On-Line Selection of Discriminative Tracking Features

Robert Collins and Yanxi Liu
(and later, Marius Leordeanu)

ICCV 2003
Classification-based Tracking

training frame

foreground
background

Classifier

train a classifier

Classifier

test frame

label pixels
Observation

Tracking success/failure is highly correlated with our ability to distinguish object appearance from background.

Suggestion:

Explicitly seek features that best discriminate between object and background samples.
Continuously adapt feature used to deal with changing background, changes in object appearance, and changes in lighting conditions.
Why Do Adaptive Feature Selection

The best feature to use changes over time, due to varying illumination and background characteristics encountered while tracking.
Selecting Good Features for Tracking

Motivation: real-time, adaptive feature selection for better distinguishing target from background while tracking.

Approach: use a computationally simple method for computing “goodness” of each candidate feature so we can rank order them.

“Goodness” → discrimination between foreground/background
Feature Selection Prior Work

Feature Selection: choose M features from N candidates (M << N)

Traditional Feature Selection Strategies
  • Forward Selection
  • Backward Selection
  • Branch and Bound

Viola and Jones, Cascaded Feature Selection for Classification

Bottom Line: slow, off-line process
Prior Work


Choose between five 2D color spaces

Flesh probability image $P_i$ formed by histogram backprojection using color space $I$

Sample face and background windows selected

Color space $i$ evaluated using

$$\frac{\sum_{\text{face}} P_i^2}{\|\text{face}\| \sum_{\text{background}} P_i}$$
Variance Ratio

We want a computationally simple method for computing “goodness” of a feature space candidate so we can rank order them.

\[
\frac{\text{Var between classes}}{\text{Var within classes}} = \frac{\text{Var ([class1, class2])}}{\text{Var(class1)+Var(class2)}}
\]

Idea is that we want features that map object and background pixels to map to uniform values that are far apart.

in mathspeak: minimize within class variance while maximizing cross-class variance

notice the center-surround concept appearing again!
Computing Variance Ratio

From training samples, compute the following

\[ P(x|\text{obj}) = p(x) = \frac{H_{\text{obj}}(f(x))}{n_{\text{obj}}} \]

\[ P(x|\text{bg}) = q(x) = \frac{H_{\text{bg}}(f(x))}{n_{\text{bg}}} \]

Note, these are empirical class-conditional distributions for a specific feature \( f \)

They are computed as normalized histograms
Computing Variance Ratio

variance of feature values on $p=\{\text{class1, class2}\}$

$$
\text{var}(L; p) = E[L^2(i)] - (E[L(i)])^2
$$

$$
= \sum_i p(i)L^2(i) - [\sum_i p(i)L(i)]^2
$$

ratio of total variance (class1+class2 samples) to sum of variances of single class samples

$$
\text{VR}(L; p, q) \equiv \frac{\text{var}(L; (p+q)/2)}{[\text{var}(L; p) + \text{var}(L, q)]}
$$
Intuition Behind Variance Ratio

- Little overlap in the discriminant function
  - High between-group variance relative to within-group variance
- Greater overlap in the discriminant function
  - Low between-group variance relative to within-group variance
Related to Fisher Discriminant Function

File: noname.mat, # of points $K = 26$

$$J_F(w) = \frac{(m_1' - m_2')^2}{D_1^2 + D_2^2}$$

where $m_i = \frac{1}{N_i} \sum_{y \in C'_i} y$ and $D_i^2 = \sum_{y \in C'_i} (y - m_i')^2$. 
Limitation of Variance Ratio

Implicit assumption that distribution of both classes are unimodal (it is ideal when both are Gaussians).

We will do a nonlinear transformation of each “raw” feature such that it results in more unimodal values for object and background pixels.
Motivation for Tuned Features

Note: this example explains why we don’t just use LDA
Computing Tuned Features

empirical probability distributions

\[ P(x|\text{obj}) = p(x) = \frac{H_{\text{obj}}(f(x))}{n_{\text{obj}}} \]

\[ P(x|\text{bg}) = q(x) = \frac{H_{\text{bg}}(f(x))}{n_{\text{bg}}} \]

weight image for tracking

\[ L(x) = \log \frac{p(x)}{q(x)} \]

log likelihood ratio

\[ L(i) = \log \frac{\max \{p(i), \delta\}}{\max \{q(i), \delta\}} \]

implementation detail (avoid log(0) or xx/0)
Example: 1D Color Feature Spaces

Color features: integer linear combinations of R,G,B

\[ \frac{(a \ R + b \ G + c \ B)}{(|a|+|b|+|c|)} + \text{offset} \]

where \(a,b,c\) are \{-2,-1,0,1,2\} and offset is chosen to bring result back to 0,…,255.

Note: this includes some common simple feature combinations

- \(R+G+B\) (intensity)
- \(2G-R-B\) (excess green)
- \(R-B\) (opponent colors)

Barring algebraically redundant features, we have 49 candidates
The 49 color feature candidates roughly uniformly sample the space of 1D marginal distributions of RGB.
Example

training frame          test frame

foreground

background

sorted variance ratio
Example: Feature Ranking
## More Sample Feature Rankings

<table>
<thead>
<tr>
<th>Object/background designation</th>
<th>Likelihood from most discriminative feature</th>
<th>Likelihood from least discriminative feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-2G+2B</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
</tr>
<tr>
<td>R+2G+2B</td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
<tr>
<td>-R-2G+2B</td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
<tr>
<td>-R+2G</td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
</tr>
<tr>
<td>R</td>
<td><img src="image9.png" alt="Image" /></td>
<td><img src="image10.png" alt="Image" /></td>
</tr>
<tr>
<td>R+2G-B</td>
<td><img src="image11.png" alt="Image" /></td>
<td><img src="image12.png" alt="Image" /></td>
</tr>
<tr>
<td>-R+2G-B</td>
<td><img src="image13.png" alt="Image" /></td>
<td><img src="image14.png" alt="Image" /></td>
</tr>
<tr>
<td>-R+2G</td>
<td><img src="image15.png" alt="Image" /></td>
<td><img src="image16.png" alt="Image" /></td>
</tr>
<tr>
<td>2R-B</td>
<td><img src="image17.png" alt="Image" /></td>
<td><img src="image18.png" alt="Image" /></td>
</tr>
</tbody>
</table>
Overview of Tracking Algorithm

Note: since log likelihood images contain negative values, must use modified mean-shift algorithm as described in Collins, CVPR’03
Avoiding Model Drift

Problem: Adaptive appearance models have a tendency to “drift”
background pixels mistakenly incorporated into the object model pull the model off the correct location, leading to more misclassified background pixels, and so on.

Our solution: force foreground object distribution to be a combination of current appearance and original appearance (anchor distribution)

\[
\text{anchor distribution} = \text{object appearance histogram from first frame} \\
\text{object distribution} = (\text{current distribution} + \text{anchor distribution}) / 2
\]

Note: this limits the ability to drift, but also limits the ability of the appearance model to adapt to large color changes
Examples

We show samples of tracking, demonstrating the following benefits of this approach:

• Enhancing ability to track low contrast objects
• Adapting to changing illumination/backgrounds
• Minimizing the effects of background distractions
Benefits: Tracking Hard-to-See Objects

Trace of selected features
Benefits: Adapting to Changing Illumination / Background

Trace of selected features
Benefits: Minimizing Distractions

For multi-color objects, the algorithm can avoid distractors by automatically adjusting color emphasis.
Benefits: Minimizing Distractions

Avoiding distractors by adjusting color emphasis
More Detail

top 3 weight (log likelihood) images
More Detail

top 3 weight (log likelihood) images
A Tracking Failure

[Image of a busy street with cars and a black rectangle highlighting a specific area.]

[Additional images showing different views of the same scene with labels R+G, R+2G, and 2R+2G+D.]
Problem with Variance Ratio

Although variance ratio does well at picking features that make the object appear distinctive from the overall background, it is unable to recognize spatially coherent clusters of high likelihood scores that represent potential distractors.

As a result, mean shift may jump to a nearby vehicle.
Problem with Variance Ratio

Variance ratio examines the overall distribution of likelihood scores within the object region and surrounding background.

Problem: this approach favors likelihood images with the object having high contrast with the average background score, even though there may be an equally high contrast distractor in the surrounding background region.

We would prefer this image for tracking.

Weaker object response but NO distractors.
Motivation: Distractor-Resistance

Figure 10: Left: car passing another car of a similar color. Middle: weight image chosen by the variance ratio. Right: weight image chosen by the method developed in this section. The variance ratio favors features that produce weight images where the object has high contrast with respect to the average background, even though there may be an equally high contrast distractor nearby (middle image). We prefer the weight image at the far right for tracking – despite the poor contrast between object and background, there are no nearby distractors to tempt the tracker.
We have found an improvement to using variance ratio score for selecting features that minimize potential distractors and clutter.
Distractor-Resistent Selection

Solution: to avoid distractors, we must do better spatial reasoning about peaks in the location likelihood image.

Our approach:

1) Given a candidate feature likelihood surface, smooth it with a Gaussian kernel related to the scale of the mean-shift window (the result represents the actual surface that the mean-shift algorithm performs hill-climbing on).  [we’ll see another reason why later]

2) Extract the central object peak

3) Find the next highest peak – this represents the most likely distractor

4) Measure feature “goodness” as a function of these two peak heights (ratio or difference, depending on whether we are using a plain or log likelihood image)

Note, this is related to the notion of “the margin” in a traditional classifier sense (in this case foreground/background classifier). We are thus choosing the feature that maximizes this margin, therefore also minimizing the probability of making a classification error.
Example

More Examples
Peak Difference Computation

- Weight Image (log likelihood)
- Primary Peak Identified
- Object and Maximum Distractor Identified
- Prior Shape Estimate
- Weight Image
- Secondary Peak Identified
- Peak Diff Score
Probabilistic Explanation

empirical probability distributions

\[ P(x|obj) = p(x) = \frac{H_{obj}(f(x))}{n_{obj}} \]

\[ P(x|bg) = q(x) = \frac{H_{bg}(f(x))}{n_{bg}} \]

weight image for tracking

\[ L(x) = \log \frac{p(x)}{q(x)} \] log likelihood ratio

\[ L(i) = \log \frac{\max \{p(i), \delta\}}{\max \{q(i), \delta\}} \] implementation detail
( avoid log(0) or xx/0 )
Probabilistic Explanation

$X_0$ region where we know object is (cause we tracked it here)

$X_1$ region with strongest secondary peak (this could be confused as object in next frame)

Let $c_0 =$ class label our algorithm will give pixels in $X_0$
$c_1 =$ class label our algorithm will give pixels in $X_1$

Consider likelihood of two “events”
\[ A \equiv \{c_0 = \text{obj}, c_1 = \text{bg}\} \text{ and } B \equiv \{c_0 = \text{bg}, c_1 = \text{obj}\}. \]
Probabilistic Explanation

Consider likelihood of two “events”

\[ A \equiv \{ c_0 = \text{obj}, c_1 = \text{bg} \} \text{ and } B \equiv \{ c_0 = \text{bg}, c_1 = \text{obj} \}. \]

We would like to maximize the likelihood of event A and minimize the likelihood of event B.

We therefore want to maximize

\[
\frac{P(A|X_0, X_1)}{P(B|X_0, X_1)} = \frac{P(c_0 = \text{obj}, c_1 = \text{bg}|X_0, X_1)}{P(c_0 = \text{bg}, c_1 = \text{obj}|X_0, X_1)}
\]
Probabilistic Explanation

now crank through the math

\[
P(c_0 = \text{obj}, c_1 = \text{bg} | X_0, X_1) \quad \frac{P(c_0 = \text{bg}, c_1 = \text{obj} | X_0, X_1)}{P(c_0 = \text{bg}, c_1 = \text{obj} | X_0, X_1)}
\]

\[\downarrow \text{apply Bayes rule, with priors denoted by } \pi\]

\[
= \frac{P(X_0, X_1 | c_0 = \text{obj}, c_1 = \text{bg}) \pi(c_0 = \text{obj}, c_1 = \text{bg})}{P(X_0, X_1 | c_0 = \text{bg}, c_1 = \text{obj}) \pi(c_0 = \text{bg}, c_1 = \text{obj})}
\]

\[\downarrow \text{class conditional independence; replace constant prior by } C\]

\[
= C \quad \frac{P(X_0 | c_0 = \text{obj})P(X_1 | c_1 = \text{bg})}{P(X_0 | c_0 = \text{bg})P(X_1 | c_1 = \text{obj})}
\]
Probabilistic Explanation

\[ C \frac{P(X_0|c_0 = \text{obj})P(X_1|c_1 = \text{bg})}{P(X_0|c_0 = \text{bg})P(X_1|c_1 = \text{obj})} \]

\[ \downarrow \text{independence over pixels in region} \]

\[ C \prod_{X_0} \frac{P(x|\text{obj})}{P(x|\text{bg})} \prod_{X_1} \frac{P(x|\text{bg})}{P(x|\text{obj})} \]

\[ \downarrow \text{substitute empirical distributions (computed from histograms)} \]

\[ C \prod_{X_0} \frac{p(x)}{q(x)} \prod_{X_1} \frac{q(x)}{p(x)} \]
Probabilistic Explanation

Dropping the constant prior term $C$, we maximize the log of the equation

\[
\log \left( \prod_{X_0} \frac{p(x)}{q(x)} \prod_{X_1} \frac{q(x)}{p(x)} \right) = \log \left( \prod_{X_0} \frac{p(x)}{q(x)} / \prod_{X_1} \frac{p(x)}{q(x)} \right)
\]

\[
= \sum_{X_0} \log \frac{p(x)}{q(x)} - \sum_{X_1} \log \frac{p(x)}{q(x)}
\]

\[
= \sum_{X_0} L(x) - \sum_{X_1} L(x)
\]

the gaussian smoothing approximates these sums peak difference score
Example

Distractor Example: Passing a Similar Car
Segmenting the Object

Observation: the likelihood image typically gives, visually, a reasonable description of the shape of the object.

Make this concrete by segmenting out a binary shape mask

Note: if we threshold likelihood at 0, we are actually making a color segmentation decision based on

\[ p(x|\text{object}) > p(x|\text{background}) \] for each pixel color x

This assumes that, within our window, object pixels and background pixels are equally likely to occur. If that isn’t true, we should modify our decision rule to choose

\[ p(x|\text{object})p(\text{object}) > p(x|\text{background})p(\text{background}) \]
Segmentation from Likelihood Image

- Region of interest
- Log likelihood
- Threshold at 0
- Intersect region of interest
Modify Algorithm to Add Shape

Based on likelihood image, and previous shape, segment object to create a binary shape mask.

Only sample object pixels from where bitmask has a 1, and background pixels where bitmask is 0. This should give “cleaner” statistics on object and background color distributions.

Also, when evaluating color features for feature selection, add a term that scores consistency of shape. Since object shape should change slowly over time, we don’t want to change to a new feature that “segments” our object very differently.
Shape Consistency

Shape consistency over time imposed in two ways.

1) when segmenting likelihood image, old shape mask is imposed to ensure that new shape mask cannot grow/shrink too much.

   ![Diagram showing shape consistency](image)
   - old shape
   - new shape boundary is constrained to lie within +/- epsilon of old boundary

2) when evaluating color features for feature selection, features are ranked-ordered by consistency of shape with old mask, using chamfer distance.
Shape Comparison

We have implemented shape comparison using based on chamfer distance.

Add up values distance values where mask=1

shape 1

Distance transform

shape 2

score
Comparison with Old Version

old version (no shape)  new version (incorporating shape)
Comparison continued

old version (no shape)  new version (incorporating shape)
A Failure of New Algorithm

Model drift is still an issue!

current work trying to address this problem

1) allow a parameterized model of shape (e.g. rectangle + bounded affine transforms) that defines the space of valid “expected shapes”
2) penalize deviation of proposed shape from an expected shape.
Summary

Features that best discriminate between foreground and background pixels are good features to use for tracking

• Variance ratio can be used as an efficient on-line feature selection method
• Log-likelihood ratio is used as a nonlinear feature mapping that turns potentially multi-modal object/background distributions into two uni-modal distributions

Contributions

• We introduce a framework for evaluating tracking features based on ability to discriminate foreground from background
• Continuous evaluation and adaptation of tracking features allows a tracker to handle changing background, changes in object appearance, and changes in lighting conditions.
Where to Go From Here

- Try random projection for feature selection, rather than exhaustive evaluation of all available features candidates.
- Need more accurate sampling from object/background distributions when rectangular windows don’t describe object shape well. Use oriented rectangles or ellipses?
- Zhaozheng Yin @PSU has a different approach to distractor-resistant tracking.