Converting Jpeg to 2D Geometry in a CAD System

Brook Gotschall
California State University, Long Beach, California
bjgotcha@yahoo.com

May 14, 2009
Outline

- Background
- Motivation
- The Problem
- The Approach
- The Solution
- Conclusion
Background

➢ Mechanical Engineer
➢ Experience in aerospace
What’s inside
Some things are the same
A Generic Bracket

What is the most economical way to fabricate it?
Fabricating the Part

- Drawing produced by hand.
  - Full-scale.
  - Un-dimensioned.
- Supplied to fabricator.
- Image transferred to sheet metal.
- Sheet metal cut out by a skilled machinist.
Motivation

- Hand drawings are out.
- Electronic definition is in.
Electronic Definition

- Fabricators use automated machines.
- They require computer-based data.
- Answer:

**CAD (Computer-aided design):** The use of computer technology for the design of objects, real or virtual. [4]
Computer-Aided Design

Unigraphics

CATIA

AutoCAD
Why care about those old hand drawings?

- Need to replace a worn or broken part.
- Desire an improvement to an old design.
- New design interfaces with old.
Can’t just use the old data as is?

- Very difficult to find someone who still uses the older technology.
- Electronic copy of data reduces error and eliminates guess-work.
So…

➢ We need our Jpeg image…
➢ Translated into CAD data…
The Problem

- How to translating Jpeg to CAD?
  - Translate the pixel coordinates as points.
  - Extract the lines, arcs, and splines, then translate.
Pixel Coordinates as Points

➤ Looking at the image in Matlab...

➤ The white pixels have a value of 255.

➤ The “black” pixels are... ~76.
Translate pixel coordinates

- Any pixels less than 80:

10,766 points
Extract and Translate

- Clarify image.
- Identify the lines, arcs, and splines.
- Derive equations.
- Translate to CAD.
The Approach

- Find a method to clarify the image.
- Identify curve (line, arc, spline) geometry from pixel coordinates.
- Define which equations best fit or define a curve entity to the pixel coordinates.
- Generate a neutral format data file to translate the curve entities.
Clarify the Image

- Sobel Edge Detector
- Wavelet decomposition
- Sharpening
Sobel Edge Detector

- Sobel Edge Detector: 

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td></td>
</tr>
</tbody>
</table>

- Consider:
Sobel Edge Detector

-1 2 1
0 0 0
-1 -2 -1

convolve
Wavelet Decomposition

- Add the horizontal and vertical components.
Sharpening

Procedure:

1) Smooth the image with an averaging filter.
2) Subtract the smoothed image from the original.
3) Add this difference to the original. [9]

1-D Example:
Identifying a Curve

- **Bézier curve**: A parametric curve important in computer graphics and related fields.
- **Hough transform** (pronounced 'huff'): A feature extraction technique used in image analysis, computer vision, and digital image processing.
Hough Transform

- Polar form of equation for a line: \( x \cos \theta + y \sin \theta = \rho \)

- Create a matrix whose rows represent rho and columns represent theta.

- Define Rho between \(-\text{Maxrho}\) and \(+\text{Maxrho}\) where Maxrho is the diagonal length of the image.

- Define Theta between \(-90\) and \(+90\).

- This matrix represents all possible lines through the image.

- For each “line”, count the number of intersections with pixels in the image.
Derive Equations

- Use polyfit in Matlab.
  - Easily produces coefficients line and spline equations.
Neutral Format Data File

IGES (Initial Graphics Exchange Specification) (pronounced eye-jess): Defines a neutral data format that allows the digital exchange of information among Computer-aided design (CAD) systems.

Type 100: CIRCULAR ARC
Type 110: LINE
Type 116: POINT
The Solution

- Use sharpening to clarify the image.
- Use Hough Transform to identify curve geometry.
- Use polyfit to define curve entities.
- Use IGES to translate data for import to CAD.
Sharpening the image

1/9 X

Original

One pass

Two passes
Using Hough Transform

- Processing the image through a Hough Transform yielded the following:

  - Max number of intersections: 354
    at $\theta = -89.7^\circ$
    and $\rho = -707.7$
  - Next max: 311
    at $\theta = -89.7^\circ$
    and $\rho = -39$
Using Hough Transform

- Rotate $x,y$ coordinates of image by $-89.7^\circ$ and scatter plot:
Using Hough Transform

- The transform actually found two lines.

Nearly co-linear
Polyfit and clean up

- Extract points within a 6 pixel wide band around $\rho = -707.7$.
- Use polyfit with degree 1 to find a best fit line.
- Remove points which are too far from the line.
- Check for gaps and deal with individual line segments separately.
Curve definition and output

- Take line end points and reverse the rotation.
- Write those coordinates to a text file.
- Change the extension of the file from .txt to .igs.
- Import to CAD.
Conclusion

- Achieved:
  - Found a line.
  - Determined sufficient definition of the line for translation.

- Future:
  - Find the remainder of the linear entities.
  - Find arcs and splines.
  - Connect the ends of the lines, arcs, etc.
  - Do actual translation.
References

Nasa.gov
Space.com
Boeing.com
Wikipedia.org, “AutoCAD”, c. 2009
Wikipedia.org, “CATIA”, c. 2008
The Mathworks, Inc. c. 2009
Wikipedia.org, “Bézier curve”
Wikipedia.org, “IGES”