Bag of Words

• Goal: Build a Bag of Words system
  – That measures similarity between videos
  – By two weeks from Friday (April 25th)
• Components: Training
  – Interest point extractor
  – Interest point descriptor
  – Clustering algorithm to produce codes
• Components: Run time
  – Interest point extractor
  – Interest point descriptor
  – Match descriptors to codes
  – Compare histograms of codes
Interest Points – Other Methods

• We know SIFT
• How else can we extract interest points?
  – Hessian (SURF) : similar in spirit to SIFT
  – MSER : based on “stable regions”
  – Fixed Grid or Random : does it really matter?
• Interest points tailored for video
  – Spatio-temporal interest points (STIP)
    • Laptev’s approach
    • Dollár’s approach
MSER (Matas et al, 2002)

• DoGs find roughly circular blobs
  – On center/off-surround or vice-versa
  – At a given scale (and we search scale space)

• Why does it have to be a circular blob?

• Why not look for “interesting” regions
  – Of any shape?
  – Of any size?
  – True affine invariance!
Extremal Regions

• Extremal regions are an example of image segmentation
  – Dividing image pixels into tiles

<table>
<thead>
<tr>
<th>05</th>
<th>03</th>
<th>04</th>
<th>09</th>
<th>09</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>03</td>
<td>05</td>
<td>09</td>
<td>11</td>
</tr>
<tr>
<td>02</td>
<td>03</td>
<td>09</td>
<td>11</td>
<td>13</td>
</tr>
<tr>
<td>39</td>
<td>37</td>
<td>40</td>
<td>44</td>
<td>45</td>
</tr>
<tr>
<td>40</td>
<td>50</td>
<td>42</td>
<td>43</td>
<td>46</td>
</tr>
</tbody>
</table>
Extremal Regions (II)

• Initially segmentation: threshold at lowest possible value

<table>
<thead>
<tr>
<th></th>
<th>05</th>
<th>03</th>
<th>04</th>
<th>09</th>
<th>09</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>03</td>
<td>05</td>
<td>09</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>02</td>
<td>03</td>
<td>09</td>
<td>11</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>37</td>
<td>40</td>
<td>44</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>50</td>
<td>42</td>
<td>43</td>
<td>46</td>
<td></td>
</tr>
</tbody>
</table>

Red pixels form foreground region (size = 1); all other pixels are regions of size 1
Extremal Regions (III)

• Now increase the threshold by 1

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>05</td>
<td>03</td>
<td>04</td>
<td>09</td>
<td>09</td>
</tr>
<tr>
<td>00</td>
<td>03</td>
<td>05</td>
<td>09</td>
<td>11</td>
</tr>
<tr>
<td>02</td>
<td>03</td>
<td>09</td>
<td>11</td>
<td>13</td>
</tr>
<tr>
<td>39</td>
<td>37</td>
<td>40</td>
<td>44</td>
<td>45</td>
</tr>
<tr>
<td>40</td>
<td>50</td>
<td>42</td>
<td>43</td>
<td>46</td>
</tr>
</tbody>
</table>

No change
Extremal Regions (IV)

- Now increase the threshold again

<table>
<thead>
<tr>
<th>05</th>
<th>03</th>
<th>04</th>
<th>09</th>
<th>09</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>03</td>
<td>05</td>
<td>09</td>
<td>11</td>
</tr>
<tr>
<td>02</td>
<td>03</td>
<td>09</td>
<td>11</td>
<td>13</td>
</tr>
<tr>
<td>39</td>
<td>37</td>
<td>40</td>
<td>44</td>
<td>45</td>
</tr>
<tr>
<td>40</td>
<td>50</td>
<td>42</td>
<td>43</td>
<td>46</td>
</tr>
</tbody>
</table>

Foreground region is now size 2
Extremal Regions (V)

• And again…

```
  05  03  04  09  09
  00  03  05  09  11
  02  03  09  11  13
  39  37  40  44  45
  40  50  42  43  46
```

Foreground region is now size 5
MSER Algorithm

- Run threshold from 0 to 255
- For every value, record size of every foreground region
  - In general, there will be multiple disconnected foreground regions
  - When two regions merge (on threshold increase), the smaller goes away & the bigger grows
- Every region now has a history of sizes
- The “maximally stable” one is the one that minimizes:

\[ q(i) = \frac{|Q_{i+\Delta} \setminus Q_{i-\Delta}|}{|Q_i|} \]
How about Random?

• Marée, et al 2005 did Bag of Words object recognition using random interest points
  – Picked \((x, y)\) positions at random
  – Picked scales at random (by distribution)
• Performed almost as well as SIFT
  – Perhaps all we need is a sufficient sampling?
Uniformly Sampled IPs

• Lazebnik et al 2006 created a fixed spatial pyramid of interest points
  – Small scale IPs placed every 8x8 pixels
  – Each descriptor covers a 16x16 patch
  – Down sample the image (using a pyramid) and repeat until the whole image is one 16x16 patch

• Outperforms other methods
• Now standard for still images (but not video)
Spatio-Temporal Interest Points

• So far, we have extracted IPs from images
• What about video data?
  – Extract IPs from every frame (expensive)
  – Subsample frames and extract IPs
• But isn’t there information in the time dimension?
• Two approaches:
  – \((x, y, t)\) corners (Laptev)
  – Spatial corners with frequencies (Dollar)
Video as 3D Data Cube

A video is a 3D data cube with dimensions x, y and time (t)

Source: NASA

http://breckon.eu/toby/demos/videovolumes/
3D Convolution

- 2D convolution is the dot product with a rectangle at every location in an image.
- 3D convolution is the dot product with a 3D volume at every location in a video cube.

3D Corners
(Laptev & Lindenberg 2003)

- We already covered 2D corners as IPs
  - Compute $dl/dx$ and $dl/dy$
  - Compute covariance matrix (structure tensor)
  - If the smaller eigenvalue is large, it’s a corner

- We can do the same thing in x, y and time
  - Compute $dl/dx$, $dl/dy$ and $dl/dt$ (3D convolution)
  - Compute 3x3 covariance matrix
  - If the smallest eigenvalue is large, it’s a corner
Laptev’s STIPs

Of course, these change with time…

http://www.di.ens.fr/~laptev/download.html

Not in OpenCV, but can be downloaded from:
http://www.di.ens.fr/~laptev/download.html
An alternative: Dollár’s STIPs
Dollár, et al 2005

- Idea: Find local frequency peaks
- For every pixel, compute

\[ R = \left( I \ast g \ast h_{ev} \right)^2 + \left( I \ast g \ast h_{odd} \right)^2 \]

- Where \( g \) is a 1D Gaussian applied in time
- \( h_{ev} \) is a cosine wave
- \( h_{odd} \) is a sine wave
- Local optima of \( R \) are interest points
Dollár’s STIPS

Of course, these change with time...

http://iselab.cvc.uab.es/silverage.php?q=SelectiveSTIPs

Not in OpenCV