Remember eigenfaces?

- They don’t work very well for detection
Issues: speed, features

• Case study: Viola Jones face detector
• Exploits two key strategies:
  – simple, super-efficient, but useful features
  – pruning (cascaded classifiers)

• Next few slides adapted Grauman & Liebe’s tutorial
  – [Link](http://www.vision.ee.ethz.ch/~bleibe/teaching/tutorial-aaai08/)
• Also see Paul Viola’s talk (video)
  – [Link](http://www.cs.washington.edu/education/courses/577/04sp/contents.html#DM)

Feature extraction

“Rectangular” filters

Feature output is difference between adjacent regions

Efficiently computable with integral image: any sum can be computed in constant time

Avoid scaling images → scale features directly for same cost

Viola & Jones, CVPR 2001
### Integral Image

<table>
<thead>
<tr>
<th>34</th>
<th>11</th>
<th>33</th>
<th>3</th>
<th>19</th>
<th>19</th>
<th>18</th>
<th>18</th>
<th>2</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>34</td>
<td>8</td>
<td>36</td>
<td>11</td>
<td>5</td>
<td>11</td>
<td>5</td>
<td>6</td>
<td>15</td>
<td>33</td>
</tr>
<tr>
<td>17</td>
<td>22</td>
<td>17</td>
<td>4</td>
<td>6</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>18</td>
<td>28</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>15</td>
<td>22</td>
<td>5</td>
<td>1</td>
<td>20</td>
<td>10</td>
<td>12</td>
<td>22</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>1</td>
<td>22</td>
<td>19</td>
<td>29</td>
<td>6</td>
<td>20</td>
<td>9</td>
<td>27</td>
</tr>
<tr>
<td>16</td>
<td>7</td>
<td>11</td>
<td>17</td>
<td>15</td>
<td>2</td>
<td>25</td>
<td>19</td>
<td>29</td>
<td>10</td>
</tr>
<tr>
<td>34</td>
<td>26</td>
<td>29</td>
<td>31</td>
<td>5</td>
<td>6</td>
<td>30</td>
<td>17</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>33</td>
<td>28</td>
<td>30</td>
<td>4</td>
<td>28</td>
<td>21</td>
<td>26</td>
<td>5</td>
<td>32</td>
<td>21</td>
</tr>
<tr>
<td>1</td>
<td>18</td>
<td>13</td>
<td>5</td>
<td>27</td>
<td>16</td>
<td>28</td>
<td>19</td>
<td>32</td>
<td>23</td>
</tr>
<tr>
<td>12</td>
<td>13</td>
<td>23</td>
<td>13</td>
<td>7</td>
<td>21</td>
<td>5</td>
<td>2</td>
<td>15</td>
<td></td>
</tr>
</tbody>
</table>

**Original Image**

**IIR Filter**

**Weights on Integral Image**

**Weight on Original Image**

*Slide courtesy of Andrew Gallagher*
Integral Image

Original Image

Integral Image

Original Image

Integral Image

Original Image

Integral Image

Original Image

Integral Image

Original Image

Integral Image

Original Image

Integral Image

Original Image

Integral Image

Original Image

Integral Image

Original Image

Integral Image

Original Image

Integral Image

Original Image

Integral Image

Original Image

Integral Image

Original Image
Integral Image

Original Image

Integral Image

IIR Filter

Weight on Original Image

Weights on Integral Image

Integral Image

Weights on Original Image

IIR Filter

Weight on Original Image

Slide courtesy of Andrew Gallagher
Integral Image

Original Image

Integral Image

Weights on Integral Image

Weight on Original Image

IIR Filter

O(N) Operations!

~400 in this case

Slide courtesy of Andrew Gallagher
Integral Image

\[
\begin{array}{cccccccc}
34 & 11 & 33 & 3 & 19 & 19 & 18 & 18 & 2 \ 44 \\
34 & 8 & 36 & 11 & 5 & 11 & 5 & 6 & 15 \ 33 \\
17 & 22 & 17 & 4 & 6 & 3 & 5 & 7 & 35 \\
11 & 22 & 22 & 5 & 20 & 10 & 12 & 22 \\
8 & 7 & 1 & 22 & 19 & 29 & 6 & 20 & 9 \ 27 \\
16 & 7 & 11 & 17 & 15 & 2 & 25 & 19 & 29 \ 10 \\
34 & 26 & 29 & 31 & 5 & 6 & 30 & 17 & 4 \ 10 \\
33 & 28 & 30 & 4 & 28 & 21 & 26 & 5 & 32 \ 21 \\
1 & 18 & 13 & 5 & 27 & 16 & 28 & 19 & 32 \ 23 \\
12 & 13 & 16 & 23 & 13 & 7 & 21 & 5 & 2 \ 15 \\
\end{array}
\]

\[
\begin{array}{cccc}
\sum & 69 & 6 & 6 & \\
\text{Integral} & 69 & 4 & 4 & \\
\end{array}
\]
### Integral Image

<table>
<thead>
<tr>
<th>34</th>
<th>11</th>
<th>33</th>
<th>3</th>
<th>19</th>
<th>19</th>
<th>18</th>
<th>18</th>
<th>2</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>34</td>
<td>8</td>
<td>36</td>
<td>11</td>
<td>5</td>
<td>11</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>33</td>
</tr>
<tr>
<td>17</td>
<td>22</td>
<td>17</td>
<td>4</td>
<td>6</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>35</td>
<td>18</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>15</td>
<td>22</td>
<td>5</td>
<td>1</td>
<td>20</td>
<td>10</td>
<td>12</td>
<td>22</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>1</td>
<td>22</td>
<td>19</td>
<td>29</td>
<td>6</td>
<td>20</td>
<td>9</td>
<td>27</td>
</tr>
<tr>
<td>16</td>
<td>7</td>
<td>11</td>
<td>17</td>
<td>15</td>
<td>2</td>
<td>25</td>
<td>19</td>
<td>29</td>
<td>10</td>
</tr>
<tr>
<td>34</td>
<td>26</td>
<td>29</td>
<td>31</td>
<td>5</td>
<td>6</td>
<td>30</td>
<td>17</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>33</td>
<td>28</td>
<td>30</td>
<td>4</td>
<td>28</td>
<td>21</td>
<td>26</td>
<td>5</td>
<td>32</td>
<td>21</td>
</tr>
<tr>
<td>1</td>
<td>18</td>
<td>13</td>
<td>5</td>
<td>27</td>
<td>16</td>
<td>28</td>
<td>19</td>
<td>32</td>
<td>23</td>
</tr>
<tr>
<td>12</td>
<td>13</td>
<td>16</td>
<td>23</td>
<td>13</td>
<td>7</td>
<td>21</td>
<td>5</td>
<td>2</td>
<td>15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sum</th>
<th>Cost</th>
<th>Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>141</td>
<td>9</td>
<td>39</td>
</tr>
</tbody>
</table>

**Integral**

| 141 | 4    | 12         |

| Integral way: | 762-621=141 |

Slide courtesy of Andrew Gallagher

---

### Large library of filters

Considering all possible filter parameters: position, scale, and type:

- 180,000+ possible features associated with each 24 x 24 window

Use AdaBoost both to select the informative features and to form the classifier

Viola & Jones, CVPR 2001

K. Grauman, B. Leibe
**AdaBoost for feature+classifier selection**

- Want to select the single rectangle feature and threshold that best separates *positive* (faces) and *negative* (non-faces) training examples, in terms of *weighted* error.

\[
h_t(x) = \begin{cases} 
+1 & \text{if } f_t(x) > \theta_t \\
-1 & \text{otherwise}
\end{cases}
\]

Resulting weak classifier:

For next round, reweight the examples according to errors, choose another filter/threshold combo.

**Viola & Jones, CVPR 2001**

---

**AdaBoost: Intuition**

Consider a 2-d feature space with *positive* and *negative* examples.

Each weak classifier splits the training examples with at least 50% accuracy.

Examples misclassified by a previous weak learner are given more emphasis at future rounds.

Figure adapted from Freund and Schapire
AdaBoost: Intuition

Final classifier is combination of the weak classifiers
• Given example images \((x_1, y_1), \ldots, (x_n, y_n)\) where \(y_i = 0, 1\) for negative and positive examples respectively.
• Initialize weights \(w_{t,i} = \frac{1}{n}\) for \(y_i = 0, 1\) respectively, where \(n\) and \(T\) are the number of negatives and positives respectively.
• For \(t = 1, \ldots, T\):
  1. Normalize the weights,
     \[
     w_{t,i}^{new} = \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}
     \]
     so that \(w_t\) is a probability distribution.
  2. For each feature, \(j\), train a classifier \(h_j\) which is restricted to using a single feature. The error is evaluated with respect to \(w_t\), \(e_t = \sum_i w_t |h_j(x_i) - y_i|\).
  3. Choose the classifier, \(h_t\), with the lowest error \(e_t\).
  4. Update the weights:
     \[
     w_{t+1,i} = w_t \cdot \left(\frac{1}{1 + \frac{e_t}{e_{best}}}\right)^{e_t}
     \]
     where \(e_t = 0\) if example \(x_i\) is classified correctly, \(e_t = 1\) otherwise, and \(h_t = \frac{1}{\sum_{j=1}^{T} e_j}\).
• The final strong classifier is:
  \[
  h(x) = \begin{cases} 
    1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq T/2 \\
    0 & \text{otherwise}
  \end{cases}
  \]
  where \(\alpha_t = \log \frac{1}{T_{t}}\)

**AdaBoost Algorithm**

Start with uniform weights on training examples

For \(T\) rounds

- Evaluate weighted error for each feature, pick best.

Re-weight the examples:
- Incorrectly classified -> more weight
- Correctly classified -> less weight

Final classifier is combination of the weak ones, weighted according to error they had.

Fleuret & Geman, IJCV 2001
Rowley et al., PAMI 1998
Viola & Jones, CVPR 2001

**Cascading classifiers for detection**

For efficiency, apply less accurate but faster classifiers first to immediately discard windows that clearly appear to be negative; e.g.,

- Filter for promising regions with an initial inexpensive classifier
- Build a chain of classifiers, choosing cheap ones with low false negative rates early in the chain

Fleuret & Geman, IJCV 2001
Rowley et al., PAMI 1998
Viola & Jones, CVPR 2001

Figure from Viola & Jones CVPR 2001
Viola-Jones Face Detector: Summary

- Train with 5K positives, 350M negatives
- Real-time detector using 38 layer cascade
- 6061 features in total throughout layers
- [Implementation available in OpenCV: http://www.intel.com/technology/computing/opencv/]

K. Grauman, B. Leibe

Viola-Jones Face Detector: Results

First two features selected
Viola-Jones Face Detector: Results

K. Grauman, B. Leibe
Viola-Jones Face Detector: Results

Detecting profile faces?

Detecting profile faces requires training separate detector with profile examples.
Viola-Jones Face Detector: Results

Questions?