Class overview

Administrivia & Policies

Lecture 1
  – Perspective projection (review)
  – Rigid motions (review)
  – Camera Calibration

Readings: Forsythe & Ponce, 1.1, 2.1, 2.2, 2.3, 3.1, 3.2
Vision

• What does it mean, to see? “to know what is where by looking”.
• How to discover from images what is present in the world, where things are, what actions are taking place.

from Marr, 1982
Why study Computer Vision?

• One can “see the future” (and avoid bad things…)!  
• Images and movies are everywhere; fast-growing collection of useful applications  
  – building representations of the 3D world from pictures  
  – automated surveillance (who’s doing what)  
  – movie post-processing  
  – face finding  
• Greater understanding of human vision  
• Various deep and attractive scientific mysteries  
  – how does object recognition work?
Why study Computer Vision?

• People draw distinctions between what is seen
  – “Object recognition”
  – This could mean “is this a fish or a bicycle?”
  – It could mean “is this George Washington?”
  – It could mean “is this poisonous or not?”
  – It could mean “is this slippery or not?”
  – It could mean “will this support my weight?”
  – Great mystery
    • How to build programs that can draw useful distinctions based on image properties.
Computer vision class, fast-forward
Cameras, lenses, and sensors

• Pinhole cameras
• Lenses
• Projection models
• Geometric camera parameters

Figure 1.16 The first photograph on record, la table servie, obtained by Nicéphore Niepce in 1822. Collection Harlinge–Viollet.

Image filtering

- Review of linear systems, convolution
- Bandpass filter-based image representations
- Probabilistic models for images

Oriented, multi-scale representation
4.1 **NEWTON'S SUMMARY DRAWING** of his experiments with light. Using a point source of light and a prism, Newton separated sunlight into its fundamental components. By reconverging the rays, he also showed that the decomposition is reversible.

From *Foundations of Vision*, by Brian Wandell, Sinauer Assoc., 1995
Models of texture

A Parametric Texture Model based on Joint Statistics of Complex Wavelet Coefficients


Parametric model

Non-parametric model
Statistical classifiers

- MIT Media Lab face localization results.
- Applications: database search, human machine interaction, video conferencing.
Multi-view Geometry

What are the relationships between images of point features in more than one view?

Given a point feature in one camera view, predict it’s location in a second (or third) camera?
Ego-Motion / “Match-move”

Where are the cameras?

Track points, estimate consistent poses…

Render synthetic objects in real world!
Ego-Motion / “Match-move”

*Video*

See “Harts War” and other examples in Gallery of examples for Matchmove program at www.realviz.com
Structure from Motion

What is the shape of the scene?
Segmentation

How many ways can you segment six points?

(or curves)
Segmentation

• Which image components “belong together”?
• Belong together=lie on the same object
• Cues
  – similar colour
  – similar texture
  – not separated by contour
  – form a suggestive shape when assembled
Tracking

Follow objects and estimate location..
- radar / planes
- pedestrians
- cars
- face features / expressions

Many ad-hoc approaches...

General probabilistic formulation: model density over time.
Tracking

• Use a model to predict next position and refine using next image

• Model:
  – simple dynamic models (second order dynamics)
  – kinematic models
  – etc.

• Face tracking and eye tracking now work rather well
Articulated Models

Find most likely model consistent with observations…. (and previous configuration)
Articulated tracking

- Constrained optimization
- Coarse-to-fine part iteration
- Propagate joint constraints through each limb
- Real-time on Ghz pentium…
slow
And…

• Visual Category Learning
• Image Databases
• Image-based Rendering
• Visual Speechreading
• Medical Imaging
Administrivia

- Syllabus
- Grading
- Collaboration Policy
- Project
<table>
<thead>
<tr>
<th>Lecture</th>
<th>Date</th>
<th>Description</th>
<th>Readings</th>
<th>Assignments</th>
<th>Material</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2/3</td>
<td>Course Introduction\Cameras, Lenses and Sensors</td>
<td>Req: FP 1.1, 2.1, 2.2, 2.3, 3.1, 3.2</td>
<td>PS0 out</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2/5</td>
<td>Image Filtering</td>
<td>Req: FP 7.1 - 7.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2/10</td>
<td>Image Representations: pyramids</td>
<td>Req: FP 7.7, 9.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2/12</td>
<td>Texture</td>
<td>Req: FP 9.1, 9.3, 9.4</td>
<td>PS0 due</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2/17</td>
<td>Monday Classes Held (NO LECTURE)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2/19</td>
<td>Color</td>
<td>Req: FP 6.1-6.4</td>
<td>PS1 out</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2/24</td>
<td>Local Features</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2/26</td>
<td>Multiview Geometry</td>
<td>Req: FP 10</td>
<td>PS1 due</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>3/2</td>
<td>Affine Reconstruction</td>
<td>Req: FP 12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>3/4</td>
<td>Projective Reconstruction</td>
<td>Req: FP 13</td>
<td>PS2 out</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>3/9</td>
<td>Scene Reconstruction</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>3/11</td>
<td>Non-Rigid Motion</td>
<td></td>
<td>PS2 due</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>3/16</td>
<td>Morphable and Active Appearance Models</td>
<td></td>
<td>EX1 out</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>3/18</td>
<td>Model-Based Object Recognition</td>
<td></td>
<td>EX1 due</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3/23-3/25</td>
<td>Spring Break (NO LECTURE)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>Topic</td>
<td>Notes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------</td>
<td>--------------------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3/23-3/25</td>
<td>Spring Break (NO LECTURE)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3/30</td>
<td>Face Detection and Recognition I</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4/1</td>
<td>Face Detection and Recognition II</td>
<td>Project proposal due</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4/6</td>
<td>Category Learning</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4/9</td>
<td>Segmentation I</td>
<td>PS3 out</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4/13</td>
<td>Segmentation II</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4/15</td>
<td>Medical Imaging</td>
<td>PS3 due</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4/20</td>
<td>Tracking I</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4/22</td>
<td>Tracking II</td>
<td>PS4 out</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4/27</td>
<td>Image-Based Rendering</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4/29</td>
<td>Example-based inference</td>
<td>PS4 due</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5/4</td>
<td>Multimodal Interfaces</td>
<td>EX2 out</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5/6</td>
<td>Image Databases</td>
<td>EX2 due</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5/11</td>
<td>Project Presentations 11-2pm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5/13</td>
<td>Projects Due--no class</td>
<td>Project final report due</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(extension to 5/16 on request)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Grading

- Two take-home exams
- Five problem sets with lab exercises in Matlab
- No final exam
- Final project
Collaboration Policy

Problem sets may be discussed, but all written work and coding must be done individually. Take-home exams may not be discussed. Individuals found submitting duplicate or substantially similar materials due to inappropriate collaboration may get an F in this class and other sanctions.
Project

The final project may be

- An original implementation of a new or published idea
- A detailed empirical evaluation of an existing implementation of one or more methods
- A paper comparing three or more papers not covered in class, or surveying recent literature in a particular area

A project proposal not longer than two pages must be submitted and approved by April 1st.
Problem Set 0

• Out today, due 2/12
• Matlab image exercises
  – load, display images
  – pixel manipulation
  – RGB color interpolation
  – image warping / morphing with interp2
  – simple background subtraction
• All psets graded loosely: check, check-, 0.
• (Outstanding solutions get extra credit.)
Map showing 400 Technology Square

The building says "Forrester" on the side. Only the parking garage side building entrance is unlocked. (After normal business hours, the elevator to our floor and the building itself are both locked.) Exiting the elevator on the 6th floor, you'll see a pair of glass doors on one side. Enter the left glass door, then turn right at every opportunity to find my office, room 601.

back to my home page, Sept., 2002.
Cameras, lenses, and calibration

Today:

• Camera models (review)
• Projection equations (review)

You should have been exposed to this material in previous courses; this lecture is just a (quick) review.

• Calibration methods (new)
7-year old’s question

Why is there no image on a white piece of paper?
Virtual image, perspective projection

- Abstract camera model - box with a small hole in it
They are formed by the projection of 3D objects.

Images are two-dimensional patterns of brightness values.

Animal eye: a looonnng time ago.

Photographic camera: Niepce, 1816.

Pinhole perspective projection: Brunelleschi, XVth Century.
Camera obscura: XVIth Century.
The equation of projection

\[
\begin{align*}
    x' &= f' \frac{x}{z} \\
    y' &= f' \frac{y}{z}
\end{align*}
\]
Distant objects are smaller
Geometric properties of projection

- Points go to points
- Lines go to lines
- Planes go to the whole image or a half-plane
- Polygons go to polygons
- Degenerate cases
  - line through focal point to point
  - plane through focal point to line

\[
\begin{align*}
x' &= f' \frac{x}{z} \\
y' &= f' \frac{y}{z}
\end{align*}
\]
Parallel lines meet

Common to draw film plane *in front* of the focal point.
Moving the film plane merely scales the image.
Vanishing points

- Each set of parallel lines (=direction) meets at a different point
  - The *vanishing point* for this direction
- Sets of parallel lines on the same plane lead to *collinear* vanishing points.
  - The line is called the *horizon* for that plane
What if you photograph a brick wall head-on?
Two-point perspective

It's easy to draw simple forms in two-point perspective.

Linear perspective allows artists to trick the eye into seeing depth on a flat surface.

1. Draw a horizon line.

2. Make a vanishing point.

3. Draw a square or rectangle.

4. Draw orthogonal lines from shape corners to vanishing point.

5. Draw a horizontal line to end your form.

6. Draw a vertical line to make the form's side.

7. Erase the orthogonal lines.

8. Now you have a 3-D form in one-point perspective!

9. Add another form.

10. Add windows and doors.

11. Try a lower horizon line.

12. Try stacking forms!

http://www.sanford-artedventures.com/create/tech_1pt_perspective.html
Draw a horizon line.

Connect top corners to opposite vanishing points.

Draw lightly so you can erase!

Draw two vanishing points on the horizon line near the page edges.

Draw a vertical line for the front edge of your form.

Erase extra orthogonal lines.

Try stacking forms.

Try a lower horizon line.

Add windows and doors!

Draw more forms!
Weak perspective

• Issue
  – perspective effects, but not over the scale of individual objects
  – collect points into a group at about the same depth, then divide each point by the depth of its group
  – Adv: easy
  – Disadv: wrong
Orthographic projection
How large a pinhole?
2.18 DIFFRACTION LIMITS THE QUALITY OF PINHOLE OPTICS. These three images of a bulb filament were made using pinholes with decreasing size. (A) When the pinhole is relatively large, the image rays are not properly converged, and the image is blurred. (B) Reducing the size of the pinhole improves the focus. (C) Reducing the size of the pinhole further worsens the focus, due to diffraction. From Ruechardt, 1958.
The reason for lenses
Water glass refraction
The thin lens, first order optics

\[
\frac{1}{z'} - \frac{1}{z} = \frac{1}{f} \quad \text{or} \quad f = \frac{R}{2(n-1)}
\]

All rays through P also pass through P', but only for points at -z: "depth of field".

Forsyth&Ponce
More accurate models of real lenses

- Finite lens thickness
- Higher order approximation to $\sin(\theta)$
- Chromatic aberration
- Vignetting
Thick lens

Figure 1.11  A simple thick lens with two spherical surfaces.
Lens systems can be designed to correct for aberrations described by 3rd order optics.
Vignetting
Chromatic aberration

(great for prisms, bad for lenses)
Other (possibly annoying) phenomena

• Chromatic aberration
  – Light at different wavelengths follows different paths; hence, some wavelengths are defocussed
  – Machines: coat the lens
  – Humans: live with it

• Scattering at the lens surface
  – Some light entering the lens system is reflected off each surface it encounters (Fresnel’s law gives details)
  – Machines: coat the lens, interior
  – Humans: live with it (various scattering phenomena are visible in the human eye)
Summary so far

• Want to make images
• Pinhole camera models the geometry of perspective projection
• Lenses make it work in practice
• Models for lenses
  – Thin lens, spherical surfaces, first order optics
  – Thick lens, higher-order optics, vignetting.
Some background material…

- Rigid motion: translation and rotation
- Homogenous coordinates
Translation

\[
^A P = \begin{pmatrix} A_x \\ A_y \\ A_z \end{pmatrix}, \quad ^B P = \begin{pmatrix} B_x \\ B_y \\ B_z \end{pmatrix}
\]

How does \(^B P\) relate to \(^A P\)?

\[
^B P = ^A P + ^B O_A
\]
$^A P = \begin{pmatrix} A_x \\ A_y \\ A_z \end{pmatrix}$  
$^B P = \begin{pmatrix} B_x \\ B_y \\ B_z \end{pmatrix}$

How does $^B P$ relate to $^A P$?

$^B P = ^A R _A ^A P$
Find the rotation matrix

Project \( \overrightarrow{OP} = \begin{pmatrix} i_A & \hat{j}_A & \hat{k}_A \end{pmatrix} \begin{pmatrix} A_X \\ A_Y \\ A_Z \end{pmatrix} \)

onto the B frame’s coordinate axes.

\[
\begin{pmatrix} B_X \\ B_Y \\ B_Z \end{pmatrix} = \begin{pmatrix} \hat{i}_B \cdot \hat{i}_A A_X & \hat{i}_B \cdot \hat{j}_A A_Y & \hat{i}_B \cdot \hat{k}_A A_Z \\ \hat{j}_B \cdot \hat{i}_A A_X & \hat{j}_B \cdot \hat{j}_A A_Y & \hat{j}_B \cdot \hat{k}_A A_Z \\ \hat{k}_B \cdot \hat{i}_A A_X & \hat{k}_B \cdot \hat{j}_A A_Y & \hat{k}_B \cdot \hat{k}_A A_Z \end{pmatrix}
\]
Rotation matrix

\[
\begin{pmatrix}
B_X \\
B_Y \\
B_Z
\end{pmatrix} = \begin{pmatrix}
\hat{i}_B \cdot \hat{i}_A A_X & \hat{i}_B \cdot \hat{j}_A A_Y & \hat{i}_B \cdot \hat{k}_A A_Z \\
\hat{j}_B \cdot \hat{i}_A A_X & \hat{j}_B \cdot \hat{j}_A A_Y & \hat{j}_B \cdot \hat{k}_A A_Z \\
\hat{k}_B \cdot \hat{i}_A A_X & \hat{k}_B \cdot \hat{j}_A A_Y & \hat{k}_B \cdot \hat{k}_A A_Z
\end{pmatrix}
\]

implies

\[
B P = B R \ A P
\]

where

\[
B A R = \begin{pmatrix}
\hat{i}_B \cdot \hat{i}_A & \hat{i}_B \cdot \hat{j}_A & \hat{i}_B \cdot \hat{k}_A \\
\hat{j}_B \cdot \hat{i}_A & \hat{j}_B \cdot \hat{j}_A & \hat{j}_B \cdot \hat{k}_A \\
\hat{k}_B \cdot \hat{i}_A & \hat{k}_B \cdot \hat{j}_A & \hat{k}_B \cdot \hat{k}_A
\end{pmatrix}
\]
Translation and rotation

Let’s write

\[ BP = ^B R \begin{bmatrix} \begin{pmatrix} B_X \\ B_Y \\ B_Z \\ 1 \end{pmatrix} \end{bmatrix} \begin{bmatrix} \begin{pmatrix} B \end{pmatrix} \end{bmatrix} + ^B O_A \begin{bmatrix} \begin{pmatrix} A_X \\ A_Y \\ A_Z \\ 1 \end{pmatrix} \end{bmatrix} \]

as a single matrix equation:
Homogenous coordinates

• Add an extra coordinate and use an equivalence relation

• for 3D
  – equivalence relation
    \( k^*(X,Y,Z,T) \) is the same as
    \( (X,Y,Z,T) \)

• Motivation
  – Possible to write the action of a perspective camera as a matrix
Homogenous/non-homogenous transformations for a 3-d point

- From non-homogenous to homogenous coordinates: add 1 as the 4\text{th} coordinate, ie
  \begin{pmatrix}
  x \\
  y \\
  z \\
  1
  \end{pmatrix}
  \rightarrow
  \begin{pmatrix}
  x \\
  y \\
  z \\
  1
  \end{pmatrix}

- From homogenous to non-homogenous coordinates: divide 1\text{st} 3 coordinates by the 4\text{th}, ie
  \begin{pmatrix}
  x \\
  y \\
  z \\
  T
  \end{pmatrix}
  \rightarrow
  \frac{1}{T}
  \begin{pmatrix}
  x \\
  y \\
  z
  \end{pmatrix}
Homogenous/non-homogenous transformations for a 2-d point

• From non-homogenous to homogenous coordinates: add 1 as the 3\(^{rd}\) coordinate, ie

\[
\begin{pmatrix}
x \\
y
\end{pmatrix} \rightarrow
\begin{pmatrix}
x \\
y \\
1
\end{pmatrix}
\]

• From homogenous to non-homogenous coordinates: divide 1\(^{st}\) 2 coordinates by the 3\(^{rd}\), ie

\[
\begin{pmatrix}
x \\
y \\
z
\end{pmatrix} \rightarrow
\frac{1}{z}
\begin{pmatrix}
x \\
y
\end{pmatrix}
\]
The camera matrix, in homogenous coordinates

- Turn previous expression into HC’s
  - HC’s for 3D point are (X,Y,Z,T)
  - HC’s for point in image are (U,V,W)

\[
\begin{pmatrix}
X \\
Y \\
\frac{Z}{f}
\end{pmatrix} =
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1/f & 0
\end{pmatrix}
\begin{pmatrix}
X \\
Y \\
Z \\
T
\end{pmatrix}
\]

What about an orthographic camera?

HC          Non-HC
The projection matrix for orthographic projection, homogenous coordinates

\[
\begin{pmatrix}
U \\
V \\
W
\end{pmatrix}
=
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
X \\
Y \\
Z \\
T
\end{pmatrix}
\]

\[
= \begin{pmatrix}
X \\
Y \\
T
\end{pmatrix}
\rightarrow \frac{1}{T}
\begin{pmatrix}
X \\
Y
\end{pmatrix}
\]

HC       Non-HC
Camera calibration

Use the camera to tell you things about the world:

– Relationship between coordinates in the world and coordinates in the image: *geometric camera calibration*.

– (Relationship between intensities in the world and intensities in the image: *photometric camera calibration*, not covered in this course, see 6.801 or text)
Intrinsic parameters

Perspective projection

\[ u = f \frac{x}{z} \]

\[ v = f \frac{y}{z} \]
Intrinsic parameters

But “pixels” are in some arbitrary spatial units…

\[ u = f \frac{x}{z} \]
\[ v = f \frac{y}{z} \]
Intrinsic parameters

But “pixels” are in some arbitrary spatial units

\[ u = \alpha \frac{x}{z} \]

\[ v = \alpha \frac{y}{z} \]
Intrinsic parameters

Maybe pixels are not square...

\[ u = \alpha \frac{x}{z} \]

\[ v = \alpha \frac{y}{z} \]
Intrinsic parameters

Maybe pixels are not square

\[ u = \alpha \frac{x}{z} \]

\[ v = \beta \frac{y}{z} \]
Intrinsic parameters

We don’t know the origin of our camera pixel coordinates…

\[ u = \alpha \frac{x}{z} \]

\[ v = \beta \frac{y}{z} \]
Intrinsic parameters

We don’t know the origin of our camera pixel coordinates

\[ u = \alpha \frac{x}{z} + u_0 \]

\[ v = \beta \frac{y}{z} + v_0 \]
May be skew between camera pixel axes…

\[
\begin{align*}
  u & = \alpha \frac{x}{z} + u_0 \\
  v & = \beta \frac{y}{z} + v_0
\end{align*}
\]
May be skew between camera pixel axes

\[ u = \alpha \frac{x}{z} - \alpha \cot(\theta) \frac{y}{z} + u_0 \]

\[ v = \frac{\beta}{\sin(\theta)} \frac{y}{z} + v_0 \]
Intrinsic parameters

Using homogenous coordinates, we can write this as:

\[
\begin{pmatrix}
    u \\
    v \\
    1
\end{pmatrix} = \frac{1}{z} \begin{pmatrix}
    \alpha & -\alpha \cot(\theta) & u_0 & 0 \\
    0 & \beta & v_0 & 0 \\
    0 & \frac{\beta}{\sin(\theta)} & 1 & 0 \\
    0 & 0 & 0 & 1
\end{pmatrix} \begin{pmatrix}
    x \\
    y \\
    z \\
    1
\end{pmatrix}
\]

or:

\[
\vec{p} = \frac{1}{z} \begin{pmatrix} K & \vec{0} \end{pmatrix} \vec{P}
\]
Extrinsic parameters: translation and rotation of camera frame

\[ C_P = C_R W P + C O_W \]

\[
\begin{pmatrix}
C_X \\
C_Y \\
C_Z \\
1
\end{pmatrix} =
\begin{pmatrix}
- & - & - & | \\
- & C_R & - & C O_W & | \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
W_X \\
W_Y \\
W_Z \\
1
\end{pmatrix}
\]

Non-homogeneous coordinates

Homogeneous coordinates

Block matrix form
Combining extrinsic and intrinsic calibration parameters

\[ \mathbf{\tilde{p}} = \frac{1}{z} \left( \mathbf{K} \ 0 \right) \mathbf{\tilde{P}} \quad \text{Intrinsic} \]

\[ \mathbf{^{C}P} = \mathbf{^{C}VR} \mathbf{^{W}P} + \mathbf{^{C}O}_{W} \quad \text{Extrinsic} \]

\[ \mathbf{\tilde{p}} = \frac{1}{z} \mathbf{K} \left( \mathbf{^{C}VR} \mathbf{^{C}O}_{W} \right) \mathbf{\tilde{P}} \]

\[ \mathbf{\tilde{p}} = \frac{1}{z} \mathbf{M} \mathbf{\tilde{P}} \]

Forsyth\&Ponce
Other ways to write the same equation

\[ \vec{p} = \frac{1}{z} M \vec{P} \]

\[
\begin{pmatrix}
  u \\
  v \\
  1
\end{pmatrix} = \frac{1}{z}
\begin{pmatrix}
  \cdot & m_1^T & \cdot \\
  \cdot & m_2^T & \cdot \\
  \cdot & m_3^T & \cdot \\
  1 & & 1
\end{pmatrix}
\begin{pmatrix}
  W_x \\
  W_y \\
  W_z \\
  1
\end{pmatrix}
\]

\[
\begin{aligned}
  u &= \frac{m_1 \cdot \vec{P}}{m_3 \cdot \vec{P}} \\
  v &= \frac{m_2 \cdot \vec{P}}{m_3 \cdot \vec{P}}
\end{aligned}
\]

\( z \) is in the *camera* coordinate system, but we can solve for that, since \( 1 = \frac{m_3 \cdot \vec{P}}{z} \), leading to:
Calibration target

The Opti-CAL Calibration Target Image

http://www.kinetic.bc.ca/CompVision/opti-CAL.html
Camera calibration

From before, we had these equations relating image positions, u,v, to points at 3-d positions P (in homogeneous coordinates):

\[ u = \frac{m_1 \cdot \vec{P}}{m_3 \cdot \vec{P}} \]
\[ v = \frac{m_2 \cdot \vec{P}}{m_3 \cdot \vec{P}} \]

So for each feature point, i, we have:

\[ (m_1 - u_i m_3) \cdot \vec{P}_i = 0 \]
\[ (m_2 - v_i m_3) \cdot \vec{P}_i = 0 \]
Camera calibration

Stack all these measurements of $i=1\ldots n$ points

\[
(m_1 - u_i m_3) \cdot \vec{P}_i = 0
\]

\[
(m_2 - v_i m_3) \cdot \vec{P}_i = 0
\]

into a big matrix:

\[
\begin{pmatrix}
P_1^T & 0^T & -u_1 P_1^T \\
0^T & P_1^T & -v_1 P_1^T \\
\vdots & \vdots & \vdots \\
P_n^T & 0^T & -u_n P_n^T \\
0^T & P_n^T & -v_n P_n^T
\end{pmatrix}
\begin{pmatrix}
m_1 \\
m_2 \\
m_3
\end{pmatrix}
=
\begin{pmatrix}
0 \\
0 \\
0
\end{pmatrix}
\]
Camera calibration

In vector form:

\[
\begin{pmatrix}
P_1^T & 0^T & -u_1P_1^T \\
0^T & P_1^T & -v_1P_1^T \\
\vdots & \vdots & \vdots \\
P_n^T & 0^T & -u_nP_n^T \\
0^T & P_n^T & -v_nP_n^T
\end{pmatrix}
\begin{pmatrix}
m_1 \\
m_2 \\
m_3
\end{pmatrix}
= 
\begin{pmatrix}
0 \\
0 \\
\vdots \\
0
\end{pmatrix}
\]

Showing all the elements:

\[
\begin{pmatrix}
P_{1x} & P_{1y} & P_{1z} & 1 & 0 & 0 & 0 & 0 & -u_1P_{1x} & -u_1P_{1y} & -u_1P_{1z} & -u_1 \\
0 & 0 & 0 & P_{1x} & P_{1y} & P_{1z} & 1 & -v_1P_{1x} & -v_1P_{1y} & -v_1P_{1z} & -v_1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
P_{nx} & P_{ny} & P_{nz} & 1 & 0 & 0 & 0 & 0 & -u_nP_{nx} & -u_nP_{ny} & -u_nP_{nz} & -u_n \\
0 & 0 & 0 & P_{nx} & P_{ny} & P_{nz} & 1 & -v_nP_{nx} & -v_nP_{ny} & -v_nP_{nz} & -v_n
\end{pmatrix}
\begin{pmatrix}
m_{11} \\
m_{12} \\
m_{13} \\
m_{14} \\
m_{21} \\
m_{22} \\
m_{23} \\
m_{24} \\
m_{31} \\
m_{32} \\
m_{33} \\
m_{34}
\end{pmatrix}
= 
\begin{pmatrix}
0 \\
0 \\
\vdots \\
0
\end{pmatrix}
\]
We want to solve for the unit vector \( \mathbf{m} \) (the stacked one) that minimizes \( |P\mathbf{m}|^2 \).

The minimum eigenvector of the matrix \( P^TP \) gives us that (see Forsyth & Ponce, 3.1)

\[
\begin{pmatrix}
P_{xx} & P_{xy} & P_{xz} & 1 & 0 & 0 & 0 & 0 & -u_1P_{xx} & -u_1P_{xy} & -u_1P_{xz} & -u_1 \\
0 & 0 & 0 & 0 & P_{xx} & P_{xy} & P_{xz} & 1 & -v_1P_{xx} & -v_1P_{xy} & -v_1P_{xz} & -v_1 \\
P_{nx} & P_{ny} & P_{nz} & 1 & 0 & 0 & 0 & 0 & -u_nP_{nx} & -u_nP_{ny} & -u_nP_{nz} & -u_n \\
0 & 0 & 0 & 0 & P_{nx} & P_{ny} & P_{nz} & 1 & -v_nP_{nx} & -v_nP_{ny} & -v_nP_{nz} & -v_n
\end{pmatrix}
\begin{pmatrix}
m_{11} \\
m_{12} \\
m_{13} \\
m_{14} \\
m_{21} \\
m_{22} \\
m_{23} \\
m_{24} \\
m_{31} \\
m_{32} \\
m_{33} \\
m_{34}
\end{pmatrix}
= \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{pmatrix}
\]
Once you have the M matrix, can recover the intrinsic and extrinsic parameters as in Forsyth&Ponce, sect. 3.2.2.