74.795 Local Vision: Edge Detection

Jacky Baltes
Department of Computer Science
University of Manitoba
Email: jacky@cs.umanitoba.ca
WWW: http://avocet.cs.umanitoba.ca
Optical Illusions: Subjective Contours

Kanizsa figure: subjective contours
Optical Illusions: Lines
System Model

- Image analysis
  - Preprocessing: remove noise and irrelevant information
  - Data reduction: reduce the data in the spatial or frequency domain
  - Feature analysis: extracted features are examined and evaluated
Zooming and Interpolation

- Zero order hold: repeat previous pixel values
- First-order hold: Bi-linear interpolation
  - First expand rows, then columns

Enlarge an N×N sized image to a size of (2N)×(2N).
Arbitrary Zooming Factors

Take two adjacent values and **linearly interpolate** more than one value between them (let \( k \) be the enlargement number)
1. Subtract the two adjacent values
2. Divide the value by \( k \)
3. Add the result to the smaller value, and keep adding the result from the second step in a running total until all \((k-1)\) intermediate pixel locations are filled.

\[
\begin{align*}
125 + 5 & \rightarrow 130 + 5 \rightarrow 135 \rightarrow 140 \\
(140-125)/3 & = 5
\end{align*}
\]
Translation

\[
\begin{bmatrix}
x' \\
y'
\end{bmatrix} = \begin{bmatrix}
T_x \\
T_y
\end{bmatrix} + \begin{bmatrix}
x \\
y
\end{bmatrix}
\]
Rotation

- Clockwise rotation of the image

\[
\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(t) & \sin(t) \\ -\sin(t) & \cos(t) \end{bmatrix} \ast \begin{bmatrix} x \\ y \end{bmatrix}
\]

- 3 x 3 matrix to represent 2D rotation and translation in homogenous coordinates
Edge Detection

- Colours are very susceptible to lighting
- An important pre-processing step is edge detection
- How do we find an edge?
  - Sharp contrast in the image
  - Derivative of the image function $I(i,j)$
  - Approximate derivative with $I(i+1,j) - I(i-1,j)$
Convolution

- Edge Detection and many other image preprocessing steps can be implemented as a convolution
- A convolution mask $M(r,c)$ is a matrix that is applied to each pixel in the image
- Specifies weights of the neighbors

\[
\sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} I(r-x, c-y)M(x, y)
\]
Convolution

- What is the output of $[0, 255, 0, 0, \ldots]$?
- -255?
- Use divisor and offset to normalize result of convolution to 0..255
- How to deal with colour images?
  - Convert to grey scale
  - Handle each channel separately
Sobel Edge Detection

- To reduce noise, average over several rows
- Weigh rows differently
- Divisor = 8, Offset = 128
Sobel Edge Detection

- Use separate convolution matrices for horizontal and vertical

<table>
<thead>
<tr>
<th>Horizontal</th>
<th>Vertical</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1 0 +1</td>
<td>-1 -2 -1</td>
</tr>
<tr>
<td>-2 0 +2</td>
<td>0 0 0</td>
</tr>
<tr>
<td>-1 0 +1</td>
<td>+1 +2 +1</td>
</tr>
</tbody>
</table>
**Convolution**

- Many other filters can be implemented efficiently as convolution
- One problem: what to do at the borders
- What does the following filter do?

<table>
<thead>
<tr>
<th>+1</th>
<th>+1</th>
<th>+1</th>
</tr>
</thead>
<tbody>
<tr>
<td>+1</td>
<td>+1</td>
<td>+1</td>
</tr>
<tr>
<td>+1</td>
<td>+1</td>
<td>+1</td>
</tr>
<tr>
<td>+1</td>
<td>+1</td>
<td>+1</td>
</tr>
</tbody>
</table>
Blurring (Simple)

- Blurring is used to reduce noise in the image

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>+1</td>
<td>+1</td>
<td>+1</td>
</tr>
<tr>
<td>+1</td>
<td>+1</td>
<td>+1</td>
</tr>
<tr>
<td>+1</td>
<td>+1</td>
<td>+1</td>
</tr>
</tbody>
</table>
**Convolution**

- If the coefficients of the mask sum to greater than 1, average brightness is increased, otherwise decreased.
- If the coefficients are alternating positive and negative, then edges are enhanced.
- If the coefficients are all positive, then the image will be blurred (edges will be reduced).
Template Matching Convolution

- Find Specific features in the image

Divisor = 4
Median Filter

• Blurring (Mean filter) is susceptible to single pixel outliers
• Median filter replaces the central pixel with the median of its neighbouring pixels

Sort the value – $3, 3, 4, 4, 5, 5, 5, 6, 7$
"What are the objects to be analyzed?"

Pre-processing, image enhancement

Binary operations

Morphological operations and feature extraction

Classification and matching
**Segmentation**

- **Full segmentation**: Individual objects are separated from the background and given individual ID numbers (labels).

- **Partial segmentation**: The amount of data is reduced (usually by separating objects from background) to speed up the further processing.

- Segmentation is often the most difficult problem to solve in the process; there is no universal solution!

- The problem can be made much easier if solved in cooperation with the constructor of the imaging system (choice of sensors, illumination, background etc).
Three Types of Segmentation

- **Classification** – Based on some similarity measure between pixel values. The simplest form is thresholding.

- **Edge-based** – Search for edges in the image. They are then used as borders between regions

- **Region-based** – Region growing, merge & split

Common idea: search for discontinuities or/and similitudes in the image
Thresholding (Global and Local)

- **Global**: based on some kind of histogram: grey-level, edge, feature etc.
  - Lighting conditions are extremely important, and it will only work under very controlled circumstances.

- **Fixed thresholds**: the same value is used in the whole image

- **Local (or dynamic thresholding)**: depends on the position in the image. The image is divided into overlapping sections which are thresholded one by one.
Thresholding

Select an initial estimate for $T$
Segment the image using $T$. This produces 2 groups: $G_1$, pixels with value $>T$ and $G_2$, with value $<T$
Compute $\mu_1$ and $\mu_2$, average pixel value of $G_1$ and $G_2$
New threshold: $T=1/2(\mu_1+\mu_2)$
Repeat steps 2 to 4 until $T$ stabilizes.

Very easy + very fast
Assumptions: normal dist. + low noise
Optimal Thresholding

- Based on the shape of the current image histogram. Search for valleys, Gaussian distributions etc.
Histograms

To love…

…and to hate
Thresholding and illumination

- Solutions:
  - Calibration of the imaging system
  - Percentile filter with very large mask
  - Morphological operators
MR non-uniformity

- median filtering

- thresholding
More thresholding

- Can also be used on other kinds of histogram: grey-level, edge, feature etc.

Multivariate data (⇒ see next lectures)

- Problems:
  - Only considers the graylevel pixel value, so it can leave “holes” in segmented objects.
    - Solution: post-processing with morphological operators
    - Requires strong assumptions to be efficient
    - Local thresholding is better ⇒ see region growing techniques