Graphical Object Models for Detection and Tracking

Leonid Sigal (ls@cs.brown.edu)
Department of Computer Science
Brown University

Joined work with:
- Ying Zhu, Siemens Corporate Research, Princeton, NJ
- Dorin Comaniciu, Siemens Corporation Research, Princeton, NJ
- Michael J. Black, Department of Computer Science, Brown University
Object Detection and Tracking

Vehicle

Pedestrian
Why Object Detection is Hard?

- Many target objects
- Appearance/lighting changes
- Partial occlusions
- Different orientations (articulations) of the object
- Different scale of objects
Why Object Detection is Hard?

- Many target objects
- Appearance/lighting changes
- Partial occlusions
- Different orientations (articulations) of the object
- Different scale of objects
Why Object Detection is Hard?

- Many target objects
- Appearance/lighting changes
- Partial occlusions
- Different orientations (articulations) of the object
- Different scale of objects
Why Object Detection is Hard?

- Many target objects
- Appearance/lighting changes
- Partial occlusions
- Different orientations (articulations) of the object
- Different scale of objects
Why Object Detection is Hard?

- Many target objects
- Appearance/lighting changes
- Partial occlusions
- Different orientations (articulations) of the object
- Different scale of objects
Object Detection

Is This a Vehicle?
Object Detection

Is This a Vehicle?
Object Detection

Is This a Vehicle?
Object Detection

Is This a Vehicle?
Object Detection: Machine Learning Approach

$$\begin{bmatrix}
  f_1 \\
  f_2 \\
  \vdots \\
  f_N
\end{bmatrix}$$

Classifier

Vehicle / Not-Vehicle
Object Detection: Using Pixel Values as Features

A Trainable Pedestrian Detection System (‘98)
Papageorgiou, Evgeniou, Poggio

\[ \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix} \]

SVM Classifier

\[ f_i \quad \text{Pixel value at location } i, \text{ where } i \text{ is in the patch (N~128x64=8192)} \]

- Many training examples to learn
- Requires many support vectors
Object Detection: Feature Selection + Classification

“Pedestrian” 
Class Examples

“Background” 
Class Examples

Learn Features to Use for Classification 
and the Classifier itself

Classifier
Object Detection: AdaBoost Approach

- Wavelet-like over-complete set of features, with simple weak classifiers $[-1,1]$

$\sim 40,000$ features to choose from
Object Detection:
AdaBoost Approach

Robust Real-time Object Detection ('01)
P. Viola and M. Jones

\[
\begin{bmatrix}
  h_1(f_1) \\
  h_2(f_2) \\
  \vdots \\
  h_N(f_N)
\end{bmatrix}
\]

\[
\sum_{i} \alpha_i = H(Y_i) > \frac{1}{2} \quad \text{Vehicle}
\]

\[
\leq \frac{1}{2} \quad \text{Not-Vehicle}
\]

\(N\) is much smaller than the number of pixels (~100)
Object Detection: AdaBoost Approach

- Tends to produce many false positives (need motion information Viola & Jones ’04)
- Does not explicitly model object parts, or their spatial relationship
Why parts are useful?

- Parts are easier to model
- Parts are robust to appearance changes (due to articulations and lighting)
- Parts can be reused
Part-Based Object Detection

- Example-Based Object Detection in Images by Components (’01)
  A. Mohan, C. Papageorgiou, T. Poggio

- Object Class Recognition by Unsupervised Scale-Invariant Learning (’03)
  R. Fergus, P. Perona, A. Zisserman

- A Bayesian Approach to Unsupervised One-Shot Learning of Object Categories (’03)
  L. Fei-Fei, R. Fergus, P. Perona

- Human detection based on a probabilistic assembly of robust part detectors (’04)
  K. Mikolajczyk, C. Schmid, A. Zisserman

- Unlike all previous methods
  - We use graphical model to represent an object, which results in elegant inference algorithm
  - We incorporate temporal constraints

- Supervised learning (unlike Fergus, Perona, Zisserman)
Object is represented as a 2-layer graphical model.
Each part of the object (and the object itself) is a node.
Spatial (and temporal) constraints are encoded using conditional distributions.
Graphical Object Models: Modeling Parts

- Each part/object has an associated AdaBoost detector

\[ X^i = [x, y, s]^T \]

- 3D parameter vector \((X^i)\) defining the position and the scale of the part/object in an image to be estimated
Graphical Object Models: Spatio-Temporal Extension

Spatial model can be extended in time
Graphical Object Models

- The joint distribution of the 2-layer spatio-temporal model can be written:

\[
P(X^O_0, X^C_0, X^C_1, \ldots, X^C_N, \ldots, X^O_T, X^C_T, X^C_1, \ldots, X^C_N, Y_i \ldots Y_T) = \\
\frac{1}{Z} \prod_{ij} \psi_{ij}(X^O_i, X^O_j) \prod_{ik} \psi_{ik}(X^O_i, X^C_k) \prod_{ikl} \psi_{kl}(X^C_k, X^C_l) \\
\prod_i \phi_i(X^O_i, Y_i) \prod_{ik} \phi_{ik}(X^C_k, Y_i)
\]
Graphical Object Models

The joint distribution of the 2-layer spatio-temporal model can be written:

State of object at time $T$

$$P(X^O_0, X^C_0, X^C_1, \ldots, X^C_N, \ldots, X^O_T, X^C_T, X^C_1, \ldots, X^C_N, Y_i \ldots Y_T) = \frac{1}{Z} \prod_{ij} \psi_{ij}(X^O_i, X^O_j) \prod_{ik} \psi_{ik}(X^O_i, X^C_k) \prod_{ikl} \psi_{kl}(X^C_k, X^C_l) \prod_i \phi_i(X^O_i, Y_i) \prod_{ik} \phi_{ik}(X^C_k, Y_i)$$
Graphical Object Models

The joint distribution of the 2-layer spatio-temporal model can be written:

\[
P(X_0^O, X_0^O, X_0^C, \ldots, X_N^C, \ldots, X_T^O, X_T^O, X_T^C, \ldots, X_T^C, Y_1 \ldots Y_T) =
\]

\[
\frac{1}{Z} \prod_{ij} \psi_{ij}(X_i^O, X_j^O) \prod_{ik} \psi_{ik}(X_i^O, X_i^C) \prod_{ikl} \psi_{kl}(X_i^C, X_i^C)
\]

\[
\prod_i \phi_i(X_i^O, Y_i) \prod_{ik} \phi_{ik}(X_i^C, Y_i)
\]
The joint distribution of the 2-layer spatio-temporal model can be written:

\[
P(X_0^O, X_0^C_0, X_0^C_1, \ldots, X_0^C_N, \ldots, X_T^O, X_T^C_0, X_T^C_1, \ldots, X_T^C_N, Y_i \ldots Y_T) = \frac{1}{Z} \prod_{ij} \psi_{ij}(X_i^O, X_j^O) \prod_{ik} \psi_{ik}(X_i^O, X_i^C_k) \prod_{ikl} \psi_{kl}(X_i^C_k, X_i^C_l) \prod_i \phi_i(X_i^O, Y_i) \prod_{ik} \phi_{ik}(X_i^C_k, Y_i)
\]
Graphical Object Models

- The joint distribution of the 2-layer spatio-temporal model can be written:

\[ P(X_0^O, X_0^{C_o}, X_0^{C_1}, \ldots, X_N^{C_N}, \ldots, X_T^O, X_T^{C_o}, X_T^{C_1}, \ldots, X_T^{C_N}, Y_1 \ldots Y_T) = \]

\[ \frac{1}{Z} \prod_{ij} \psi_{ij}(X_i^O, X_j^O) \prod_{ik} \psi_{ik}(X_i^O, X_k^{C_o}) \prod_{ikl} \psi_{kl}(X_k^{C_o}, X_l^{C_1}) \]

\[ \prod_i \phi_i(X_i^O, Y_i) \prod_{ik} \phi_{ik}(X_k^{C_o}, Y_i) \]
Graphical Object Models

- The joint distribution of the 2-layer spatio-temporal model can be written:

$$P(X^O_0, X^C_o, X^C_1, \ldots, X^C_N, \ldots, X^O_T, X^C_T, X^C_1, \ldots, X^C_N, Y_i \ldots Y_T) = \frac{1}{Z} \prod_{ij} \psi_{ij}(X^O_i, X^O_j) \prod_{ik} \psi_{ik}(X^O_i, X^C_k) \prod_{ikl} \psi_{kl}(X^C_k, X^C_l) \prod_i \phi_i(X^O_i, Y_i) \prod_{ik} \phi_{ik}(X^C_k, Y_i)$$

Spatial constraints between objects and its components
Graphical Object Models

The joint distribution of the 2-layer spatio-temporal model can be written:

\[
P(X_0^O, X_0^C, X_1^C, \ldots, X_N^C, \ldots, X_T^O, X_T^C, X_T^C, \ldots, X_T^C, Y, \ldots Y_T) =
\]

\[
\frac{1}{Z} \prod_{ij} \psi_{ij}(X_i^O, X_j^O) \prod_{ik} \psi_{ik}(X_i^O, X_i^C) \prod_{ikl} \psi_{kl}(X_i^C, X_i^C) \\
\prod_i \phi_i(X_i^O, Y_i) \prod_{ik} \phi_{ik}(X_i^C, Y_i)
\]

Spatial constraints between components of the objects
Graphical Object Models

The joint distribution of the 2-layer spatio-temporal model can be written:

\[
P(X_0^O, X_0^C, X_1^C, \ldots, X_N^C, \ldots, X_T^O, X_T^C, X_T^C, \ldots, X_T^C, Y_i \ldots Y_T) =
\]

\[
\frac{1}{Z} \prod_{ij} \psi_{ij}(X_i^O, X_j^O) \prod_{ik} \psi_{ik}(X_i^O, X_i^C) \prod_{ikl} \psi_{kl}(X_i^C, X_i^C) \prod_i \phi_i(X_i^O, Y_i) \prod_{ik} \phi_{ik}(X_i^C, Y_i)
\]

Evidence for the object
Graphical Object Models

The joint distribution of the 2-layer spatio-temporal model can be written:

\[
P(X_0^O, X_0^{C_o}, X_0^{C_1}, \ldots, X_N^{C_N}, \ldots, X_T^O, X_T^{C_o}, X_T^{C_1}, \ldots, X_T^{C_N}, Y_i \ldots Y_T) =
\]

\[
\frac{1}{Z} \prod_{ij} \psi_{ij}(X_i^O, X_j^O) \prod_{ik} \psi_{ik}(X_i^O, X_i^{C_k}) \prod_{ikl} \psi_{ikl}(X_i^{C_k}, X_i^{C_l})
\]

\[
\prod_i \phi_i(X_i^O, Y_i) \prod_{ik} \phi_{ik}(X_i^{C_k}, Y_i)
\]

Evidence for the each component of the object
Inference Algorithm

- Inference in such graphical models can be estimated using Belief Propagation.
- But, not when
  - State-space is continuous, and
  - Messages are not Gaussian.
- This forces the use of approximate inference algorithms (PAMPAS / Non-Parametric BP)
  - M. Isard (CVPR ’03)
  - E. Sudderth, A. Ihler, W. Freeman, A. Willsky (CVPR ’03)
Learning Temporal and Spatial Constraints

- Constraints (conditional distributions) are modeled using a Mixture of Gaussians with a single Gaussian outlier process.

\[
\psi_{ij}(X_j | X_i) = \lambda^0 N(\mu_{ij}, \Lambda_{ij}) + (1 - \lambda^0) \sum_{m=1}^{M_{ij}} q_{ijm} N(F_{ijm}(X_i), G_{ijm}(X_i))
\]

- Learned from the set of labeled patterns.
AdaBoost Image Likelihood

- Given a set of labeled patterns AdaBoost learns the weighted combination of base classifiers

\[ H(Y \mid X^i) = \sum_{k=1}^{K} \alpha_k h_k(Y \mid X^i) \]

- The final strong classifier gives the confidence that a patch of the image \( Y \) defined by the state \( X_i \) is of the desired class
AdaBoost Image Likelihood

- We can convert the confidence score $H(Y \mid X^i)$ into a likelihood by:

  $$\phi_i(Y \mid X^i) \propto \exp \left( \frac{H(Y \mid X^i)}{T} \right)$$

- $T$ is the “temperature” parameter that controls the smoothness of the likelihood function.

- Note, that the image likelihoods are assumed to be independent (not strictly so due to the possible overlap).
Non-Parametric Belief Propagation (PAMPAS)

- Represent messages and beliefs by a discrete set of weighted samples/kernels (i.e. Mixture of Gaussians)
Non-Parametric Belief Propagation (PAMPAS)

- Non-Parametric BP can be approximately solved using Monte Carlo integration
- For details, please see:

**Attractive people: Assembling loose-limbed models using non-parametric belief propagation (NIPS ’03)**
L. Sigal, M. I. Isard, B. H. Sigelman, M. J. Black

**Tracking Loose-limbed People (CVPR ’04)**
L. Sigal, S. Bhatia, S. Roth, M. J. Black, M. Isard
Preliminary Experiments: Vehicle Detection and Tracking

Top-Left

Top-Right

Original Image

Bottom-Left

Bottom-Right

Object
Preliminary Experiments: Vehicle Detection and Tracking

- Frame 12
- Frame 32
- Frame 52

- Part detectors are unreliable
Preliminary Experiments:
Vehicle Detection and Tracking
Preliminary Experiments: Pedestrian Detection

**Initialization (based on likelihood)**

- **Head**
- **Left-Upper Body**
- **Right-Upper Body**
- **Lower-Body**
- **Object**

**Pedestrian Parts/Components**

- **Object (GOM+BP)**
- **Parts (GOM+BP)**
Conclusions

- New framework that provides unified approach to object/detection and tracking
  - Tracking can benefit from object detection to resolve transient failures
  - Object detection can benefit from temporal consistency
- Part-based object detection and tracking formulated using Graphical Models and solved using approximate BP
- We can successfully detect and track two classes of objects (pedestrians and cars)
Future Work

- Image likelihoods are not really independent (correlations may be explicitly modeled)

- Multi-target detection
  - Currently we can detect multiple targets by exclusion (one target at a time)

- Unsupervised / semi-supervised learning of Graphical Object Models

Distributed Occlusion Reasoning for Tracking with Nonparametric Belief Propagation
E. Sudderth, M. Mandel, W. Freeman, A. Willsky

NIPS ‘04
Thank you !!!