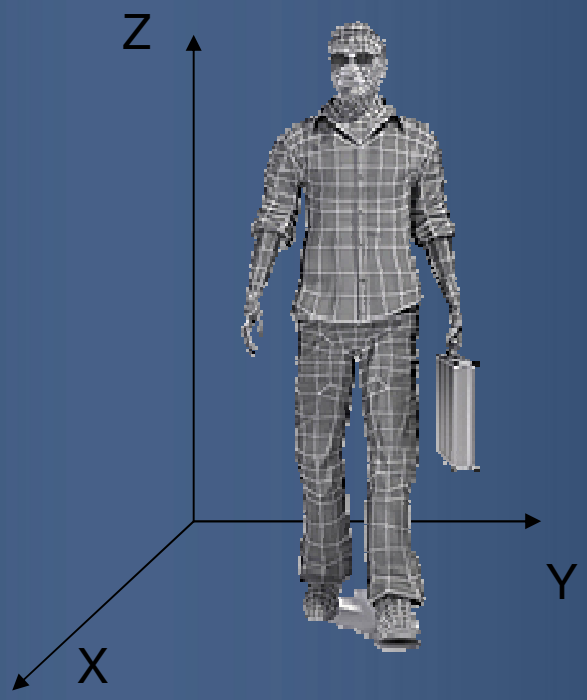
SFS

(Shape from shiloutte)



Jordi Cenzano
December - 2008

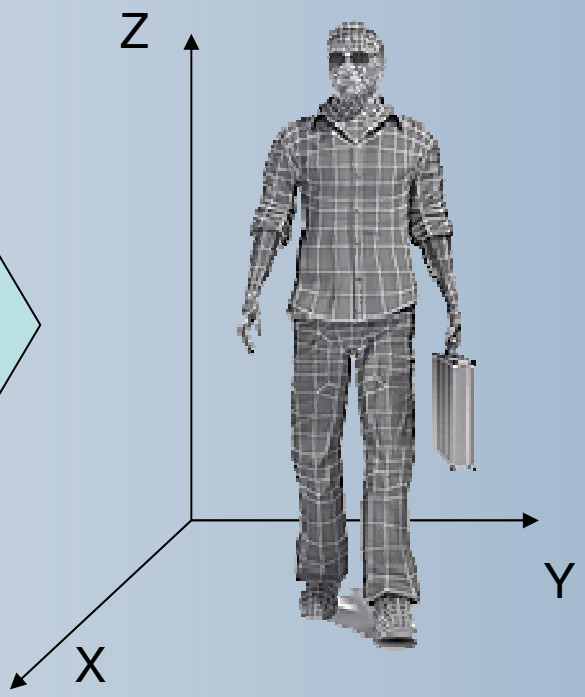
Index

- SFS (Shape From Shiloutte) – Visual Hull.
  - Introduction
  - From 2D to 3D
    - Thin lens
    - Calibrate
      - Intrinsics (K)
      - Extrinsics (P_{cw})
      - Projection matrix (P_{rj})
- SFS
- New: Object labeling
- New: PSFS (Partitioning SFS)

SFS - Visual Hull

Introduction (1/3)

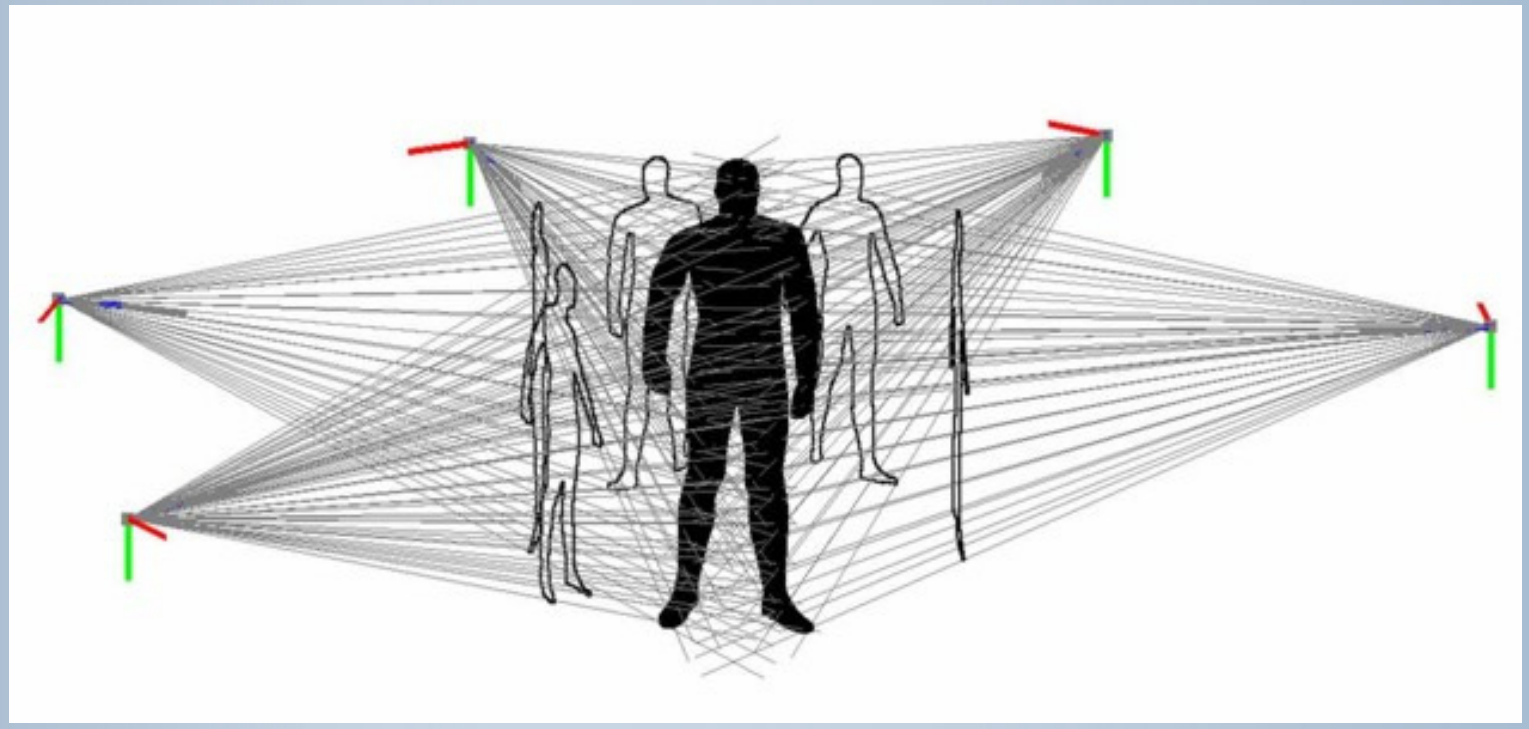
N x 2D Images



1 x 3D Mesh & Textures

SFS - Visual Hull

Introduction (2/3): Silhoutte and visual hull



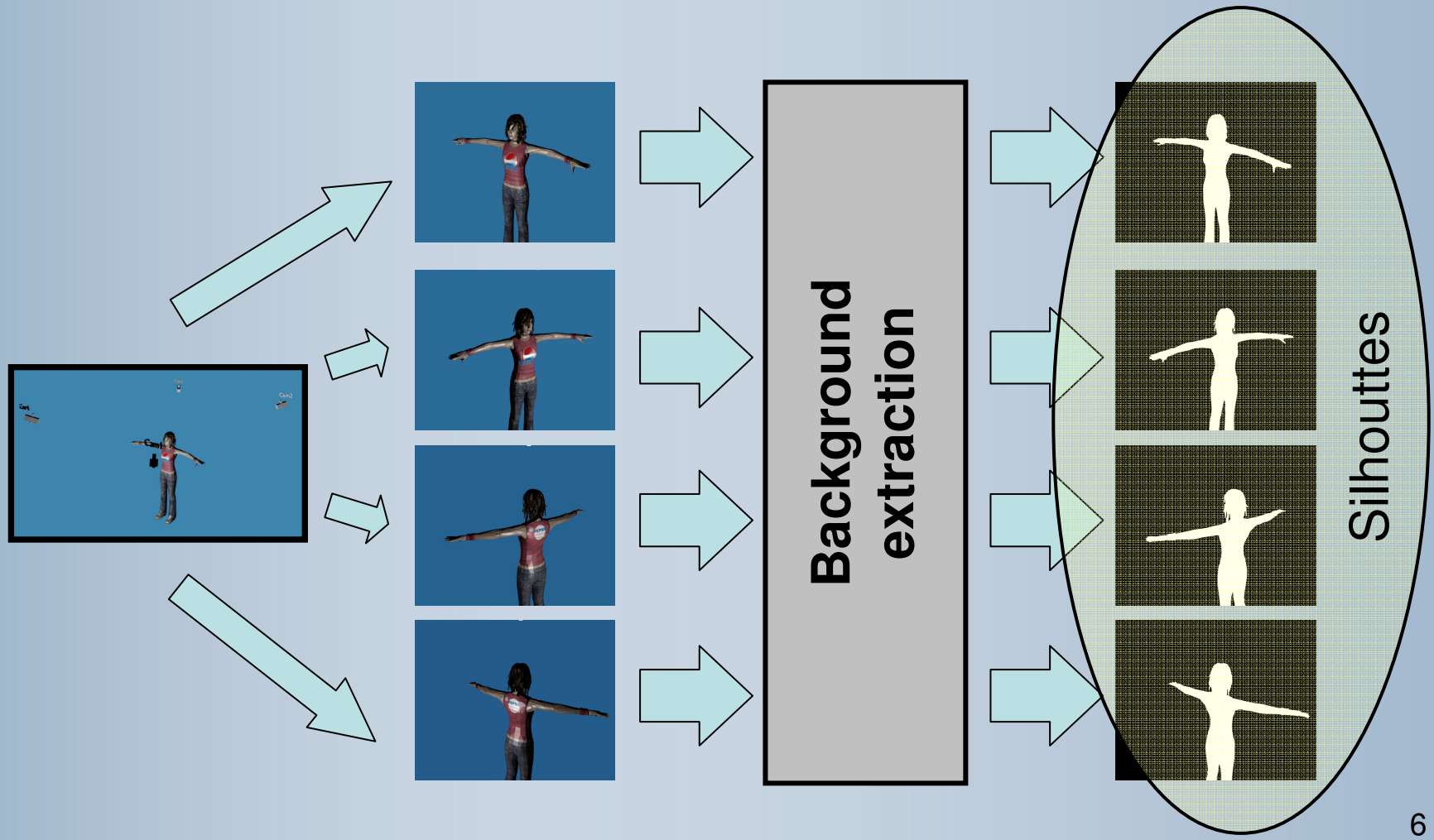
http://www.stanford.edu/group/biomotion/CurrentProjectsImages/VisualHull.jpg

SFS - Visual Hull

Introduction (3/3): Background extraction

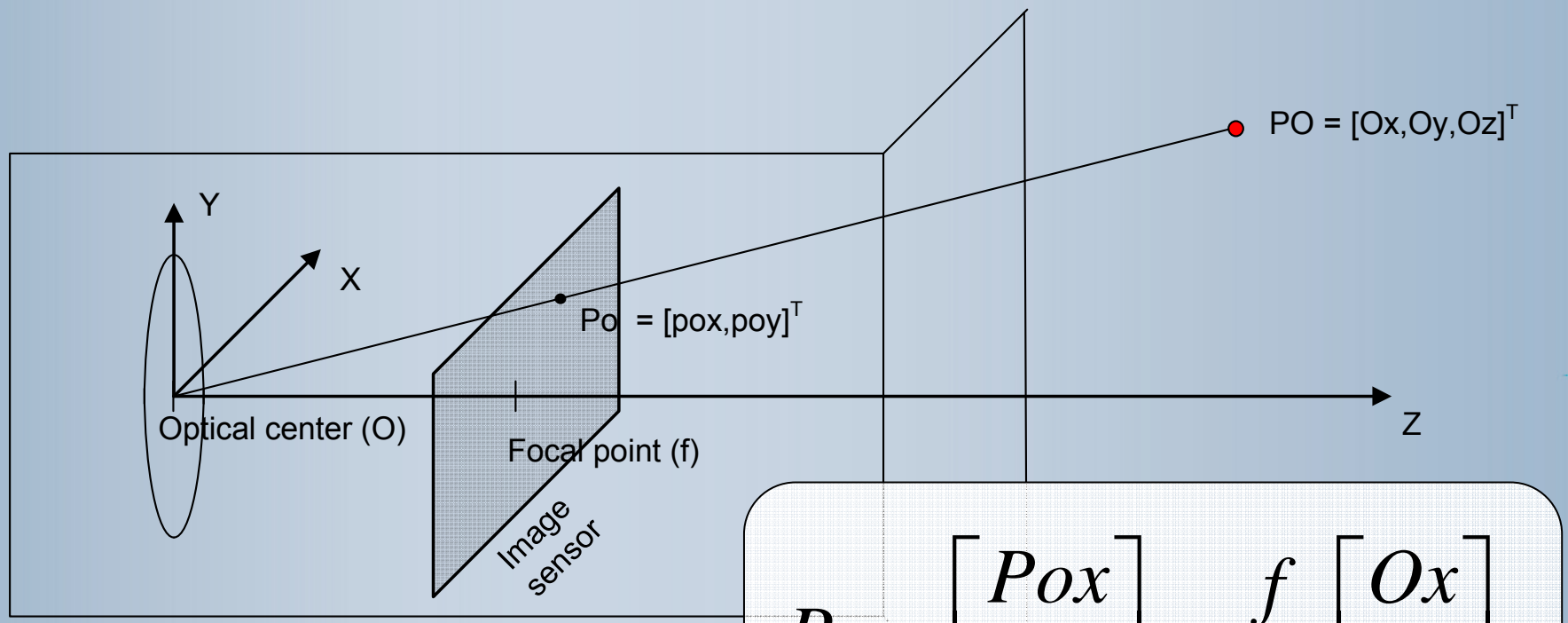
SFS - Visual Hull

Introduction (3/3): Background extraction



SFS - Visual Hull

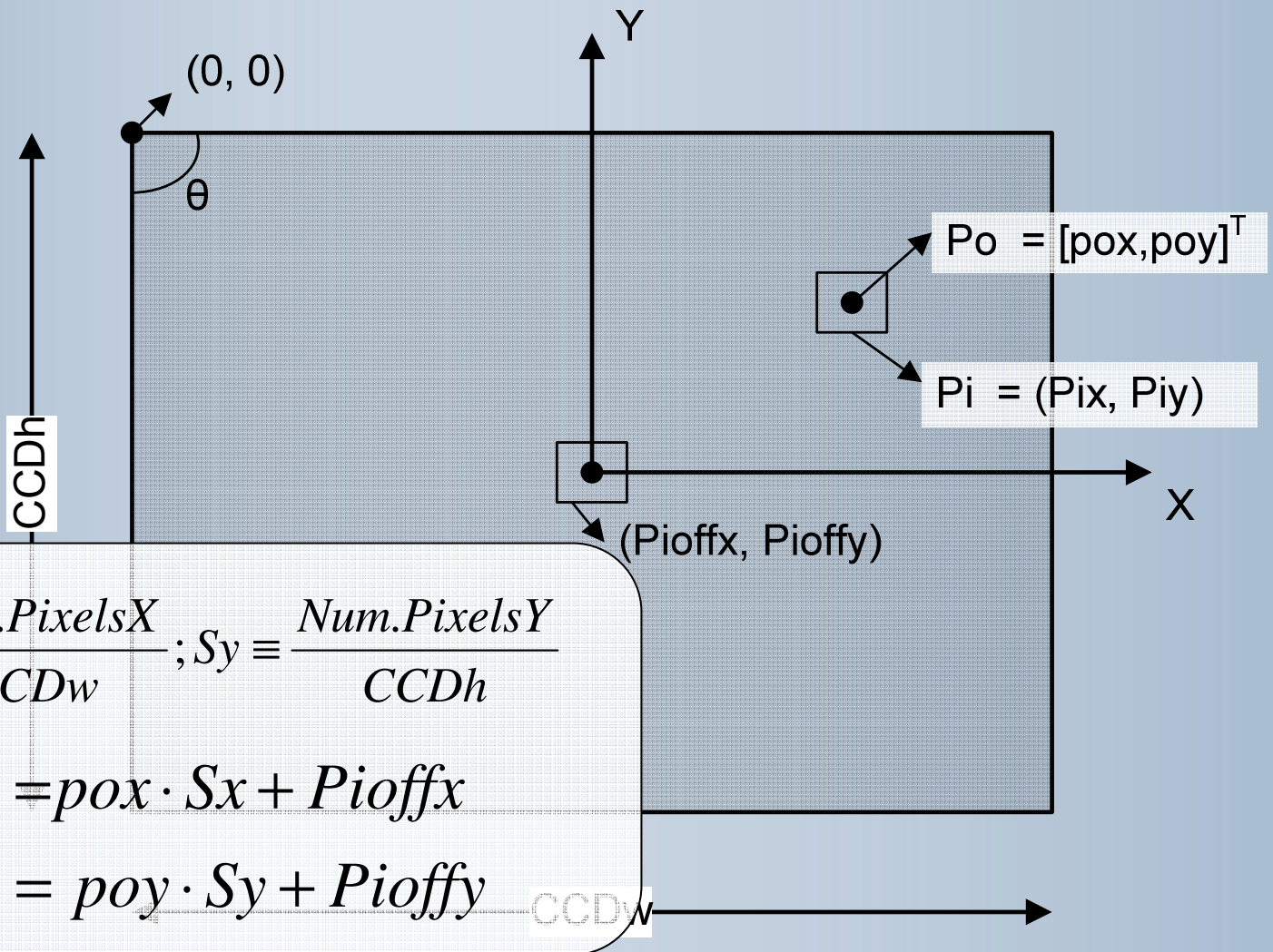
From 2D to 3D (1/6): Thin lens



$$Po = \begin{bmatrix} Pox \\ Poy \end{bmatrix} = \frac{f}{Oz} \begin{bmatrix} Ox \\ Oy \end{bmatrix}$$

SFS - Visual Hull

From 2D to 3D (2/6): Pixel



$$Sx \equiv \frac{Num.PixelsX}{CCDw}; Sy \equiv \frac{Num.PixelsY}{CCDh}$$

$$Pix = pox \cdot Sx + Pioffx$$

$$Piy = poy \cdot Sy + Pioffy$$

SFS - Visual Hull

From 2D to 3D (3/6): Homogeneus coords.

$$Poh = \begin{bmatrix} Pox \\ Poy \\ 1 \end{bmatrix} = \frac{1}{Oz} \cdot \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} Ox \\ Oy \\ 1 \end{bmatrix}$$

Kf

Πo

$$Pi = \begin{bmatrix} Pix \\ Piy \\ 1 \end{bmatrix} = \begin{bmatrix} Sx & S_\theta & Pioffx \\ 0 & Sy & Pioffy \\ 0 & 0 & 1 \end{bmatrix} \cdot \frac{1}{Oz} \cdot \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} Ox \\ Oy \\ 1 \end{bmatrix}$$

Ks

S_θ = 1/tan (θ) (Skew factor: pixel align compensation)

SFS - Visual Hull

From 2D to 3D (4/6): Intrinsics.

$$POh = \begin{bmatrix} Ox \\ Oy \\ 1 \end{bmatrix} \qquad \lambda = Oz$$

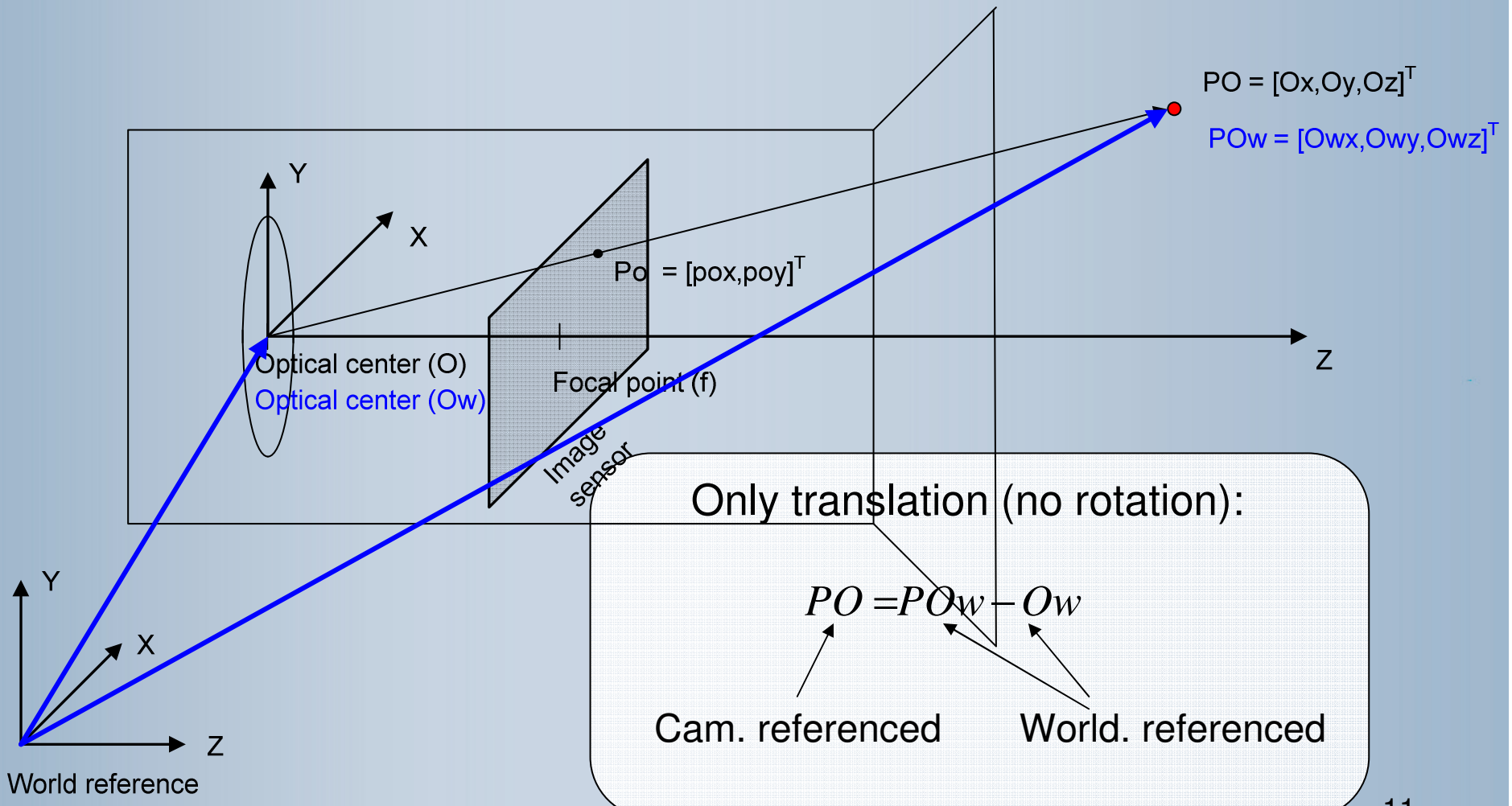
$$\lambda \cdot Pi = Ks \cdot Kf \cdot \Pi o \cdot POh$$

$$K \equiv Ks \cdot Kf \quad \text{(Instrinsics matrix)}$$

$$\lambda \cdot Pi = K \cdot \Pi o \cdot POh$$

SFS - Visual Hull

From 2D to 3D (5/6): Extrinsics.



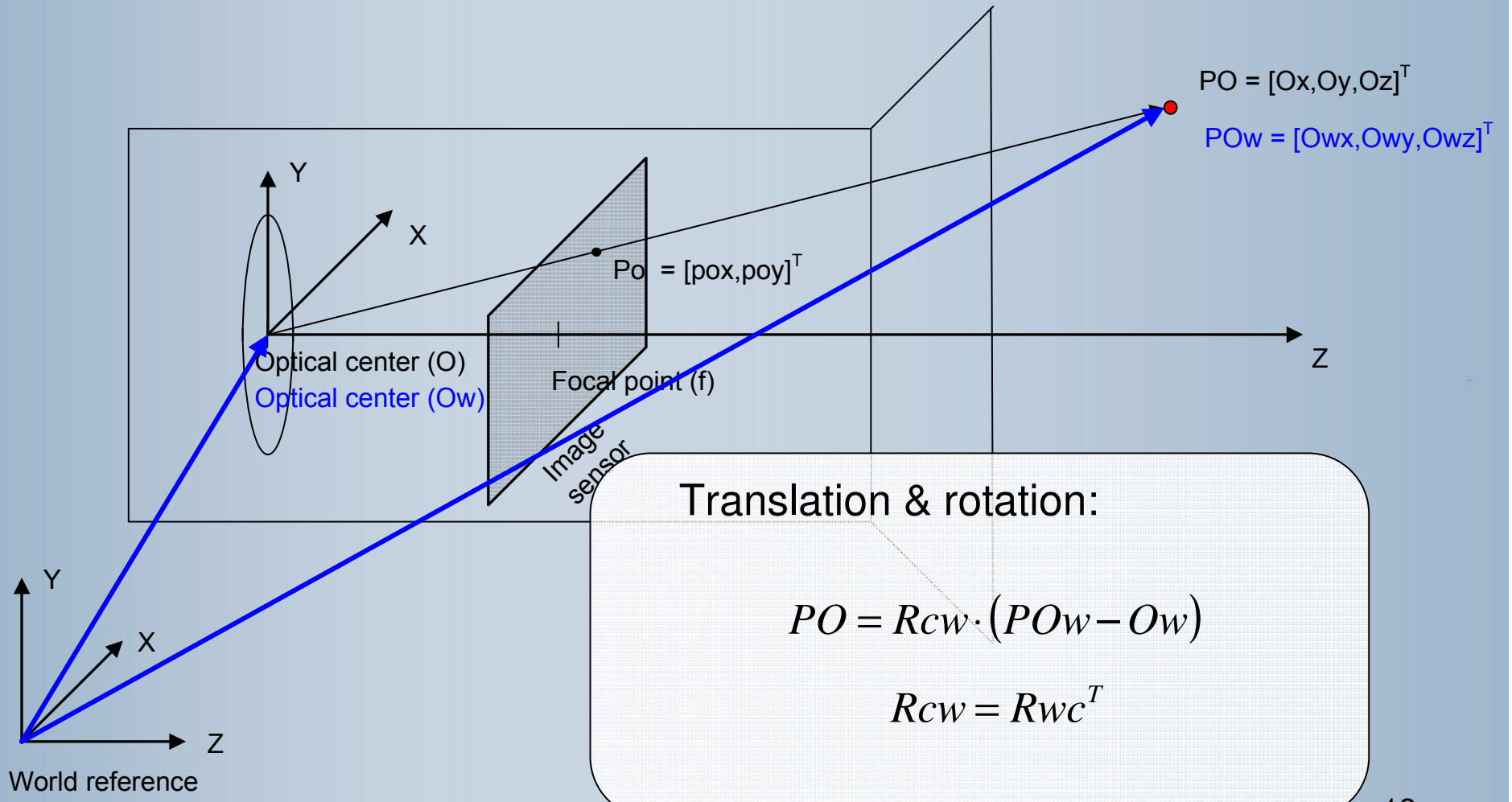
Only translation (no rotation):

$$PO = POw - Ow$$

Cam. referenced World. referenced

SFS - Visual Hull

From 2D to 3D (5/6): Extrinsics.



Translation & rotation:

$$PO = Rcw \cdot (POw - Ow)$$

$$Rcw = Rwc^{T}$$

SFS - Visual Hull

From 2D to 3D (6/6): Extrinsics.

$$\begin{bmatrix} PO \\ 1 \end{bmatrix} = \begin{bmatrix} Rwc^T & -Rwc^T \cdot Ow \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} POw \\ 1 \end{bmatrix}$$

$$Pcw \equiv \begin{bmatrix} Rwc^T & -Rwc^T \cdot Ow \\ 0 & 1 \end{bmatrix}$$ (Extrinsics matrix)

$$\lambda \cdot Pi = K \cdot \Pi o \cdot P_{cw} \cdot \begin{bmatrix} POw \\ 1 \end{bmatrix}$$

SFS - Visual Hull

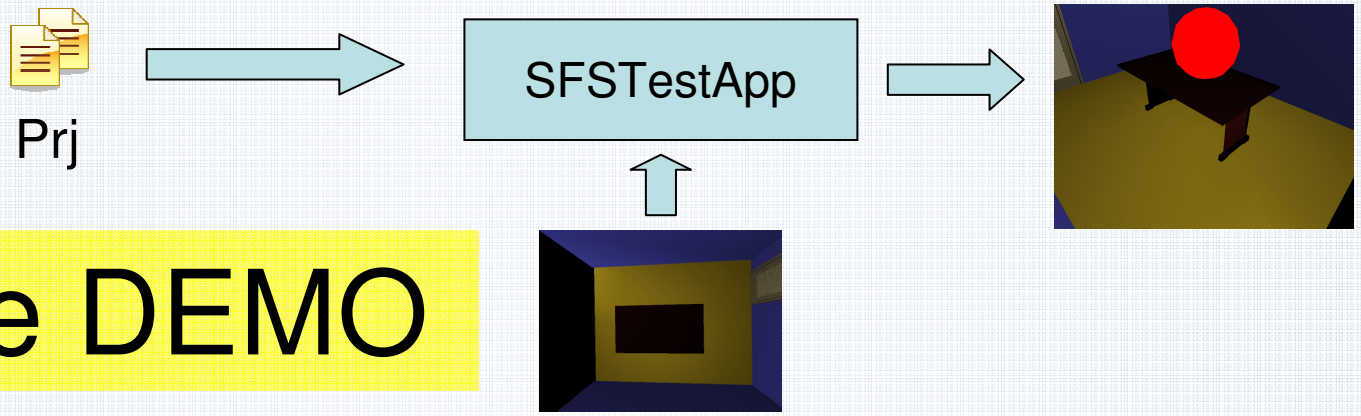
From 2D to 3D (1/1): Projection matrix.

$$P_{rj} \equiv K \cdot \Pi_o \cdot P_{cw}$$

$$\lambda \cdot Pi = P_{rj} \cdot \begin{bmatrix} POw \\ 1 \end{bmatrix}$$

SFS - Visual Hull

SFS Demo SFSTestApp (1/1):



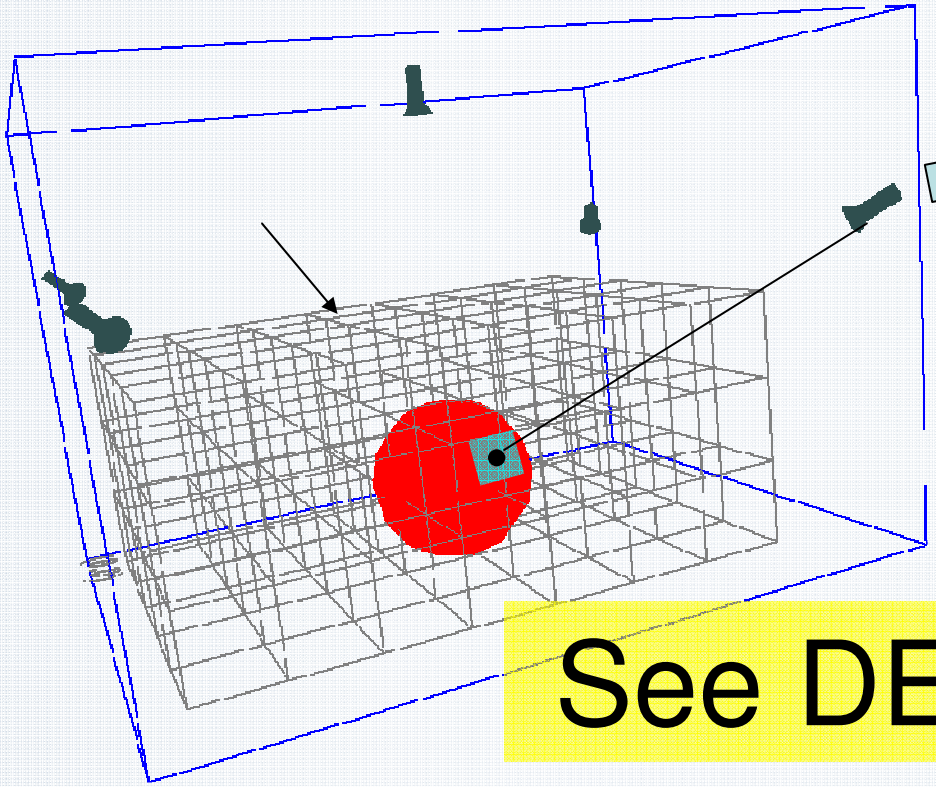
See DEMO

- It loads Prj matrix of cam.
- It loads a 3D composition of empty room (VMRL97)
- It can put 3D objects in the room
- It can save images from any viewpoint (PNG)
- **Can be used to test SFS algorithm and to evaluate them**

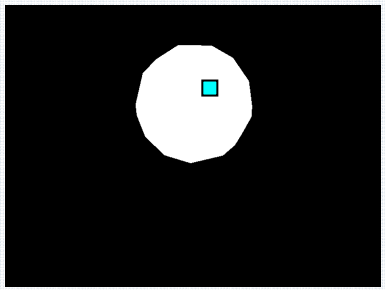
SFS - Visual Hull

SFS Demo Matlab toolbox (1/2):

SFS Implementation:



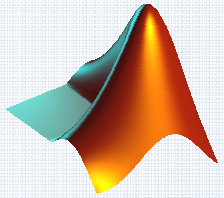
Cam1 silhouette



If (mask(pox,poy) > 0)&&(…

voxel(x,y,z) = obj

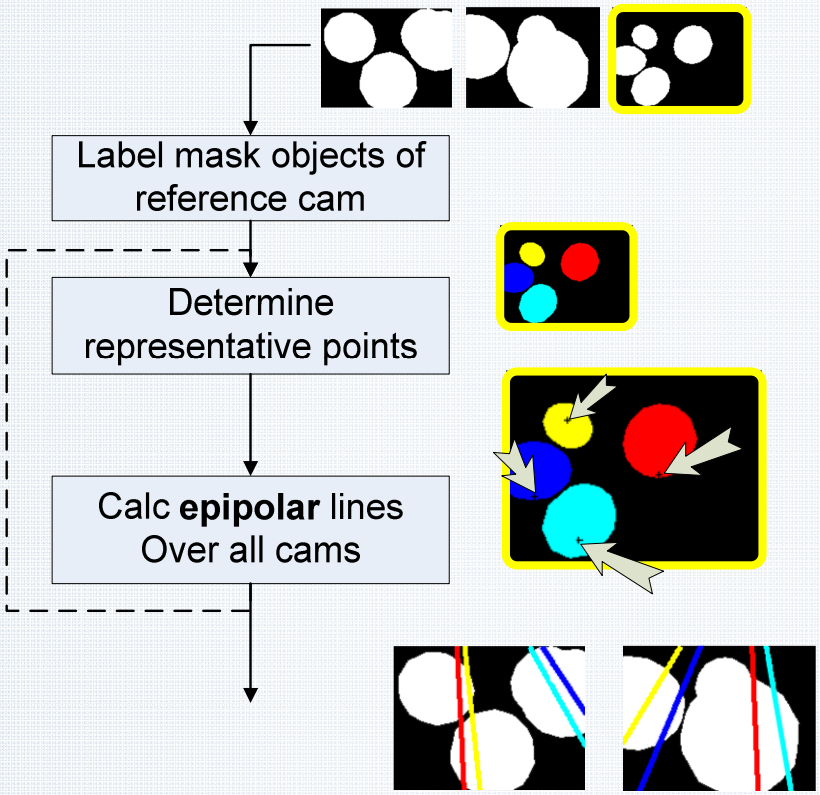
See DEMO



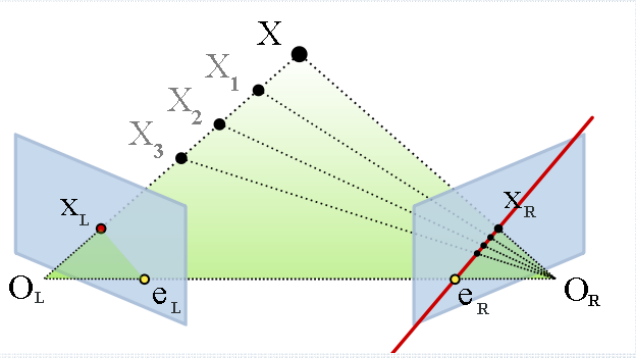
SFS - Visual Hull

SFS Demo Matlab toolbox (2/2):

Object labeling Implementation:



Epipolar geometry

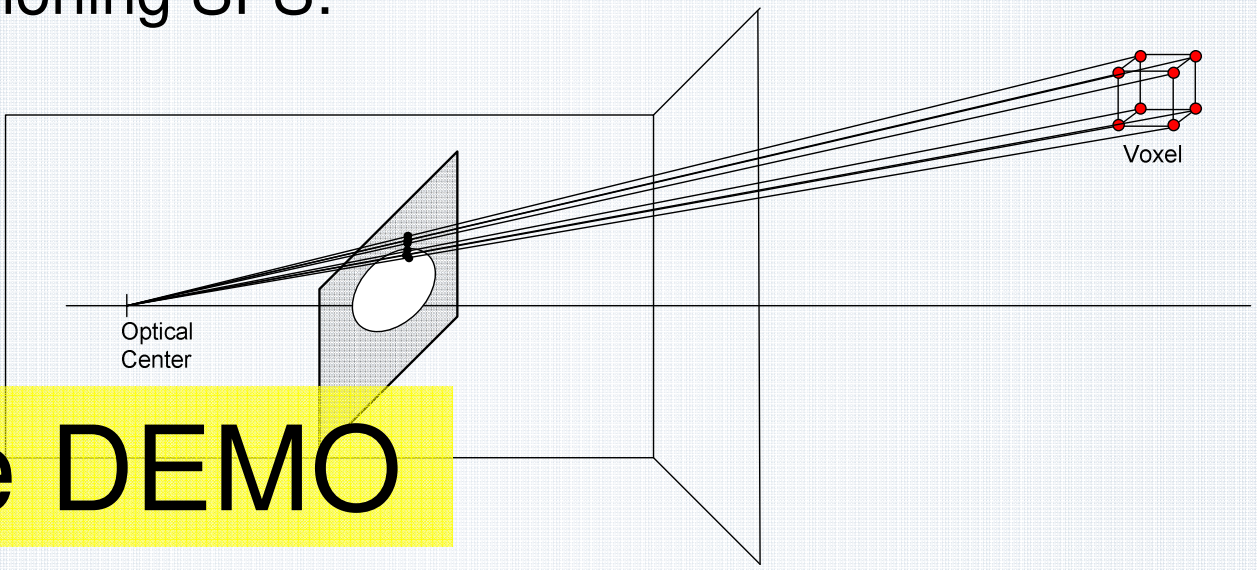


http://en.wikipedia.org/wiki/File:Epipolar_geometry.svg

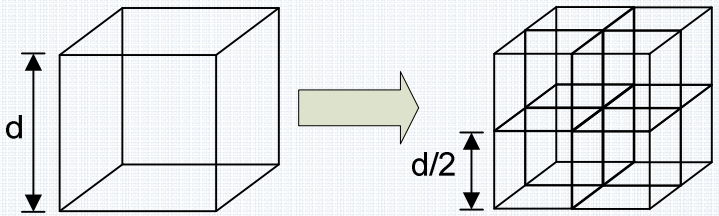
SFS - Visual Hull

SFS Demo C++ (1/1):

Partitioning SFS:



See DEMO



C++