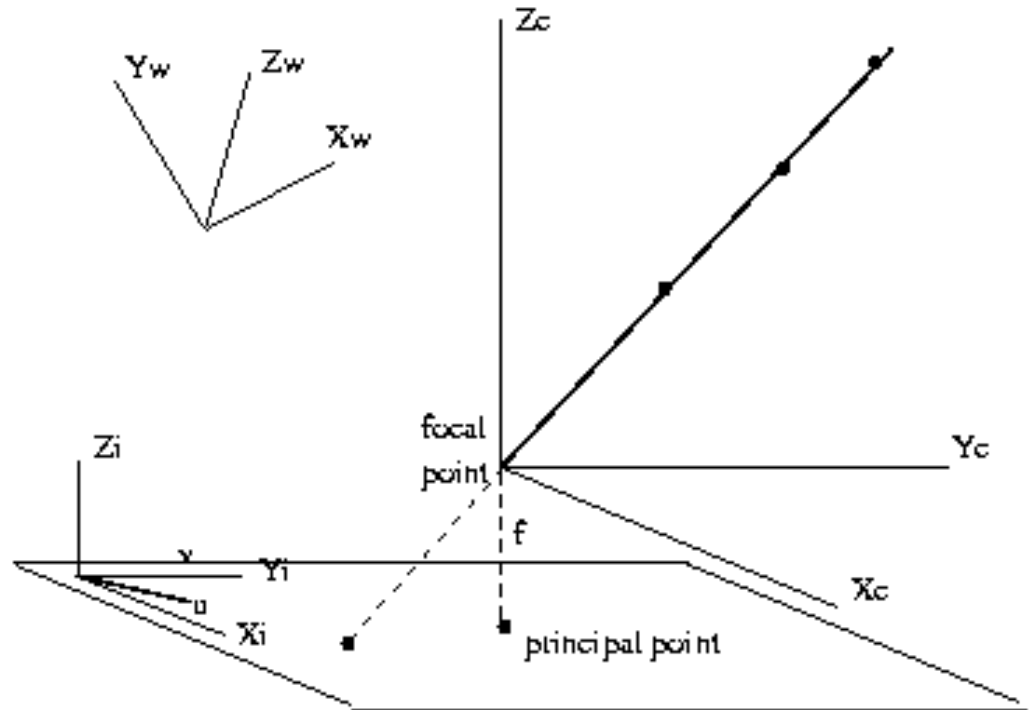


# Augmented Reality

# Camera Model and Projection

- $(X_w, Y_w, Z_w)$ : World coordinates
- $(X_c, Y_c, Z_c)$ : Camera coordinates
- $(X_i, Y_i, Z_i)$ : Image coordinates
- $(u, v, w)$ : Image affine coordinates



Camera Model is a mapping  $C: \mathbb{R}^3 \rightarrow \mathbb{R}^2$

# Develop Camera Model in three steps

Step 1.  $(x_w, y_w, z_w) \rightarrow (x_c, y_c, z_c)$  world to camera coords

This coordinate transformation is given by

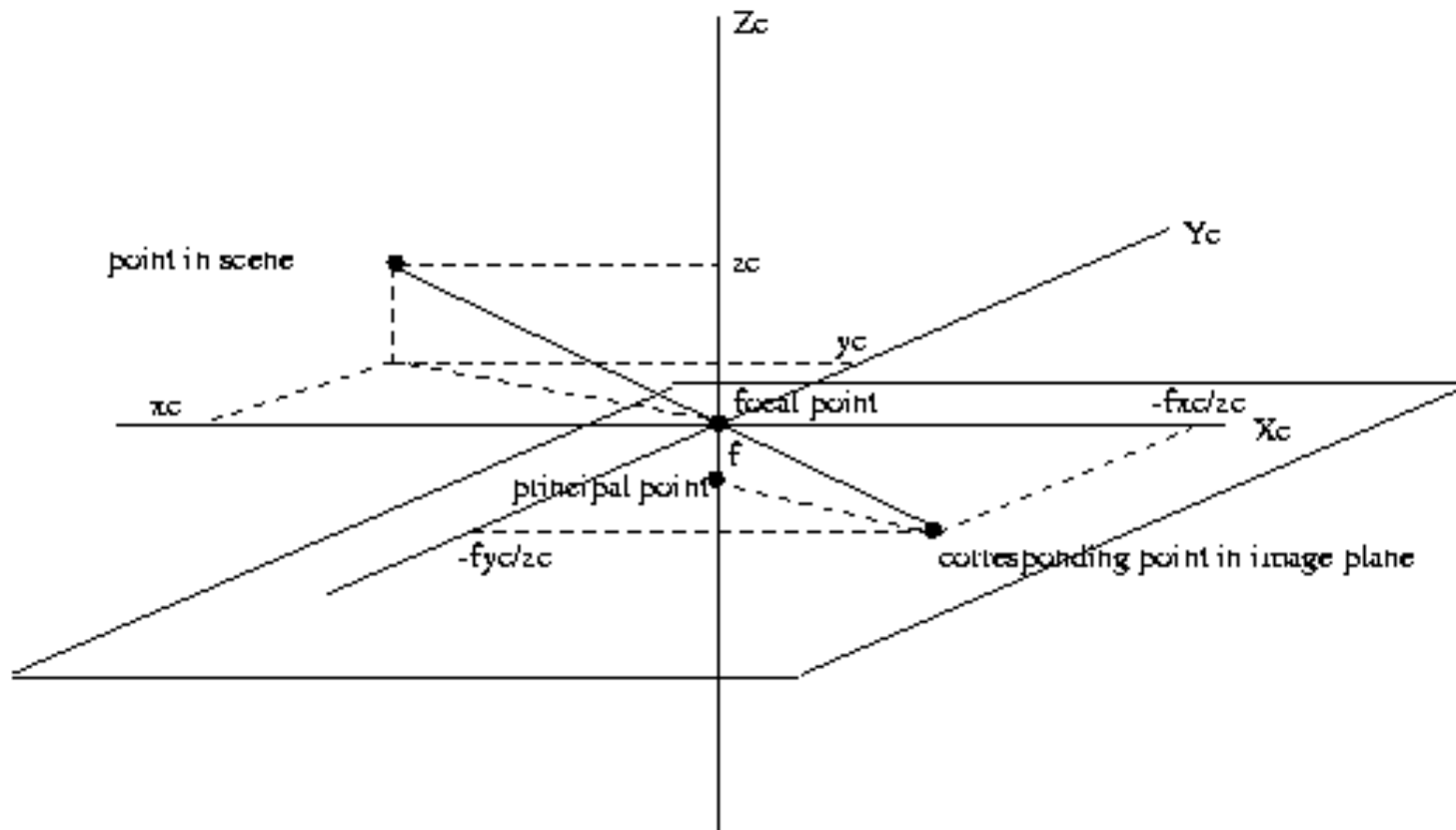
$$X_c = R(X_w - t)$$

These two 3D Euclidean coordinate systems differ by a translation of the origin, and a rotation in  $R^3$

- $X_w = [x_w \ y_w \ z_w]^T$  is a 3D point expressed in world coordinates
- $X_c = [x_c \ y_c \ z_c]^T$  is the same point in camera coordinates
- $R$  is the 3x3 rotation matrix (pitch, roll, yaw)
- $t$  is the translation vector

## Step 2: Project the point in camera coordinates onto the image plane, keeping camera coordinates

The coordinates of the corresponding point in the image plane, retaining camera coordinates, are  $U_c = [-fx_c/z_c \ -fy_c/z_c \ -f]^T$



**Step 3:  $U_c \rightarrow (u,v)$ , the image affine coordinates in the image plane.**

**The image affine coordinates are related to the camera coordinates by a translation of the origin, scaling, and possible shear of the x-axis (rotation of the x axis with respect to the y axis).**

$$[u \ v]^T = S [-f_x/z_c \ -f_y/z_c]^T - [u_0 \ v_0]^T$$

**where**

**$[u \ v]^T$  = final image affine coords in the image plane**

**$S$  = 2x2 matrix of form  $[a \ b; 0 \ c]$**

**$[u_0 \ v_0]^T$  = principal point expressed in image affine coordinates.**

# Augmented Reality

- MagicBook developed at the Human Interface Technology Laboratory of the University of Washington (HITLab)
- HITLab which they made available freely under the GNU ARToolKit, short for Augmented Reality Toolkit, is a software library written in C and C++ that can be used for building AR applications.
- Primarily developed by Dr. Hirokazu Kato of Osaka University, Japan, and HITLab at the University of Washington
- the OpenGL Utility Toolkit
- Mixed Reality Toolkit C++ Library developed at the University College London which uses Microsoft DirectX GLUT, and the OpenIllusionist library which uses wxWidgets for multiplatform GUI support

# Implementing Augmented Reality

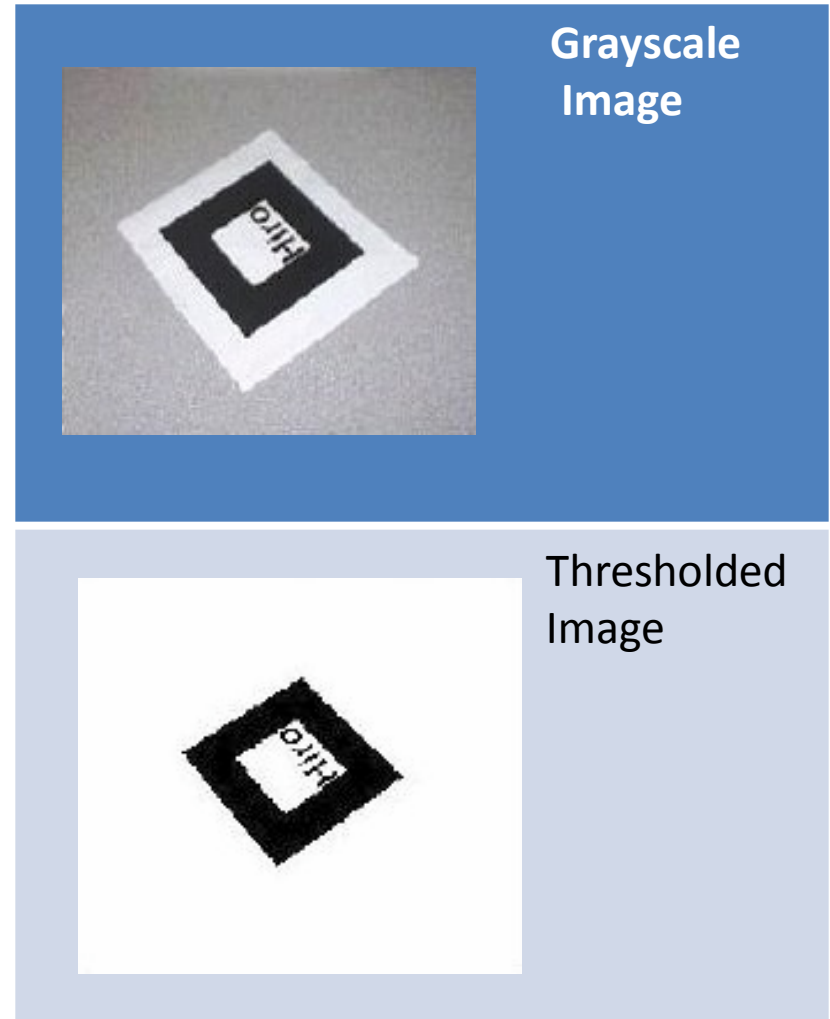
# Computer Vision

Computer vision is achieved by ARToolKit based on a basic corner detection approach with a fast pose estimation algorithm.

- Binarization and Thresholding
- Image Labeling
- Contour Extraction and Corner Detection
- Pattern Normalization and Template Matching
- Pose and Position Estimation
- Augmented Rendering

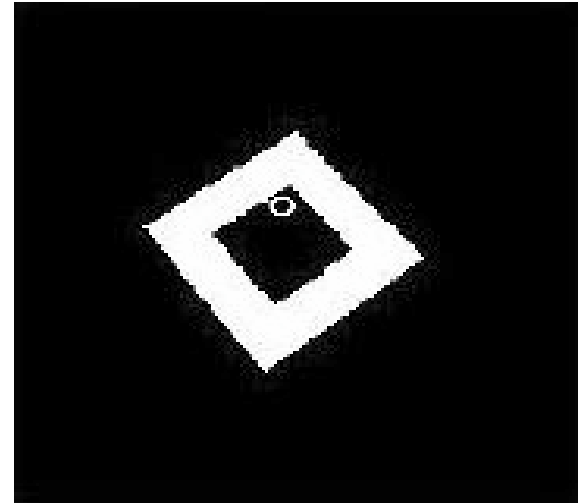
# Binarization and Thresholding

- Each frame image from the video input is first converted to a binary image (grayscale)
- Then thresholded converting the grayscale image to monochrome image.



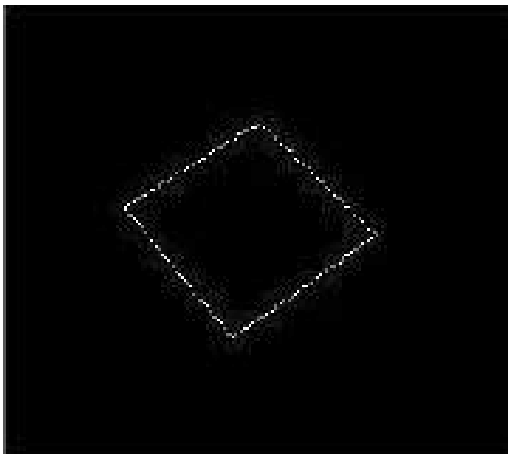
# Image Labeling

This phase of the computer vision algorithm extracts connected components from the thresholded image whose area is large enough to accommodate an image of the fiduciary marker.

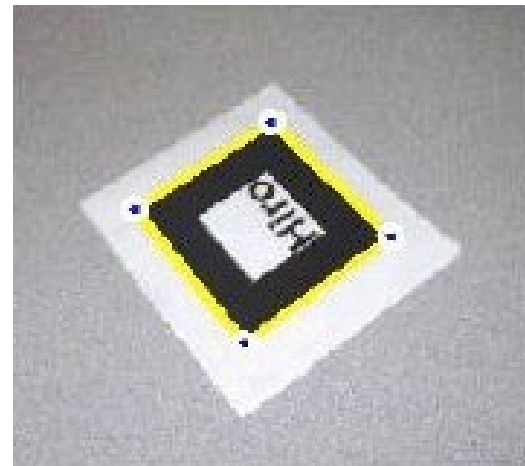


# Contour Extraction and Corner Detection

The outline contours of the labeled image are detected and regions whose contour can be fitted by four line segments are extracted. Parameters of these four line segments and coordinates of the four vertices of the regions found from the intersection of the line segments are extracted.



Contours



Extracted marker edges and corners

# Pattern Normalization and Template Matching

The regions found from contour extraction above are normalized and the sub-image within the region is compared by template matching with the fiduciary marker pattern specified by the user.

Pattern normalization and template matching

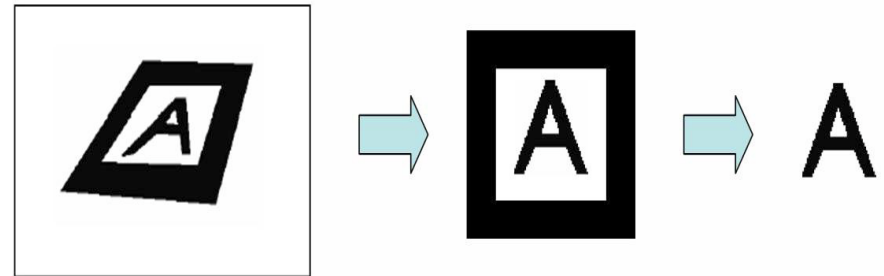
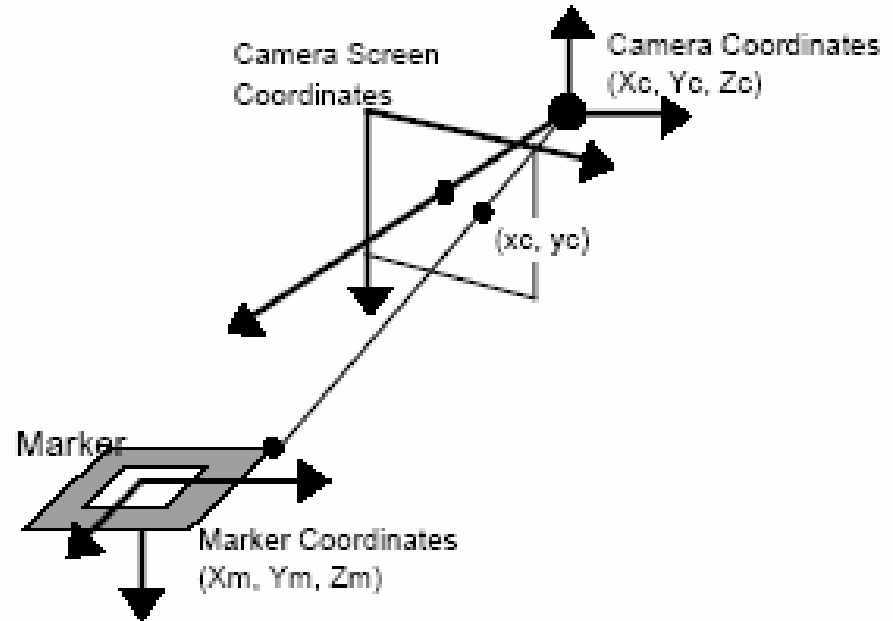


Image courtesy ARToolkit documentation



Marker coordinates and camera coordinates

# Pose and Position Estimation

The estimation by image analysis of the transformation matrix which represents the rotation R and translation T from the marker coordinates to the camera coordinates is called pose and position estimation.

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix}$$

**Two parallel sides of a square marker are projected on the image, the equations of those line segments in the camera screen coordinates are**

$$a_1x + b_1y + c_1 = 0, \quad a_2x + b_2y + c_2 = 0$$

For a marker that has been detected, the value of these parameters  $a_1, b_1, c_1$  and  $a_2, b_2, c_2$  is calculated during the contour extraction process. Also, the perspective projection matrix  $P$  is retrieved during camera calibration

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} & 0 \\ 0 & P_{22} & P_{23} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} hx_c \\ hyc \\ h \\ 1 \end{bmatrix} = P \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

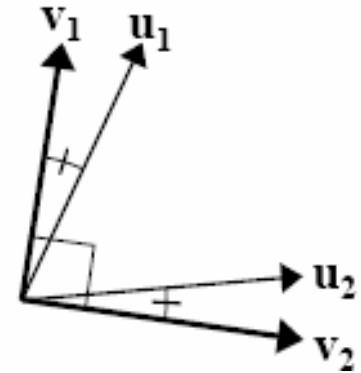
Substituting  $X_c$  and  $Y_c$  for  $X$  and  $Y$

Two parallel sides of the square marker can be represented in the camera coordinates frame by

$$a_1 P_{11} X_c + (a_1 P_{12} + b_1 P_{22}) Y_c + (a_1 P_{13} + b_1 P_{23} + c_1) Z_c = 0$$

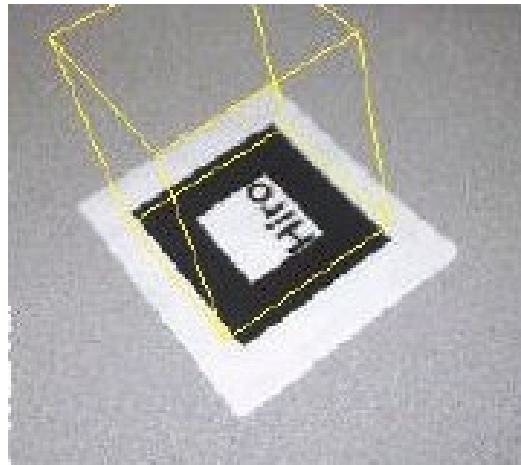
$$a_2 P_{11} X_c + (a_2 P_{12} + b_2 P_{22}) Y_c + (a_2 P_{13} + b_2 P_{23} + c_2) Z_c = 0$$

- The normal vectors of the planes are  $n_1$  and  $n_2$  respectively
- The square marker having four sides will have two sets of two parallel sides and therefore will yield two unit direction vectors  $u_1$  and  $u_2$ .
- To compensate for error, two perpendicular unit direction vectors  $v_1$  and  $v_2$  are defined in the plane that includes  $u_1$  and  $u_2$  as depicted
- Perpendicular to both  $v_1$  and  $v_2$  is  $v_3$  and gives the rotation component  $R_{3 \times 3}$  in the transformation matrix from marker coordinates to camera coordinates
- Four vertices of the marker in the marker coordinate system and the four vertices in the camera screen coordinate system, eight equations containing the translation component  $T_x$ ,  $T_y$ , and  $T_z$

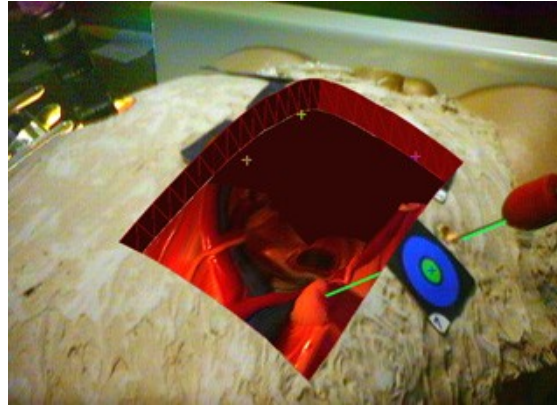


# Augmented Rendering

The virtual object is subjected to the transformation matrix obtained so that it is fitted with the fiduciary marker in the scene. Finally, the 3D object is rendered in the video frame on the video output device.



# Simulators



**Augmented Reality Visualization for Laparoscopic Surgery**  
**Ramesh Raskar MIT Media Lab**

<http://web.media.mit.edu/~raskar/>

**ARToolKit**  
**HITZ**  
**Lab**



[http://studierstube.icg.tugraz.ac.at/handheld\\_ar/businesscards.php](http://studierstube.icg.tugraz.ac.at/handheld_ar/businesscards.php)

# References

**Dr. Peter Scott, Animate Vision Principles for 3D Image Sequences,**

<http://www.cse.buffalo.edu/faculty/peter/cse668/>

**Augmented Reality Tool Kit (ARToolKit)**

<http://www.hitl.washington.edu/artoolkit/>

**Dr. Ramesh Raskar, MIT Media Lab**

[http://www.media.mit.edu/people/bio\\_raskar.html](http://www.media.mit.edu/people/bio_raskar.html)