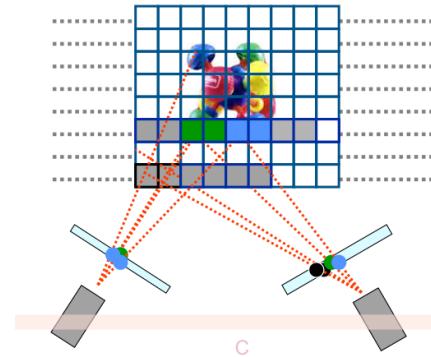
- Multiple consistent scenes

How to fix this?  
Need to use a visibility  
constraint



Another solution is presented here, which is still compatible with the same observations. This implies that in general multiple scenes in the 3D space are consistent with the same set of observations from the cameras. This complicates the problem of scene reconstruction using voxel coloring. It is possible to remove these ambiguities by introducing a visibility constraint in the reconstruction process – that is, by following the constraint that voxels are to be traversed in a particular order. The solution that is obtained by following the visibility constraint is unique and it is guaranteed to be a conservative (upper bound) estimate of the real object shape.

## The Algorithm



In details, the idea is explore voxels layer by layer, starting from those closer to the camera and then keep moving farther away from it. In this order, we perform the consistency check at each voxel and, accordingly, label voxels as full (occupied) or empty (unoccupied). The additional condition we impose (i.e., the visibility constraint) is that a voxel must be visible from at least two cameras in order to be labelled as full (if it also passes the consistency test).

## Algorithm Complexity

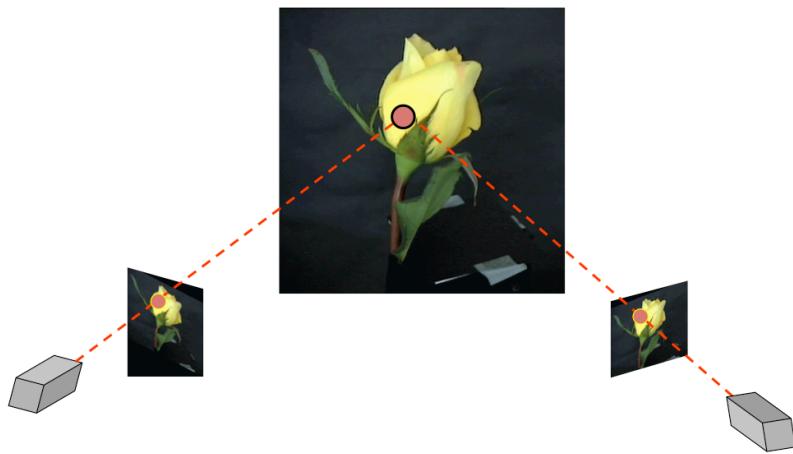
- Voxel coloring visits each  $N^3$  voxels only once
- Project each voxel into L images

$$\rightarrow O(L N^3)$$

NOTE: not function of the number of colors

If L is the number of different views from which the object is observed, then the complexity of the algorithm is  $O(LN^3)$ . It is not related in any way to the number of colors in the image.

## A Critical Assumption: Lambertian Surfaces



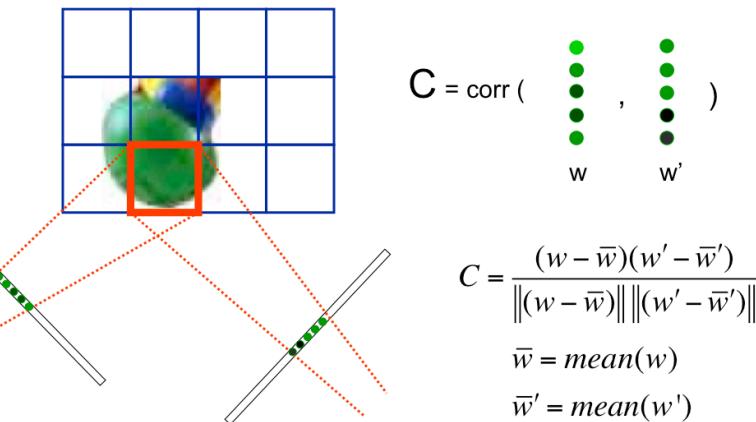
One critical assumption that needs to be made in the voxel coloring approach is that the object must be hypothesized to be lambertian – that is, the perceived luminance of a location in the object does not change with the viewpoint location and pose.

## Non Lambertian Surfaces



For non-lambertian surfaces, such as objects with reflective surfaces, the appearance of the same part of the object (e.g. the circle highlighted in red in the figure) differs when observed from different viewpoints. It is easy to realize that voxel coloring cannot be applied to these objects as the photoconsistency check will fail.

## Photoconsistency Test



If  $C > \lambda = \text{threshold} \rightarrow \text{voxel consistent}$

Under the assumption that surfaces are lambertian, let us now take a closer look at how the photo-consistency test is performed. Each voxel is projected into an image. Because a voxel has a certain physical size, it is likely that its projection encloses more than one pixel. As the figure shows, the projected voxel in the left image (indicated by the dashed red lines) encloses 5 pixels whose RGB values (colors) can be collected into a vector  $w$ . So does the projection of the voxel to the right image, yielding the vector  $w'$ . The consistency between these two vectors can be measured using the correlation or a normalized cross correlation function  $C$ . Other metrics can also be used. Ideally, if points are consistent,  $w=w'$  (and  $C=1$ ). In practice these two vectors will be different (see lecture on stereo systems) so a threshold should be introduced. If  $C$  is above a certain threshold  $\lambda$ , we regard them as being photo-consistent, otherwise we do not.

## Experimental Results



Dinosaur



72 k voxels colored  
7.6 M voxels tested  
7 min to compute on a 250MHz



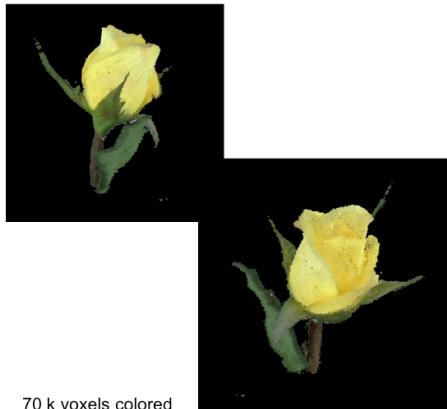
Image source: <http://www.cs.cmu.edu/~seitz/vcolor.html>

We now look at some of the experimental results obtained using Voxel coloring. The results also enlist the number of voxels colored, tested and the time taken for the reconstruction.

## Experimental Results



Flower



70 k voxels colored  
7.6 M voxels tested  
7 min to compute on a 250MHz

Image source: <http://www.cs.cmu.edu/~seitz/vcolor.html>

More results

## Experimental Results



More results

## Voxel Coloring: Conclusions

- Good things
  - Model intrinsic scene colors and texture
  - No assumptions on scene topology
- Limitations:
  - Constrained camera positions
  - Lambertian assumption

In summary, voxel coloring too has both advantages and disadvantages. Some benefits of this approach are that it works for any arbitrary scene topology and captures the intrinsic scene appearance (color and texture) in its model. Some of the drawbacks of this approach include:

- The object is assumed to be Lambertian;
- Some camera configurations are not allowed –this comes from the fact that we need to traverse the space with a certain visibility order.

## Further Contributions

- A Theory of Space carving [Kutulakos & Seitz '99]
  - Voxel coloring in more general framework
  - No restrictions on camera position
- Probabilistic Space carving

[Broadhurst & Cipolla, ICCV 2001]  
[Bhotika, Kutulakos et. al, ECCV 2002]

There have been extensions to these approaches that address some of the drawbacks that we discussed. Interested readers are referred to the relevant publications that describe these extensions.

Next lecture...

## Fitting and Matching

In the next lecture, we will learn how to fit a curve to a given set of observations, known as “Fitting”. We will also learn how to match parts of two images that correspond to the same part of an object.