Optimisation
Outline

1. Graph Fitting
   (i) Tree Structures - Dynamic Programming
   (ii) Beyond Trees - Tree Sub-Problems
   (iii) Loopy Belief Propagation

2. Response Fitting
   (i) Gradient Methods
   (ii) Convex Quadratic Fitting
   (iii) Constrained Mean Shift

3. Graph vs. Response Fitting
   (i) No Silver Bullet
   (ii) Future Directions
Still to Come......

Does the image at $p_i$ look like the $i^{th}$ part?

$$\min_p \sum_{i=1}^{N} D_i(p_i) + \lambda \ R(p)$$

Do the joint locations of the parts match the object?

$N = \text{no. of parts}$
Important Message

- Objective function is only as useful, as how well you can optimize it.
Exhaustive Search (N > 1)
Exhaustive Search (N > 1)

...
Exhaustive Search (N > 1)

Feature Extraction

Local Search

Part Responses

Color Encoding of Filter Responses

Felzenszwalb, Girshick, McAllester & Ramanan, 2010
Exhaustive Search (N > 1)

$D_1$, $D_2$, $D_N$
\[ p^* = \arg \min_p \sum_{i=1}^{N} D_i(p_i) + \lambda \, R(p) \]

\[ \mathcal{O}(M^N) \]

\[ p^* = [p_1^T, \ldots, p_N^T]^T \]
\begin{align*}
\mathbf{p}^* &= \arg \min_{\mathbf{p}} \sum_{i=1}^{N} D_i(\mathbf{p}_i) + \lambda R(\mathbf{p}) \\
\mathcal{O}(M^N) \\
\mathbf{p}^* &= [\mathbf{p}_1^T, \ldots, \mathbf{p}_N^T]^T
\end{align*}
Computational Cost

\[ p_1^* = \arg \min_{p_1} D_1(p_1) \]

\[ p_2^* = \arg \min_{p_2} D_2(p_2) \]

\[ p_N^* = \arg \min_{p_N} D_N(p_N) \]
Computational Cost

\[
\begin{align*}
\mathbf{p}_1^* &= \arg \min_{\mathbf{p}_1} D_1(\mathbf{p}_1) \\
\mathbf{p}_2^* &= \arg \min_{\mathbf{p}_2} D_2(\mathbf{p}_2) \\
\mathbf{p}_N^* &= \arg \min_{\mathbf{p}_N} D_N(\mathbf{p}_N)
\end{align*}
\]

\[
\begin{align*}
D_1 &\quad \mathcal{O}(M) \\
D_2 &\quad \mathcal{O}(M) \\
D_N &\quad \mathcal{O}(M)
\end{align*}
\]
Exhaustive Search

\[ O(M^N) \]

“We can do much better than this if the graph is sparse.”
Exhaustive Search

\[ O(NM^2) \]

“We can do much better than this if the graph is sparse.”
Tree Regularization

• Sparse graph of particular interest is a tree,

Felzenszwalb & Huttenlocher, 2005
Tree Regularization

- Sparse graph of particular interest is a tree,

Felzenszwalb & Huttenlocher, 2005
Dynamic Programming

• Globally optimal solution to any tree graph can be found using “Dynamic Programming”.
Dynamic Programming

- Globally optimal solution to any tree graph can be found using “Dynamic Programming”.

\[
score_j(p_j) = D_j(p_j) + \sum_{k \in \text{kids}(j)} m_k(p_j)
\]
Dynamic Programming

• Globally optimal solution to any tree graph can be found using “Dynamic Programming”.

\[ \text{score}_j(p_j) = D_j(p_j) + \sum_{k \in \text{kids}(j)} m_k(p_j) \]
Dynamic Programming

- Globally optimal solution to any tree graph can be found using “Dynamic Programming”.

\[
\text{score}_j(p_j) = D_j(p_j) + \sum_{k \in \text{kids}(j)} m_k(p_j)
\]
Dynamic Programming

- Globally optimal solution to any tree graph can be found using “Dynamic Programming”.

\[
score_j(p_j) = D_j(p_j) + \sum_{k \in \text{kids}(j)} m_k(p_j)
\]
Dynamic Programming

• Any tree-structure can take advantage of this redundancy.
• Globally optimal solution can be found using “Dynamic Programming”.

\[
\text{score}_j(p_j) = D_j(p_j) + \sum_{k \in \text{kids}(j)} m_k(p_j)
\]

\[
m_j(p_i) = \min_{p_j} \text{score}_j(p_j) + \lambda R(p_i, p_j)
\]
Dynamic Programming

- Any tree-structure can take advantage of this redundancy.
- Globally optimal solution can be found using “Dynamic Programming”.

\[
\text{score}_j(p_j) = D_j(p_j) + \sum_{k \in \text{kids}(j)} m_k(p_j)
\]

\[
m_j(p_i) = \min_{p_j} \text{score}_j(p_j) + \lambda R(p_i, p_j) \quad \text{“message passing”}
\]
Dynamic Programming

• Any tree-structure can taken advantage of this redundancy.
• Globally optimal solution can be found using “Dynamic Programming”.

\[
\text{score}_j(p_j) = D_j(p_j) + \sum_{k \in \text{kids}(j)} m_k(p_j)
\]

\[
m_j(p_i) = \min_{p_j} \text{score}_j(p_j) + \lambda R(p_i, p_j) \quad \text{“message passing”}
\]

\[O(M)\]
Dynamic Programming

- Any tree-structure can take advantage of this redundancy.
- Globally optimal solution can be found using “Dynamic Programming”.

\[
\text{score}_j(p_j) = D_j(p_j) + \sum_{k \in \text{kids}(j)} m_k(p_j)
\]

\[O(NM^2)\]

\[m_j(p_i) = \min_{p_j} \text{score}_j(p_j) + \lambda R(p_i, p_j) \quad \text{“message passing”}
\]

\[O(M)\]
Can we do better?

- When we assume that $R(p)$ is a quadratic, cost of searching each part becomes,

$$\text{score}_j(p_j) = D_j(p_j) + \sum_{k \in \text{kids}(j)} m_k(p_j)$$

$O(NM)$

“Uses the max-convolution/distance transform.”

Felzenszwalb & Huttenlocher, 2005
• Once we have the optimal parent score, we can back-track to find the children,
• Once we have the optimal parent score, we can back-track to find the children,
Once we have the optimal parent score, we can back-track to find the children,
Backtracking

- Once we have the optimal parent score, we can back-track to find the children,
Results

~1 second to search all scales.

Felzenszwalb, Girshick, McAllester & Ramanan, 2010
Mixtures of Parts

• We have assumed a single response for each part:

\[
\min_{\mathbf{p}} \sum_{i=1}^{N} D_i(p_i) + \lambda R(\mathbf{p})
\]
Mixtures of Parts

• One can entertain multiple responses:

\[
\min_{\mathbf{p}, \mathbf{t}} \sum_{i=1}^{N} D_i(\mathbf{p}_i, t_i) + \lambda R(\mathbf{p}, \mathbf{t})
\]

where \( t = [t_1, \ldots, t_N] \)

\( t_i \in [1, \ldots, T_i] \)

\( T_i = \text{number of detectors for part } i \)

Yang & Ramanan, 2011
Mixtures of Parts

- One can entertain multiple responses:

\[
\min_{p,t} \sum_{i=1}^{N} D_i(p_i, t_i) + \lambda R(p, t)
\]

where \( t = [t_1, \ldots, t_N] \)

\( t_i \in [1, \ldots, T_i] \)

\( T_i = \) number of detectors for part \( i \)

- Solution can still be found efficiently using Dynamic Programming.

Yang & Ramanan, 2011
Mixtures of Parts

- Approach is useful, as it can allow smaller detectors.
- Smaller detectors increase likelihood of the aperture problem.
Mixtures of Parts

• Approach is useful, as it can allow smaller detectors.
• Smaller detectors increase likelihood of the aperture problem.

\[
\min_{p,t} \sum_{i=1}^{N} D_i(p_i, t_i) + \lambda R(p, t)
\]

• Regularizer attempts to ensure that linked detectors look for solutions with similar orientations.
Mixtures of Parts

Figure 6: Results on the Parse dataset. We show 27 part bounding boxes reported by our algorithm for each image. The top 3 rows show successful examples, while the bottom row shows failure cases. Examining failure cases from left to right, we find our model is not flexible enough to model horizontal people, is confused by overlapping people, suffers from double-counting phenomena common to tree models (both the left and right legs fire on the same image region), and is confused when objects partially occlude people.

Figure 7: Results on the Buffy dataset. We show 17 part bounding boxes, corresponding to upper body parts, reported by our algorithm. The left 4 columns show successful examples, while the right column shows failure cases. From top to bottom, we see that our model still has difficulty with raised arms and is confused by vertical limb (like clutter in the background).

Yang & Ramanan, 2011
Non-Trees?

• What about non-tree structures?
Non-Trees?

- What about non-tree structures?
Non-Trees?

• What about non-tree structures?
Tree Sub-Problems

• One popular solution is to break a more complicated graph into a number of tree sub-problems.
  Felzenszwalb & Huttenlocher, 2005 (Importance Sampling using Trees)
  Wang & Mori, 2008 (Mixtures of Trees)
  Tian & Scharloff, 2010 (Branch & Bound with Trees)
One popular solution is to break a more complicated graph into a number of tree sub-problems.

- Felzenszwalb & Huttenlocher, 2005 (Importance Sampling using Trees)
- Wang & Mori, 2008 (Mixtures of Trees)
- Tian & Scharloff, 2010 (Branch & Bound with Trees)

These approaches have a couple of drawbacks,
- They assume that the problem can be broken down into tree sub-problems.
- Finding a solution is slow.
- Only guaranteed of a local-minima (with exception of Branch & Bound methods).
Loopy Belief Propagation

- Loopy belief propagation is another approach.
- Attempts to apply tree-based belief propagation iteratively.
- A number of drawbacks,
  - No guarantee of convergence (although empirically performs well).
  - Can be slow to converge.
  - Requires the graph to be relatively sparse.

Gu & Kanade, 2007
Another Direction....

- Up until now we have been discussing methods that take advantage of redundancies in,

\[
\min_{\mathbf{p}} \sum_{i=1}^{N} D_i(\mathbf{p}_i) + \lambda \ R(\mathbf{p})
\]
Another Direction....

• Up until now we have been discussing methods that take advantage of redundancies in,

\[
\min_p \sum_{i=1}^{N} D_i(p_i) + \lambda R(p)
\]
Another Direction....

• Up until now we have been discussing methods that take advantage of redundancies in,

\[
\min_p \sum_{i=1}^{N} D_i(p_i) + \lambda R(p)
\]

“Graph Fitting”
Another Direction....

- Are there redundancies in the responses themselves that we can take advantage of?

\[
\min_p \sum_{i=1}^{N} D_i(p_i) + \lambda R(p)
\]
Another Direction....

- Are there redundancies in the responses themselves that we can take advantage of?

\[
\min_{p} \sum_{i=1}^{N} D_i(p_i) + \lambda R(p)
\]

“Response Fitting”
Spatial Correlations

Even from a casual inspection of natural images, one can see that neighboring spatial locations are strongly correlated in intensity. This is demonstrated in Figure 3, which shows scatterplots of pairs of intensity values, separated by three different distances, and averaged over absolute position of several different natural images. The standard measurement for summarizing these dependencies is the autocorrelation function, $C(x, y)$, which gives the correlation (average of the product) of the intensity at two locations as a function of relative position. From the examples in Figure 3, one can see that the strength of the correlation falls with distance.

By computing the correlation as a function of relative separation, we are assuming that the spatial statistics in images are translation invariant. As described above, Reinagel & Zador (1999) recorded eye positions of human observers viewing natural images and found that correlation strength falls faster near these positions than generic positions.
Pixels Are Strongly Correlated

The assumption of translation invariance implies that images may be decorrelated by transforming to the frequency (Fourier) domain. The two-dimensional power spectrum can then be reduced to a one-dimensional function of spatial frequency by performing a rotational average within the two-dimensional Fourier plane. Empirically, many authors have found that the spectral power of natural images falls with frequency, $f$, according to a power law, $1/f^p$, with estimated values for $p$ typically near 2 [see Tolhurst (1992) or Ruderman & Bialek (1994) for reviews]. An example is shown Figure 4.

The environmental causes of this power law behavior have been the subject of considerable speculation and debate. One of the most commonly held beliefs is that it is due to scale invariance of the visual world. Scale invariance means that the statistical properties of images should not change if one changes the scale at which observations are made. In particular, the power spectrum should not change shape under such rescaling. Spatially rescaling the coordinates of an image by a factor of $\alpha$ leads to a rescaling of the corresponding Fourier domain axes by a factor of $1/\alpha$. Only a Fourier spectrum that falls as a power law will retain its shape under this transformation. Another commonly proposed theory is that the $1/f^2$ power spectrum is due to the presence of edges in images, because edges themselves Figure 4.

Another Direction

\[ p = \{ \text{x-translation}, \text{y-translation} \} \]
Another Direction

\[ p = \{x\text{-translation}, y\text{-translation}\} \]
Another Direction
Constrained Local Models

• Constrained local models (CLMs) encapsulate many methods in existence at the moment in non-rigid face alignment literature,

• Best known example are “Active Shape Models” (ASMs).
  
  Cootes and Taylor, 1992 (Active Shape Models)
  Zhou, Gu, and Zhang, 2003 (Bayesian Tangent Shape Models)
  Cristinacce and Cootes, 2004. (Constrained Local Models)

• Related work also in “Active Appearance Models” (AAMs).
  
  Cootes, Edwards and Taylor, 1998 (Active Appearance Models)
  Matthews and Baker, 2004 (Active Appearance Models Revisited)
Constrained Local Models

• Start with an initial guess.
Constrained Local Models

• Start with an initial guess.
Constrained Local Models

- Start with an initial guess.
- Warp to a fixed size template (e.g., 110x110 pixels).
• Try to fit a single point.
• Try to fit a single point.
CLM - Extracting Patch Responses

• Try to fit a single point.
CLM - Extracting Patch Responses

- Try to fit a single point.
Constrained Local Models

"$\text{nth Constrained Local Search Area}$"
it has been shown that the warp function is expensive and require good initialization. In order to employ general purpose optimization techniques, such as problem as a form of graph optimization \[
\end{equation}
\] image to be aligned. (b) shows the local search responses using positive examples and respectively. A high intensity value indicates a small matching error.

"Uses ZNCC"
Constrained Local Models

\[ \mathcal{O}(L) \ll \mathcal{O}(M) \]

Remember: \( \mathcal{O}(L) \ll \mathcal{O}(M) \)
CLM - Extracting Patch Responses

- 68 points in total.

(110x110 pixels)
CLM - Extracting Patch Responses

(110x110 pixels)

- 68 points in total.
CLM - Extracting Patch Responses

- Get responses for all N patch experts.

\[ D_1(p_1) = \]

\[ D_N(p_N) = \]
CLM Algorithm

• Unlike Graph Fitting, CLM algorithm is iterative:

Step 1. Pre-compute $N D_i^t(p_i)$ local responses.

Step 2. $\Delta p^* = \arg \min_{\Delta p} \sum_{i=1}^{N} D_i^t(p_i + \Delta p_i) + \lambda R(p + \Delta p)$ “Optimization step.”

Step 3. $p^{t+1} \leftarrow p^t \odot \Delta p^*$ “Update the patch center positions.”

Step 4. repeat steps 1 - 3. “Iterate until convergence.”
CLM Algorithm

- Unlike Graph Fitting, CLM algorithm is iterative:

**Step 1.** Pre-compute N \( D_i^t(p_i) \) local responses.

**Step 2.** \( \Delta p^* = \arg \min_{\Delta p} \sum_{i=1}^{N} D_i^t(p_i + \Delta p_i) + \lambda R(p + \Delta p) \) “Optimization step.”

**Step 3.** \( p^{t+1} \leftarrow p^t \odot \Delta p^* \) “Update the patch center positions.”

**Step 4.** repeat steps 1 - 3. “Iterate until convergence.”

Solving Step 2 is crucial, and most challenging component!!!
Why an Iterative Solution?

• Iterative approach allows for smaller more accurate detectors.
• Aperture problem is handled by increasingly better estimates of scale and rotation.
• Traditionally, CLMs have been applied to raw or photometric normalized pixels (not HOG style descriptors that give greater invariance).
• As discussed earlier, dealing with raw pixels does not blur/throw away position information.
• Also, raw pixels have clear computational advantages over descriptors.
In this section, we investigate Active Shape Models.

- A couple of solutions have been proposed in literature.
- Most straightforward approach is to,

\[ p_1^* = \arg \min_{p_1} D^t_1(p_1) \quad O(L) \]

\[ p_2^* = \arg \min_{p_2} D^t_2(p_2) \quad O(L) \]

\[ p_N^* = \arg \min_{p_N} D^t_N(p_N) \quad O(L) \]
Active Shape Models

- A couple of solutions have been proposed in literature.
- Most straightforward approach is to,

\[ \Delta p^* = \arg \min_{\Delta p} \sum_{i=1}^{N} w_i \|p^t_i + \Delta p_i - p^*_i\|^2 + \lambda R(p^t_i + \Delta p) \]

where,

\[ w_k = \text{patch expert confidence at local minimum} \]
\[ p^t_i = \text{patch position at iteration } t \]

A couple of solutions have been proposed in literature.
Most straightforward approach is to,

\[
\Delta p^* = \arg \min_{\Delta p} \sum_{i=1}^{N} w_i \| p_i^t + \Delta p_i - p_i^* \|^2 + \lambda R(p_i^t + \Delta p)
\]

where,

\[
w_k = \text{patch expert confidence at local minimum}
\]
\[
p_i^t = \text{patch position at iteration } t
\]

This second step has relatively no computational cost.

Can we do Better?

- A common approach in vision literature is to use gradient information (e.g. LK algorithm),

\[ D(p) = \| I(p) - T(0) \|^2 \]

Lucas & Kanade, 1981 (LK Algorithm)
Matthews and Baker, 2004 (Active Appearance Models Revisited)
Can we do Better?

• A common approach in vision literature is to use gradient information (e.g. LK algorithm),

\[ D(p) = f\{I(p) - T(0)\} \]

Lucas & Kanade, 1981 (LK Algorithm)
Matthews and Baker, 2004 (Active Appearance Models Revisited)
Can we do Better?

A common approach in vision literature is to use gradient information (e.g. LK algorithm),

\[
D(p) = f\{I(p) - T(0)\}
\]

Lucas & Kanade, 1981 (LK Algorithm)
Matthews and Baker, 2004 (Active Appearance Models Revisited)

\[
D(p + \Delta p) \approx f\{I(p) + \frac{\partial I(p)}{\partial p} \Delta p - T(0)\}
\]
Can we do Better?

- A common approach in vision literature is to use gradient information (e.g. LK algorithm),

\[ D(p) = f\{I(p) - T(0)\} \]

Lucas & Kanade, 1981 (LK Algorithm)
Matthews and Baker, 2004 (Active Appearance Models Revisited)

\[ D(p + \Delta p) \approx \Delta p^T A \Delta p + b^T \Delta p + c \]
Can we do Better?

- A common approach in vision literature is to use gradient information (e.g. LK algorithm),

\[ D(p) = f\{I(p) - T(0)\} \]

Lucas & Kanade, 1981 (LK Algorithm)
Matthews and Baker, 2004 (Active Appearance Models Revisited)

\[ D(p + \Delta p) \approx \Delta p^T A \Delta p + b^T \Delta p + c \]

“Forces local response INDIRECTLY to be a convex quadratic.”
Problems with Gradients

- Have small spatial support.
- Do not handle noise and illumination variation well.
- Need to be close to the global minima to converge.
- Only compatible with certain objective functions.

Liu et al., 2008 (Boosted Active Appearance Models)
Lucey, 2008 (Support Vector Tracking)
Convex Quadratic Fitting

- Another alternative is to force the responses DIRECTLY to be a convex quadratic,

\[
\begin{align*}
\text{Response maps, RES, and their approximations used in Convex Quadratic Fitting} & \quad = \\
\text{Another alternative is to force the responses DIRECTLY to be a convex quadratic,} & \\
\text{Convex Quadratic Fitting} & \\
\end{align*}
\]

Wang, Lucey & Cohn, 2008
Results

AAM	ASM	CQF
Mode Seeking

- Minimizing a convex quadratic has a probabilistic interpretation as finding the mode of a Gaussian distribution.
Mode Seeking

- Minimizing a convex quadratic has a probabilistic interpretation as finding the mode of a Gaussian distribution.

- Gaussian assumption is often too strong, mode seeking of other distributions can be entertained.
Mode Seeking

- Minimizing a convex quadratic has a probabilistic interpretation as finding the mode of a Gaussian distribution.

- Gaussian assumption is often too strong, mode seeking of other distributions can be entertained.

- If non-Gaussian, however, mode(s) MUST be found iteratively.
  - Uses variations of the EM-Algorithm,
    - Can be slow (iterative).
    - Guaranteed of only local minima.
Constrained Mean Shift
Constrained Mean Shift

- Saragih et al. proposed constrained mean shift.
- Can be applied extremely efficiently using LUT.
- Vary kernel density estimate (KDE) to perform coarse to fine fitting.

Saragih, Lucey and Cohn, ICCV 2009. (Constrained Mean Shifts)
Constrained Mean Shift

Each landmark formulation is the selection of the search region for one of the remaining open questions with the CLM.

3.3.2 1D vs. 2D Search Regions

...around the current landmark estimate (Cristinacce and Cootes, 2007; Wang et al., 2008a).

...performed exhaustive local search along a profile that is typically (manually) chosen to be perpendicular to the direction of largest edgedness of landmark appearance. Others search within a rectangular bounding box for detection can be evaluated using efficient capacity along the edge. As such, little motion in determining the best profile direction for these landmarks.

Although the optimal profile direction can be learned from the data through a cross-validation strategy, most methods simply define the direction as some function of the respective locations of the neighboring landmarks. Finally, in order to ensure robust fitting, the uncertainty along the profile (i.e. typically set to be equivalent to the uncertainty in the direction perpendicular to the landmark) is assumed to coincide with the uncertainty of the landmark. Therefore, restricting the search to a horizontal search profile (i.e. typically set to be equivalent to the landmark for square search regions) does not strictly require. For example, landmarks on the human face almost always include eye and lip corners which are typically localized with the help of facial features.

Deformable objects are indeed placed on edges, this is not strictly required. For example, landmarks on the file and rectangular search regions, where the choice of regions. For these landmarks, the local structure is sufficient to be known, one might choose to precompute the kernel evaluations by assuming this grid closest to the current estimate of a PDM landmark and estimate the kernel evaluations by assuming the landmark is actually placed at that node (see Fig. 4). This only involves a table lookup and can be performed efficiently. The higher the granularity of the grid the better the approximation will be, at the cost of greater storage requirements but without a significant increase in computational complexity. Although such an approximation ruins the strictly improving property of the landmark estimates and the likelihood of any...
Constrained Mean Shift

\[ \Psi_i \]

\[ \mu_i \]

\[ g_k \]

\[ N(x_i; y_i, \rho I) \approx N(g_k; c, \rho I) \]

Saragih, Lucey and Cohn, ICCV 2009. (Constrained Mean Shifts)
Results

ASM

CQF

BTSM

CMS
Results

The results show a consistent trend in the relative performance of the four methods. Firstly, CQF has the capability to fit the largest proportion of images at a given shape RMS error. The ASM method is the fastest, followed by the GMM method, and then the CQF method. The KDE method is the slowest, taking significantly longer to fit the images. The shape RMS error for each method is as follows:

- ASM: 88ms
- CQF: 98ms
- BTSM: 2410ms
- CMS: 121ms

The graphs in Figure 4 illustrate the proportion of images that can be fitted within a certain shape RMS error for each method. The x-axis represents the shape RMS error, and the y-axis represents the proportion of images. The different lines represent the four methods, with unique markers for each method.
CMS Results

Saragih, Lucey, Cohn, ICCV 2009.
Saragih, Lucey, Cohn, IJCV 2011.
Saragih, Lucey, Cohn, AFGR 2011.
Best Method is Domain Specific

No Silver Bullet
## Graph vs. Response Fitting

<table>
<thead>
<tr>
<th>Desired Characteristics</th>
<th>Graph Fitting</th>
<th>Response Fitting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-Computed Responses</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>Sub-Pixel Accuracy</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>No Initialization</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>No Descriptors</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>Real-Time Performance</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>Efficient Non-Tree Graphs</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>Joint learning $D(p)$ &amp; $R(p)$</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>Mixtures of Detectors</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>3D inference</td>
<td>×</td>
<td>✓</td>
</tr>
</tbody>
</table>
Promising Directions
Promising Directions

- Applying space-time regularizations.
Promising Directions

- Applying space-time regularizations.
- Hybrid Graph- and Response- Fitting Approaches.
Promising Directions

• Applying space-time regularizations.
• Hybrid Graph- and Response- Fitting Approaches.
• Hybrid descriptor/pixel algorithms.
Promising Directions

- Applying space-time regularizations.
- Hybrid Graph- and Response- Fitting Approaches.
- Hybrid descriptor/pixel algorithms.
- 3D inference with Graph Fitting approaches.
THANKS