Tracking & Cross-correlation

CS 510
Lecture #11
February 15, 2017

PA2

• Any questions?
  – Yes, its due Monday.

Where are we?

We are building a vision system to describe events in videos

In PA2, we are building an attention mechanism to extract potentially interesting still objects in images

For PA3, you will build an attention mechanism that extracts moving objects from images

Moving Object Detection

• Assuming a still camera
• Two algorithms:
  – Mixture of Gaussians (Stauffer & Grimson)
  – ViBE (Barnich & Van Droogenbroeck)
• But these algorithms only label pixels
  – Need to group foreground pixels into regions
  – Need to connect regions across images
  – To reason about moving objects

Connected Components

• Goal: connect pixels into regions

Connected Components (II)

• Recursively/iteratively connect neighboring foreground pixels
  – 4 connected: NESW pixels are neighbors
  – 8 connected: Diagonals are also neighbors
Connected Components Implementation Notes

- Recursive definition is simple to write
  - May "blow the stack"
- Iterative definition is faster, safer
- Connected Components is already implemented in OpenCV

Pixel Grouping

- Connected components to group pixels
- Put a bounding box around component
- May choose to add additional requirements
  - Minimum size
  - Minimum width/height
  - Merge regions that are "close enough"

Morphology

- **Dilate** grows a region
  - An element is a small mask (e.g. a 3x3 square)
  - The element is put at every possible location of the image
  - If any 1 in the mask covers a 1 in the image, then all 1 mask positions are set to 1 in the image

Morphology (II)

- **Erode** is the opposite of open
  - If a mask 1 overlaps an image 0, set all image pixels under mask 1s to zero

Morphology (III)

- **Open** is an erosion followed by a dilation
- **Close** is a dilation followed by an erosion
- Example of open

Morphology – A Personal Opinion

- I am not fond of morphology
  - Computer vision "hacks"
- But you should know what it is
- If you want to play with it, erode, dilate, open and close are all defined in OpenCV
After Pixel Grouping

- Once you have found the pixel groups in an image...
- ... How do you connect objects across frames?
- It called tracking, but first another of my diversions...

Pearson’s Correlation

\[
\frac{\sum_{x,y} (A(x,y) - \bar{A})(B(x,y) - \bar{B})}{\sqrt{\sum_{x,y} (A(x,y) - \bar{A})^2 \sum_{x,y} (B(x,y) - \bar{B})^2}}
\]

This is a very important equation...

Assumptions of Correlation

\[
\frac{\sum_{x,y} (A(x,y) - \bar{A})(B(x,y) - \bar{B})}{\sqrt{\sum_{x,y} (A(x,y) - \bar{A})^2 \sum_{x,y} (B(x,y) - \bar{B})^2}}
\]

- Two signals vary linearly
  - Constant shift to either signal has no effect.
  - Increased amplitude has no effect.
- This minimizes sensitivity to:
  - changes in (overall) illumination
  - offset or gain.

Special Cases

- Any two linear functions with positive slope have correlation 1.
  - Only the sign of the slope matters.
- Any two linear functions with differently signed slopes have correlation -1.
  - This is called anti-correlation
  - Anti-correlation = correlation for prediction.
  - For matching, it may or may not be as good...
- Correlation undefined for slope = 0 (σ=0)

Correlation (cont.)

- Correlation is sensitive to:
  - Translation
  - Rotation: in-plane and out-of-plane
  - Scale
- Because it …
  - Assumes pixels align one atop the other.
  - Compares two images pixel by pixel.
- Translation handled by convolution
  - Example, alignment by template matching

Computing Correlation

- A constant added to A does not change its correlation to any other signal, so
  - Let’s subtract average A from A()
  - Let’s subtract average B from B()
  - The mean of both signals is now zero
  - Then correlation reduces to:

\[
\frac{\sum_{x,y} (A(x,y) - \bar{A})(B(x,y) - \bar{B})}{\sqrt{\sum_{x,y} (A(x,y) - \bar{A})^2 \sum_{x,y} (B(x,y) - \bar{B})^2}}
\]

Know and love the dot product.
Computing Correlation (II)

• For zero-mean signals, we can scale them without changing their correlation scores
  – Multiply A by the inverse of its length
  – Multiply B by the inverse of its length
  – Both signals are now unit length
  – Then correlation reduces to:

\[ A \cdot B \]

• Gives rise to ‘Correlation Space’.

Correlation Space

• Why zero-mean & unit-length your images?
• Consider, for example, database retrieval
  – Compare new image A …
  – with many images in database.
  – When database images are stored in their zero-mean & unit-length form, then
  – Preprocess A (zero-mean, unit-length)
  – Compute dot products

Know and love the dot product. 😊

Correlation Space (II)

• New idea: image as a point in an N dimensional space
  • N = width x height
• Zero-mean & unit-length images lie on an N-1 dimensional “correlation space” where the dot product equals correlation.
  – This is a highly non-linear projection.
  – Points lie on an N-1 surface within the original N dimensional space.
• So consider points in 3-D ….

Useful?

• Yes
• Very commonly used.
• For example
  – Face Rec.
• Google
  – 279,000 hits

Correlation Space (II)

• Subtracting mean - translation.
• Length one - project onto sphere.
• Correlation is then:
  – Cosine of angle between vectors (points).

Useful Connection …

• Euclidean distance is inversely proportional to correlation in correlation space.

\[
\sqrt{\sum_{x,y} [a(x,y) - b(x,y)]^2} = \sqrt{\sum_{x,y} a(x,y)^2 + \sum_{x,y} b(x,y)^2 - 2 \sum_{x,y} a(x,y)b(x,y)}
\]

\[
= \sqrt{1 + \sum_{x,y} a(x,y)^2 - 2 \sum_{x,y} a(x,y)b(x,y) + 1 - \sum_{x,y} b(x,y)^2}
\]

\[
= \sqrt{2 - 2A \cdot B} = \sqrt{2 - 2\text{Corr}(A,B)}
\]

• Nearest-neighbor classifiers in correlation space maximize correlation / minimized L2 norm
Limitations

• To match images this way, they must be
  – The same width & height
  – In correspondence: coordinates match
• More importantly, objects in the scene must
  – Be in the same location
  – Be at the same scale
  – Be at the same orientation
  – Be seen from the same viewpoint