A Framework for User-guided Multi-resolution 3D Reconstruction

Felicia Brisc

School of Electronic Engineering
Dublin City University

DCU

Vision Systems Group

Photorealistic 3D models from camera views

Build your own virtual character :

A versatile framework for 3D reconstruction

Low cost, easy to operate

→ Processed by non-tech users on average PCs

Structure from motion

→ Single moving standard digital camera

Self-calibration

→ Calibration directly from images

Flexibility

→ Varying parameters (zoom in/out)

Scalability

• 3D models are amenable to access through various environments/connection types implying trade-offs between level of detail, interaction, computational costs

• perceptual importance is determined ultimately by the human factor

→ perform a **multi-resolution** 3D reconstruction (varying resolution across the model)

→ enable users to control the complexity of different surface regions with **simple image editing operations**

3D Reconstruction pipeline

Input sequence/User input

Feature tracking/Labeling

Corresponding points

Camera self-calibration

Projective reconstruction

Projective 3D structure

Upgrade to metric structure

Metric 3D structure

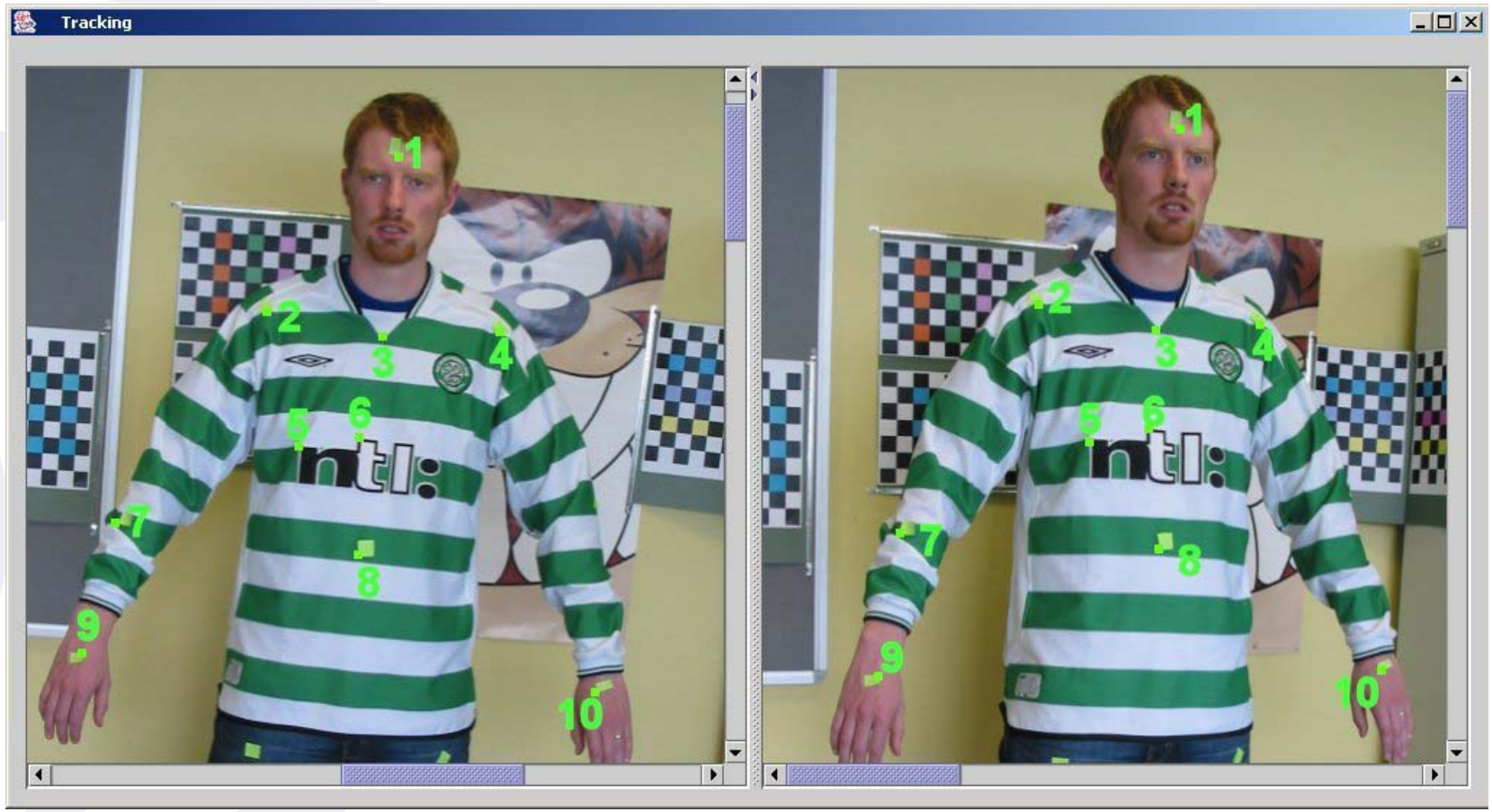
Scene reconstruction

3D model building/ voxel carving

Multi-resolution 3D voxel model

1. User Input

- **Tracking** : the user identifies several relevant points in a reference image and tracks them throughout the sequence



1. User Input

•**Region labeling** - the user selects image regions using common segmentation tools and assigns them label IDs corresponding to the chosen resolution

→ e.g. : the general reconstruction resolution is 25, while the face region is labeled for resolution 6.

2. Self-calibration

1. Projective reconstruction

→ **Projective factorization**

$$\lambda_{ij}\begin{bmatrix} x_{ij} \\ y_{ij} \\ 1 \end{bmatrix} = \mathbf{P}_i \mathbf{X}_j$$

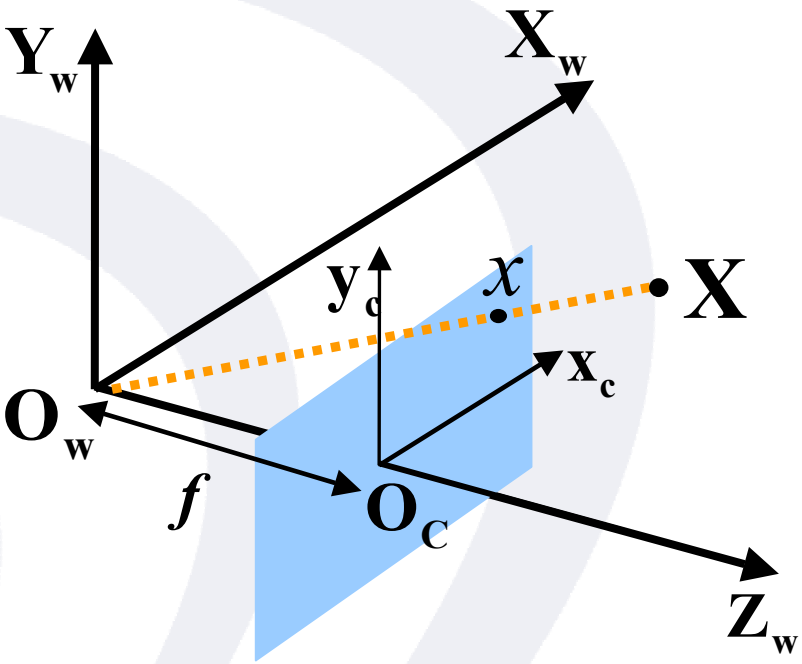


image plane

$\mathbf{P}_i$ - camera projective matrix

λ_{ij} - scale factor (projective depth)

$j = 1 \ldots m$ tracked points

$i = 1 \ldots n$ camera positions

$\underline{\mathbf{W}_s} = \mathbf{PX}$

rank 4

an iterative factorization is applied to make the equation consistent

2. Self-calibration

2. Upgrade to Euclidean (metric) structure

→ **recovery of the Absolute Quadric**

$$\omega^*_i = \mathbf{K}_i \mathbf{K}_i^T \approx \mathbf{P}_i \Omega^* \mathbf{P}_i^T$$

$$\Omega^* \rightarrow \mathbf{H}\, \Omega^*\, \mathbf{H}^T$$

-the **rectifying transformation H (4x4)** is computed by imposing constraints on the intrinsic camera parameters

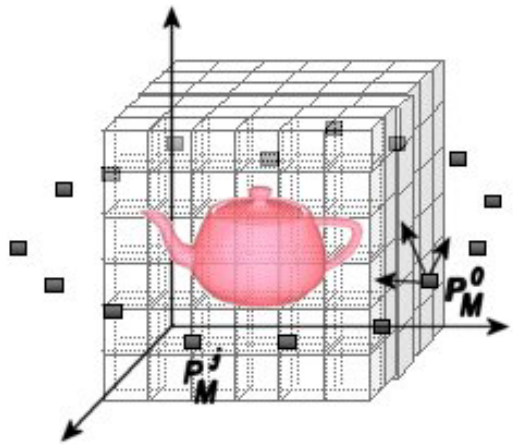
3. Multi-resolution 3D reconstruction

Generalized Voxel Coloring

- is a Space Carving technique

→ models the 3D scene by a reference frame and a volume of space in which the scene occurs

- The reconstruction is initialized with a 'bounding box' of voxels containing the 3D scene.



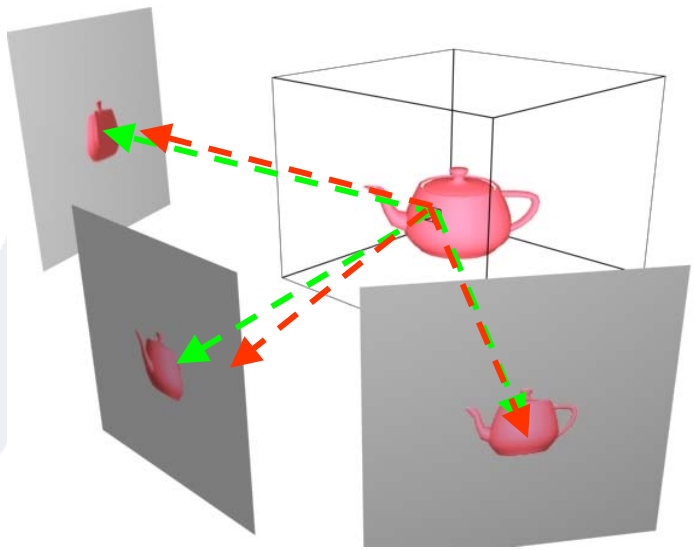
- The 3D shape of the scene is constructed by removing ("carving") voxels that are not ***photo-consistent*** with the reference views.

The photo-consistency criterion

Each voxel is projected to all camera views where it's visible

→ Voxel must not project to background in any of the reference views

→ Voxel has to be **color-consistent** – it has to project in each view to pixels of similar color, within a threshold



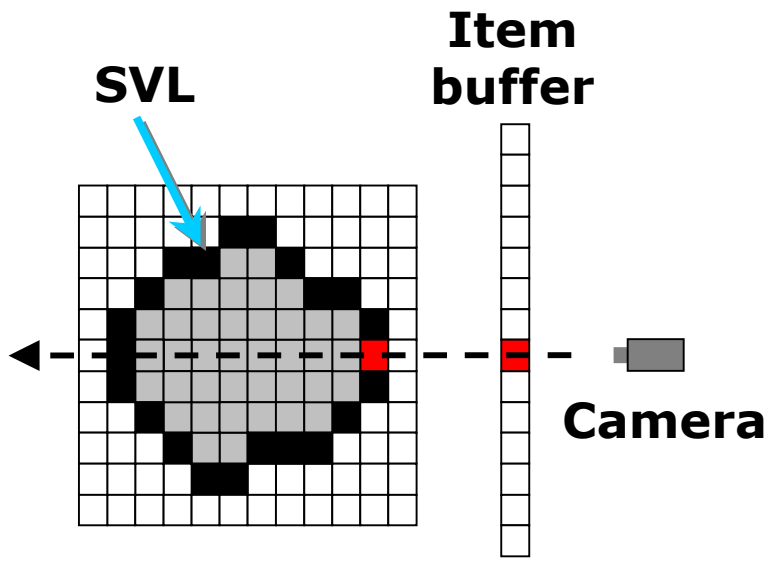
Voxel visibility handling

1. The Surface Voxel List (SVL)

2. The Item Buffer (IB)

SVL : The list of consistent voxels situated on the surface of the set of uncarved voxels.

- The SVL is initialized with the outside layer of voxels of the bounding box.
- Carved voxels are removed from the SVL, while adjacent uncarved voxels which become visible are added to the SVL.



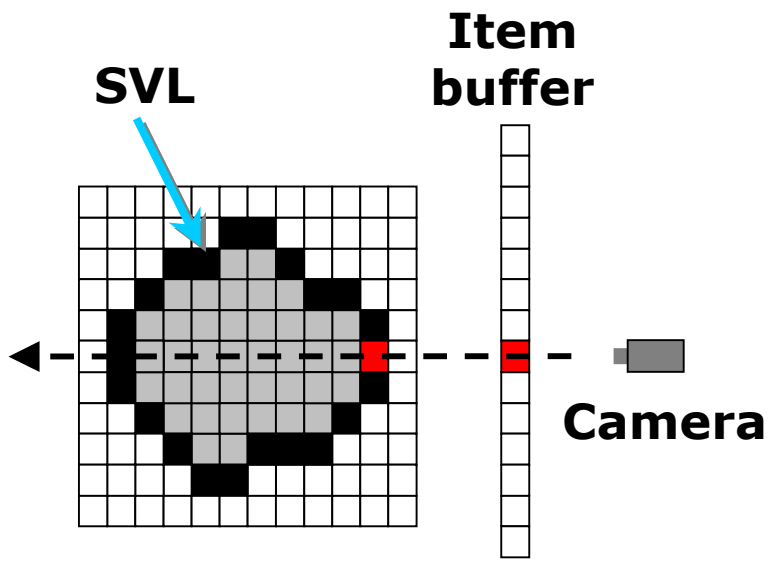
Voxel visibility handling

1. The Surface Voxel List (SVL)

2. The Item Buffer (IB)

IB : stores the ID of the closest visible voxel

- each voxel is assigned an unique ID
- the SVL is scanned sequentially in order to find all pixels a voxel V projects onto.
- if the distance from the camera to V is less than the distance stored for the pixel, the pixel's stored parameters are over-written with those of V



Voxel visibility handling

1. The Surface Voxel List (SVL)

2. The Item Buffer (IB)

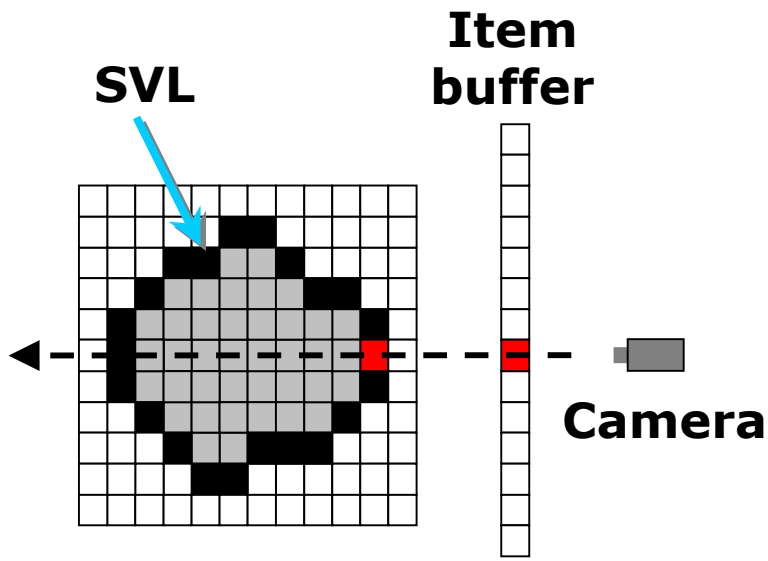
The SVL and the IB enable a bidirectional voxel-pixel mapping

Stored for each voxel on the SVL:

- color, pixel, label ID

•Stored for each pixel in the IB:

- closest visible voxel ID, label ID



Voxel carving

- initialize SVL

- compute item buffer for all images

- scan the SVL and compute for each voxel :
  - the visibility set
  - color and label statistics
  - perform photo-consistency check
  - carve inconsistent voxels / add adjacent uncarved voxels to the SVL

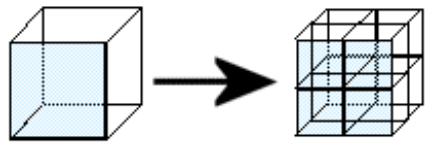
until no further voxels can be carved

Multi-resolution reconstruction

•Voxel carving is performed at the initial (coarse) resolution in order to isolate the labeled voxels

• A spatial constraint grid is applied to restrict further refining to labeled voxels

• Resolution is increased by tesselating labeled voxels into 8 subvoxels; voxel carving is performed subsequently on the refined voxels



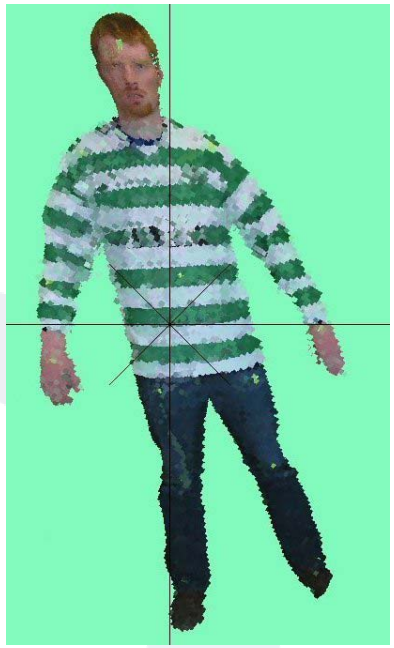
•The above steps are repeated iteratively until the required resolution is obtained

Results

- 63 tracked points
- 1 sequence of 5 images

- general resolution :25
- refined resolution :6 (face region)

Results



Left :The reconstructed human model at uniform resolution r=25

Right : Same model with the face region refined at resolution r=6

Left and right : Detail views of the previous uniform/multi-resolution 3D models