Where’s What?
Towards Multi-Modal Semantic Mapping of Urban Environments

Ingmar Posner, Mark Cummins and Paul Newman
Mobile Robotics Group
Oxford University

Sponsored by:
SEAS DTC
Systems Engineering for Autonomous Systems
EPSRC
Engineering and Physical Sciences Research Council

Saturday, 12 December 2009
• Omni-directional vision.
• Stereo/Mono.
• 2D/3D laser scanners
• GPS.
• IMUs.
• Wheel Encoders
(Some of) The data...

**Stereo-Based Visual Odometry**

Fig. 10. A correct loop closure from the 70km data set. This not an unusual match – the system typically finds correct matches in the presence of considerable scene change when the image content is distinctive.

**6DOF Pose, Range and Reflectance**

Fig. 11. A typical false negative. The 1,000km sequence contains hundreds of kilometers of similar highway scenes, so the system’s inability to correctly identify this loop closure is unsurprising. This effect depresses the recall in the 1,000km results. However, this strong perceptual aliasing generates very few false positive detections.

**Omni-directional camera**

[Sibley & Mei, RSS 2009]
Roboticists are privileged:
Many sensors covering the same, contiguous workspace.

This provides ample scope for combining data from multiple sources, e.g.

• Data fusion (e.g. for SLAM).
• Appearance-based topological mapping and loop closure detection.
• Geometric constraints on appearance-based methods.
• Sensor augmentation.
• Cue integration.
Fusion of various sources of odometry information to obtain:
• a faithful reconstruction of an environment
• the robot poses (trajectory) within the environment
Exploration is tricky:

- locally, decent estimates from onboard sensors.
- globally, large errors accumulate over not so large distances!

[Image of a map next to a street view image.]

[Cummins and Newman: FAB-MAP vX.X]
Exploiting Information from Multiple Sources: Topo-Metric.

Exploration is tricky:

- locally, decent estimates from onboard sensors.
- globally, large errors accumulate over not so large distances!

Use yet another sensor to fix this!
Exploiting Information from Multiple Sources: Topo-Metric.

This allows us to build
- large scale metric maps (kms).
- even larger scale topological maps (Mms using FabMap)!

[Smith et al, The New College Vision and Laser Data Set, IJRR 09]
Each laser point then yields a range measurement correcting camera model and associated to the nearest pixel range vector all pixels which have an associated range measurement. Thus the inaccuracies due to the first-order prior must be relied upon over larger areas. This motivates using a wide-angle camera, which we are unaware of in other work.

We shall refer to pixel in the image. For reasons which will become clear we have a range measurement — in fact very few will. We use pixel

and

where notation of

Only the solution a single linear system is required. We shall begin by introducing our notation.

In this section we shall show how a general description of the method of [12] is most relevant to this work. It uses

and

ground truth with three sparse laser measurements. Our task is to use both

and a

Ground Truth with Sparse Laser Measurements.
Ground Truth with Sparse Laser Measurements.

Interpolation using laser only.
shall also refer to pixel in the image. For reasons which will become clear we have a range measurement — in fact very few will. We use each laser point then yields a range measurement $L$ of all pixels which have an associated range measurement.

The technique requires that the supplied range measurement $x^i$ contains some high density areas from which to seed the solution. Thus the inaccuracies due to the first-order prior must be relied upon over larger areas. This motivates the use of a second-order prior. We further provide a method of filling relatively large gaps between known range data. We make the assumption that spatially close nodes sharing similar colour, the techniques described in \[12\] and \[11\] are work.

In \[13\], an MRF Belief Propagation technique is used to exploit information from multiple sources: Complement. For reasons which will become clear we have a range measurement — in fact very few will. We use each laser point then yields a range measurement $L$ of all pixels which have an associated range measurement. The notation of $\Theta$ can be formulated in such a way that in the end, small areas can be filled with high density areas from which to seed the solution.

In this section we shall show how a general description of the problem can be formulated in such a way that in the end, small areas can be filled with high density areas from which to seed the solution. In \[13\], an MRF Belief Propagation technique is used to exploit information from multiple sources. In \[13\], an MRF Belief Propagation technique is used to exploit information from multiple sources. The method of \[12\] is most relevant to this work. It uses the notion that spatially close nodes sharing similar colour, the techniques described in \[12\] and \[11\] are work.

In \[13\], an MRF Belief Propagation technique is used to exploit information from multiple sources: Complement. We make the assumption that spatially close nodes sharing similar colour, the techniques described in \[12\] and \[11\] are work.

In \[13\], an MRF Belief Propagation technique is used to exploit information from multiple sources: Complement. The method of \[12\] is most relevant to this work. It uses the notion that spatially close nodes sharing similar colour, the techniques described in \[12\] and \[11\] are work.

In \[13\], an MRF Belief Propagation technique is used to exploit information from multiple sources: Complement. We make the assumption that spatially close nodes sharing similar colour, the techniques described in \[12\] and \[11\] are work.

In \[13\], an MRF Belief Propagation technique is used to exploit information from multiple sources: Complement. We make the assumption that spatially close nodes sharing similar colour, the techniques described in \[12\] and \[11\] are work.
Exploiting Information from Multiple Sources: Complement.

Each point in the image yields a range measurement — in fact very few will. We use the notation \(\mathbf{x}_i\) to imply the index variable \(i\). Note the laser data is sparse so not every pixel will contain some high density areas from which to seed the problem can be formulated in such a way that in the end, a set of 3D points from a scan can be projected into a pixel image, yielding a true surface. Thus the inaccuracies due to the first-order prior must be relied upon over larger areas. This motivates the use of range nodes. A pixel image, \(I\) can be projected into a vectorised image (all pixels stacked in a single vector of length \(N\)) where \(N\) is the number of pixels in the image. Our task is to use both laser measurements \(L\) and a set of \(D\) images \(I\) to fuse the information from both sources to significant improvements.

Under this assumption, given a neighbourhood of neighbouring ranges \(x_i\), we make a range prediction \(\hat{x}_i\) by understanding this cost as the prediction matrix. We define a suitably formed prediction matrix. We define the prediction matrix \(P\) where \(|\Theta\|\) is the identity matrix. While details of how \(W\) is formed will be returned to later, in Section VI. For now, consider that in the absence of cues to the contrary, surfaces are smooth and continuous and smooth. This ‘pins’ the estimate to lie near the regular and dense. This favours fronto-parallel surfaces, but does not suffer too greatly from this because the range measurements are sufficiently regular and dense, so the smoothness penalty between inferred ranges and observed edges in the optical image often correspond to discontinuities in depth, and that smooth surfaces tend to correspond areas of filling relatively large gaps between known range data. It uses the notion that spatially close nodes sharing similar colour, the techniques described in [12] and [11] are able to fuse the information from both sources to significant improvements.

A. Notation

\[\begin{align*}
\mathbf{x}_i & \in \mathbb{R}^3, \\
\Theta & \in \mathbb{R}^{3 \times 3}, \\
\sigma & \in \mathbb{R}^1, \\
\mathbf{w} & \in \mathbb{R}^3.
\end{align*}\]

Interpolation using laser and vision.

Interpolation using laser only.
Another application...

Higher-Order (Semantic) Mapping of Urban Environments.
• **We can easily build coloured point clouds, but we want more.**

• **We want scene labels…**

  … to aid with
  - Navigation and planning.
  - Action selection.
  - Human-machine interaction.

Appearance can *augment* metric/topological approaches.
Ideally, the classification framework should be:

- Principled.
  - Probabilistic.
- Introspective.
  - Able to handle context.
  - Know what measurements to trust (detector model).
- Flexible.
  - Adapt (learn) class models online and in real-time.
- Fast.
  - Classify in real-time.

Goal: Find a framework which delivers on these.
What’s to come:

• Pipeline
  – Combining 3D laser and image data.
  – Representation.
  – Features.

• Classification Framework
  – Patch Classification.
  – Scene Classification.

• Results

• Conclusions
Image Data

Operate on image patches with associated 3D geometry.
Operate on image patches with associated 3D geometry.
Operate on image patches with associated 3D geometry.
Operate on image patches with associated 3D geometry.
Operate on image patches with associated 3D geometry.
Bag-of-Words Representation
Bag-of-Words Representation
Bag-of-Words Representation

Compute Descriptors

Descriptor

Descriptor

Descriptor
Bag-of-Words Representation

Cluster Centres are “Words”
Bag-of-Words Representation

Observation

\[ z = \begin{bmatrix} 1 \\ 1 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \]

- Word 1
- Word 2
- Word 3
- Word \(|V|\)

Cluster Centres are “Words”
3D Geometry
• Surface Normal

2D Geometry
• Normalised x- and y-position in image

Colour
• Local hue histogram: 15 bins over a 15 x 15 pixel neighbourhood
• Local saturation histogram: 15 bins over a 15 x 15 pixel neighbourhood

Texture (via colour variation)
• Standard deviations of hue and saturation histograms.
Classification

Scene

Local Patch Classification

Scene-Wide Patch Classification

Labels
Classification: Stage I

Scene

Local Patch Classification

Scene-Wide Patch Classification

Labels

Saturday, 12 December 2009
### Stage I: Representing Classes

<table>
<thead>
<tr>
<th>Class ( k )</th>
<th>Exemplar</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>( n_k )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grass</td>
<td><img src="image1" alt="Image" /></td>
<td><img src="image2" alt="Image" /></td>
<td><img src="image3" alt="Image" /></td>
<td><img src="image4" alt="Image" /></td>
<td><img src="image5" alt="Image" /></td>
</tr>
<tr>
<td>Tarmac / Paved</td>
<td><img src="image6" alt="Image" /></td>
<td><img src="image7" alt="Image" /></td>
<td><img src="image8" alt="Image" /></td>
<td><img src="image9" alt="Image" /></td>
<td><img src="image10" alt="Image" /></td>
</tr>
<tr>
<td>Dirt Path</td>
<td><img src="image11" alt="Image" /></td>
<td><img src="image12" alt="Image" /></td>
<td><img src="image13" alt="Image" /></td>
<td><img src="image14" alt="Image" /></td>
<td><img src="image15" alt="Image" /></td>
</tr>
<tr>
<td>Textured Wall</td>
<td><img src="image16" alt="Image" /></td>
<td><img src="image17" alt="Image" /></td>
<td><img src="image18" alt="Image" /></td>
<td><img src="image19" alt="Image" /></td>
<td><img src="image20" alt="Image" /></td>
</tr>
<tr>
<td>Smooth Wall</td>
<td><img src="image21" alt="Image" /></td>
<td><img src="image22" alt="Image" /></td>
<td><img src="image23" alt="Image" /></td>
<td><img src="image24" alt="Image" /></td>
<td><img src="image25" alt="Image" /></td>
</tr>
<tr>
<td>Bush / Foliage</td>
<td><img src="image26" alt="Image" /></td>
<td><img src="image27" alt="Image" /></td>
<td><img src="image28" alt="Image" /></td>
<td><img src="image29" alt="Image" /></td>
<td><img src="image30" alt="Image" /></td>
</tr>
<tr>
<td>Vehicle</td>
<td><img src="image31" alt="Image" /></td>
<td><img src="image32" alt="Image" /></td>
<td><img src="image33" alt="Image" /></td>
<td><img src="image34" alt="Image" /></td>
<td><img src="image35" alt="Image" /></td>
</tr>
</tbody>
</table>
### Stage I: Representing Classes

<table>
<thead>
<tr>
<th>Exemplar</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>( n_k )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grass</td>
<td><img src="image1.jpg" alt="image" /></td>
<td><img src="image2.jpg" alt="image" /></td>
<td><img src="image3.jpg" alt="image" /></td>
<td>( \cdots )</td>
</tr>
<tr>
<td>Tarmac / Paved</td>
<td><img src="image4.jpg" alt="image" /></td>
<td><img src="image5.jpg" alt="image" /></td>
<td><img src="image6.jpg" alt="image" /></td>
<td>( \cdots )</td>
</tr>
<tr>
<td>Dirt Path</td>
<td><img src="image7.jpg" alt="image" /></td>
<td><img src="image8.jpg" alt="image" /></td>
<td><img src="image9.jpg" alt="image" /></td>
<td>( \cdots )</td>
</tr>
<tr>
<td>Textured Wall</td>
<td><img src="image10.jpg" alt="image" /></td>
<td><img src="image11.jpg" alt="image" /></td>
<td><img src="image12.jpg" alt="image" /></td>
<td>( \cdots )</td>
</tr>
<tr>
<td>Smooth Wall</td>
<td><img src="image13.jpg" alt="image" /></td>
<td><img src="image14.jpg" alt="image" /></td>
<td><img src="image15.jpg" alt="image" /></td>
<td>( \cdots )</td>
</tr>
<tr>
<td>Bush / Foliage</td>
<td><img src="image16.jpg" alt="image" /></td>
<td><img src="image17.jpg" alt="image" /></td>
<td><img src="image18.jpg" alt="image" /></td>
<td>( \cdots )</td>
</tr>
<tr>
<td>Vehicle</td>
<td><img src="image19.jpg" alt="image" /></td>
<td><img src="image20.jpg" alt="image" /></td>
<td><img src="image21.jpg" alt="image" /></td>
<td>( \cdots )</td>
</tr>
</tbody>
</table>

\[
C^k = \{ C_1^k, \ldots, C_{n_k}^k \}
\]
Representing Exemplars:

\[ C_i^k \triangleq \{ p(e_1|C_i^k), \ldots, p(e_v|C_i^k) \}\]

- event that a generator for word \( j \) exists

The Detector Model:

\[ D : \begin{cases} 
  p(z_j = 1|e_j = 0), & \text{false positive probability.} \\
  p(z_j = 0|e_j = 1), & \text{false negative probability.} 
\end{cases} \]

Detector Model: we do not assume a perfect sensor.
Stage I: Patch Classification

Exemplars

Class Models
Stage I: Patch Classification

Exemplars

Class Models

New Observation

\[ z^T = [1, 1, 0, \ldots, 1] \]
Stage I: Patch Classification

Exemplars

Class Models

New Observation

Which class \( k \)?

\[
z^T = [1, 1, 0, \ldots, 1]
\]
Stage I: Patch Classification

Exemplars

New Observation

Which class $k$?

$z^T = [1, 1, 0, \ldots, 1]$

$$p(C^k | z) \propto p(z | C^k)p(C^k)$$
Stage I: Patch Classification

Exemplars

Which class \( k \)?

New Observation

\[
\mathbf{z}^T = [1, 1, 0, \ldots, 1]
\]

\[
p(C^k | \mathbf{z}) \propto p(\mathbf{z} | C^k) p(C^k)
\]

Class \( k \)
Stage I: Patch Classification

Exemplars

New Observation

Which class $k$?

Class Models

$\mathbf{z}^T = [1, 1, 0, \ldots, 1]$

New Observation

$$p(C^k | \mathbf{z}) \propto p(\mathbf{z} | C^k) p(C^k)$$
Stage I: Patch Classification

Which class $k$?

Exemplars

New Observation

Class Models

Class Prior

$p(C^k | z) \propto p(z | C^k) p(C^k)$

$z^T = [1, 1, 0, \ldots, 1]$
Stage I: Patch Classification

Exemplars

Class Models

New Observation

Which class $k$?

$z^T = [1, 1, 0, \ldots, 1]$

$p(C^k | z) \propto p(z | C^k) p(C^k)$

Class Likelihood
Stage I: The Class Likelihood

Assume:

1) None of the training data are mislabeled.
2) All exemplars within a class are equally likely.

\[
p(z|C^k) = \sum_{i=1}^{n_k} p(z|C^k_i, C^k) p(C^k_i|C^k)
\]

\[
= \frac{1}{n_k} \sum_{i=1}^{n_k} p(z|C^k_i)
\]
Stage I: Estimation the Class Likelihood

Generative Model for Bag-of-Words Data

- Certain words **tend to co-occur**, because they are generated by the same object in the world.
- So words are **not independent**.

⇒ Learn a better model of the distribution over words.
Stage I: Estimation the Class Likelihood

Generative Model for Bag-of-Words Data

- Certain words **tend to co-occur**, because they are generated by the same object in the world.
- So words are **not independent**.

Learn a better model of the distribution over words.
Stage I: Estimation the Class Likelihood

Generative Model for Bag-of-Words Data

- Instead we learn a tree-structured Bayesian network to capture a first-order approximation to the PDF over word occurrence, using the Chow Liu algorithm:

\[
p(z) = p(z_1, z_2, \ldots, z_{|v|}) \\
\approx p(z_r) \prod_{q=1}^{|v|} p(z_q | z_{pq})
\]

- And so:

\[
p(z | C_i^k) \approx p(z_r | C_i^k) \prod_{q=1}^{|v|} p(z_q | z_{pq}, C_i^k)
\]
Assumptions:
1) Detector errors are independent of class (exemplar).
2) The probability of a generator variable for word $i$ existing is independent of the observations of all other words.
Stage I: Patch Classification

Assumptions:
1) Detector errors are independent of class (exemplar).
2) The probability of a generator variable for word $i$ existing is independent of the observations of all other words.
Assumptions:
1) Detector errors are independent of class (exemplar).
2) The probability of a generator variable for word $i$ existing is independent of the observations of all other words.
Stage I: Patch Classification

Assumptions:
1) Detector errors are independent of class (exemplar).
2) The probability of a generator variable for word $i$ existing is independent of the observations of all other words.

We can compute this!
Stage I: Learning a Class Model

Remember - a class model consists of exemplars:

\[ C^k = \{ C_1^k, \ldots, C_n^k \} \]

Learn class models using *labeled training data*. So for each new exemplar:

\[
p(e_q = 1|C_i^k, z) = \frac{p(z|e_q = 1, C_i^k)p(e_q = 1|C_i^k)}{p(z|C_i^k)}
\]
Stage I: Learning a Class Model

Remember - a class model consists of exemplars:

\[ C^k = \{ C_{i1}^k, \ldots, C_{n_k}^k \} \]

Learn class models using *labeled training data*. So for each new exemplar:

\[
p(e_q = 1 | C_i^k, z) = \frac{p(z|e_q = 1, C_i^k) p(e_q = 1 | C_i^k)}{p(z|C_i^k)}
\]

Word Existence Prior
Stage I: Learning a Class Model

Remember - a class model consists of exemplars:

\[ C^k = \{ C^k_1, \ldots, C^k_{n_k} \} \]

Learn class models using *labeled training data*. So for each new exemplar:

\[
p(e_q = 1 | C^k_i, z) = \frac{p(z|e_q = 1, C^k_i)p(e_q = 1|C^k_i)}{p(z|C^k_i)}
\]

Likelihood term given word existence
Stage I: Learning a Class Model

Remember - a class model consists of exemplars:

$$C^k = \{ C_1^k, \ldots, C_{n_k}^k \}$$

Learn class models using *labeled training data*. So for each new exemplar:

$$p(e_q = 1|C_i^k, z) = \frac{p(z|e_q = 1, C_i^k)p(e_q = 1|C_i^k)}{p(z|C_i^k)}$$

Likelihood Term
Stage I: Learning a Class Model

Remember - a class model consists of exemplars:

\[ C^k = \{ C^k_1, \ldots, C^k_{n_k} \} \]

Learn class models using *labeled training data*. So for each new exemplar:

\[
p(e_q = 1 | C^k_i, z) = \frac{p(z | e_q = 1, C^k_i)p(e_q = 1 | C^k_i)}{p(z | C^k_i)}
\]

Readily learned online!
Classification: Stage II

Scene

Local Patch Classification

Scene-Wide Patch Classification

Labels
Stage II: MRFs

*Markov Random Fields (MRF)*
- Family of undirected graphical models.
- Model joint distribution over (hidden) states and the available data.

Consider:

- $N_n$ - # of nodes
- $N_c$ - # of classes
- $x \in \mathbb{Z}^{N_n}$ - label configuration
- $x_s \in \{1, \ldots, N_c\}$

Perform MAP inference by minimising an energy function:

$$E(x|\theta) = \sum_{s \in V} \theta_s(x_s) + \sum_{(s,t) \in E} \theta_{st}(x_s, x_t)$$

Well known problem in CV & ML. Can be tackled using
- Graph Cut Variants.
- (Max-Product) Loopy BP.
- Sequential Tree-reweighted Message Passing (TRW-S).

Need to determine model *structure* and *parameters*. 
Markov Random Fields (MRF)
- Family of undirected graphical models.
- Model joint distribution over (hidden) states and the available data.

Consider:
- $N_n$ - # of nodes
- $N_c$ - # of classes

$x \in \mathbb{Z}^{N_n}$ - label configuration
$x_s \in \{1, \ldots, N_c\}$

Perform MAP inference by minimising an energy function:

$$E(x|\theta) = \sum_{s \in \mathcal{V}} \theta_s(x_s) + \sum_{(s,t) \in \mathcal{E}} \theta_{st}(x_s, x_t)$$

Well known problem in CV & ML. Can be tackled using
- Graph Cut Variants.
- (Max-Product) Loopy BP.
- Sequential Tree-reweighted Message Passing (TRW-S).

Need to determine model structure and parameters.
**Markov Random Fields (MRF)**
- Family of undirected graphical models.
- Model joint distribution over (hidden) states and the available data.

Consider:

- $N_n$: # of nodes
- $N_c$: # of classes
- $\mathbf{x} \in \mathcal{Z}^{N_n}$: label configuration
- $x_s \in \{1, \ldots, N_c\}$

Perform MAP inference by minimising an energy function:

$$E(\mathbf{x}|\theta) = \sum_{s \in \mathcal{V}} \theta_s(x_s) + \sum_{(s,t) \in \mathcal{E}} \theta_{st}(x_s, x_t)$$

Well known problem in CV & ML. Can be tackled using

- Graph Cut Variants.
- (Max-Product) Loopy BP.
- Sequential Tree-reweighted Message Passing (TRW-S).

Need to determine model *structure* and *parameters*. 
Stage II: Model Structure

- Operate on *superpixels*.
- Extract structure from neighbourhood relations in 2D.
- Capture adjacency dependencies.
Stage II: Model Structure

- Operate on superpixels.
- Extract structure from neighbourhood relations in 2D.
- Capture adjacency dependencies.
- Operate on *superpixels*.
- Extract structure from neighbourhood relations in 2D.
- Capture adjacency dependencies.
Stage II: Model Structure

- Operate on superpixels.
- Extract structure from neighbourhood relations in 2D.
- Capture adjacency dependencies.

Fast and intuitive extraction of sparse graphs.
Stage II: Model Formulation

Want to minimize

$$E(x|\theta) = \sum_{s \in V} \theta_s(x_s) + \sum_{(s,t) \in E} \theta_{st}(x_s, x_t)$$

*Unary* potentials:

$$\theta_s(x_{sk}) = 1 - p(C^k|z^s)$$  - observation for node $s$

*Assuming our classes fully partition the world (i.e. $p(C^k|z^s)$ is a distribution)*

*Binary* potentials:

$$\theta_{st}(x_i, x_j) = 1 - \phi_{i,j}$$

*Constitute our environmental prior.

*Learned from training data.*
Stage II: Model Formulation

Want to minimize

\[ E(x|\theta) = \sum_{s \in V} \theta_s(x_s) + \sum_{(s,t) \in E} \theta_{st}(x_s, x_t) \]

Unary potentials:

- **Stage I**
  \[ \theta_s(x_{sk}) = 1 - p(C^k|z^s) \]
  - observation for node \( s \)

  • Assuming our classes fully partition the world (i.e. \( p(C^k|z^s) \) is a distribution)

Binary potentials:

- • Constitute our environmental prior.
- • Learned from training data.
Stage II: Model Formulation

Want to minimize

\[ E(x | \theta) = \sum_{s \in V} \theta_s(x_s) + \sum_{(s,t) \in E} \theta_{st}(x_s, x_t) \]

*Unary potentials:*

\[ \theta_s(x_{sk}) = 1 - p(C^k | z^s) \quad z^s \quad \text{- observation for node } s \]

- Assuming our classes fully partition the world (i.e. \( p(C^k | z^s) \) is a distribution)

*Binary potentials:*

\[ \theta_{st}(x_i, x_j) = 1 - \phi_{i,j} \]

- Cost of transition between labels \( i \) and \( j \).

- Constitute our environmental prior.
- Learned from training data.
Stage II: MRF Smoothing in Action

Scene

Pre MRF

Post MRF

Model
Temporal Smoothing

*Binary potentials (temporal):*

\[ \theta_{st}(x_i, x_j) = 1 - \phi_{i,j} \]

\[ \phi_{i,j} = 1, \forall i \neq j, \]

\[ \phi_{i,i} = 0, \]

\[ t = 0 \]

\[ t = -1 \]

\[ t = -2 \]
Results

Training - Stage I

- **Jericho / Oxford**
  - 13.2 km track
  - 16,538 images
  - ca. 174 x 10^6 laser points

Test

- **Oxford Science Park**
  - 3.3 km track
  - 8,536 images
  - ca. 74 x 10^6 laser points
Results: Pre-MRF

Precision

Recall
Results: Post-MRF

Precision

Recall
Results: Numbers

<table>
<thead>
<tr>
<th>Class Details</th>
<th>Pre MRF</th>
<th>Spatial Context</th>
<th>Spatio-Temporal Context</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision [%]</td>
<td>Recall [%]</td>
<td>$F_{0.5}$</td>
</tr>
<tr>
<td>Name</td>
<td># Patches</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gr</td>
<td>82</td>
<td>80.3</td>
<td>69.5</td>
</tr>
<tr>
<td>Ta</td>
<td>1286</td>
<td>79.5</td>
<td>86.1</td>
</tr>
<tr>
<td>Di</td>
<td>127</td>
<td>21.4</td>
<td>47.2</td>
</tr>
<tr>
<td>Te</td>
<td>2199</td>
<td>73.3</td>
<td>75.5</td>
</tr>
<tr>
<td>Sm</td>
<td>898</td>
<td>54.1</td>
<td>36.2</td>
</tr>
<tr>
<td>Bu</td>
<td>175</td>
<td>41.7</td>
<td>38.9</td>
</tr>
<tr>
<td>Ve</td>
<td>165</td>
<td>35.9</td>
<td>34.6</td>
</tr>
</tbody>
</table>

Legend for class shortcuts: Grass, Tarmac/Paved, Dirt Path, Textured Wall, Smooth Wall, Bush/Foliage, Vehicle
<table>
<thead>
<tr>
<th>Class Details</th>
<th>Pre MRF</th>
<th>Spatial Context</th>
<th>Spatio-Temporal Context</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Name</td>
<td># Patches</td>
<td>Precision [%]</td>
</tr>
<tr>
<td></td>
<td>Gr</td>
<td>82</td>
<td>80.3</td>
</tr>
<tr>
<td></td>
<td>Ta</td>
<td>1286</td>
<td>79.5</td>
</tr>
<tr>
<td></td>
<td>Di</td>
<td>127</td>
<td>21.4</td>
</tr>
<tr>
<td></td>
<td>Te</td>
<td>2199</td>
<td>73.3</td>
</tr>
<tr>
<td></td>
<td>Sm</td>
<td>898</td>
<td>54.1</td>
</tr>
<tr>
<td></td>
<td>Bu</td>
<td>175</td>
<td>41.7</td>
</tr>
<tr>
<td></td>
<td>Ve</td>
<td>165</td>
<td>35.9</td>
</tr>
</tbody>
</table>

Legend for class shortcuts: Grass, Tarmac/Paved, Dirt Path, Textured Wall, Smooth Wall, Bush/Foliage, Vehicle.
## Results: Timing

<table>
<thead>
<tr>
<th>Process</th>
<th>Mean (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plane Segmentation</td>
<td>2000</td>
<td>2800</td>
</tr>
<tr>
<td>Feature Extraction</td>
<td>89</td>
<td>125</td>
</tr>
<tr>
<td>Feature Quantization</td>
<td>4</td>
<td>90</td>
</tr>
<tr>
<td>Image Segmentation</td>
<td>960</td>
<td>1130</td>
</tr>
<tr>
<td>Patch Classification</td>
<td>850</td>
<td>3480</td>
</tr>
<tr>
<td>MRF Construction</td>
<td>63.5</td>
<td>453.9</td>
</tr>
<tr>
<td>MRF Inference</td>
<td>6.0</td>
<td>22.0</td>
</tr>
</tbody>
</table>

*Overall*: 4.0 seconds, 8.1 seconds

Our real-time constraint: ~ 3 s
### Results: Timing

<table>
<thead>
<tr>
<th>Process</th>
<th>Mean (ms)</th>
<th>Max (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plane Segmentation</td>
<td>2000</td>
<td>2800</td>
</tr>
<tr>
<td>Feature Extraction</td>
<td>89</td>
<td>125</td>
</tr>
<tr>
<td>Feature Quantization</td>
<td>4</td>
<td>90</td>
</tr>
<tr>
<td>Image Segmentation</td>
<td>960</td>
<td>1130</td>
</tr>
<tr>
<td>Patch Classification</td>
<td>850</td>
<td>3480</td>
</tr>
<tr>
<td>MRF Construction</td>
<td>63.5</td>
<td>453.9</td>
</tr>
<tr>
<td>MRF Inference</td>
<td>6.0</td>
<td>22.0</td>
</tr>
<tr>
<td><strong>Overall</strong></td>
<td>4.0 seconds</td>
<td>8.1 seconds</td>
</tr>
</tbody>
</table>

Our real-time constraint: ~ 3 s
So, let’s see:

• Principled.
  – Probabilistic.
• Introspective.
  – Able to handle context.
  – Know what measurements to trust (detector model).
• Flexible.
  – Adapt (learn) class models online and in real-time.
• Fast.
  – Classify in real-time.
So, let’s see:

• Principled.
  – Probabilistic.
• Introspective.
  – Able to handle context.
  – Know what measurements to trust (detector model).
• Flexible.
  – Adapt (learn) class models online and in real-time.
• Fast.
  – Classify in real-time.
So, let’s see:

• Principled.
  – Probabilistic.

• Introspective.
  – Able to handle context.
  – Know what measurements to trust (detector model).

• Flexible.
  – Adapt (learn) class models online and in real-time.

• Fast.
  – Classify in real-time.
So, let’s see:

- Principled.
  - Probabilistic.
- Introspective.
  - Able to handle context.
  - Know what measurements to trust (detector model).
- Flexible.
  - Adapt (learn) class models online and in real-time.
- Fast.
  - Classify in real-time.
So, let’s see:

• Principled.
  – Probabilistic.

• Introspective.
  – Able to handle context.
  – Know what measurements to trust (detector model).

• Flexible.
  – Adapt (learn) class models online and in real-time.

• Fast.
  – Classify in real-time.
The things we didn’t tell you …

… there are currently some obvious shortcomings.
The things we didn’t tell you …

… there are currently some obvious shortcomings.
### Comparative Analysis

<table>
<thead>
<tr>
<th>Class Details</th>
<th>Voted SVM</th>
<th></th>
<th>Pre MRF</th>
<th></th>
<th>Spatio-Temporal Context</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Precision (%)</td>
<td>Recall (%)</td>
<td>$F_{0.5}$</td>
<td>Precision (%)</td>
<td>Recall (%)</td>
<td>$F_{0.5}$</td>
</tr>
<tr>
<td>Gr</td>
<td>96.6</td>
<td>98.1</td>
<td>96.9</td>
<td>80.3</td>
<td>69.5</td>
<td>77.9</td>
</tr>
<tr>
<td>Ta</td>
<td>97.7</td>
<td>89.0</td>
<td>95.8</td>
<td>79.5</td>
<td>86.1</td>
<td>80.8</td>
</tr>
<tr>
<td>Di</td>
<td>46.4</td>
<td>84.8</td>
<td>51.0</td>
<td>21.4</td>
<td>47.2</td>
<td>24.0</td>
</tr>
<tr>
<td>Te</td>
<td>82.7</td>
<td>73.5</td>
<td>80.7</td>
<td>73.3</td>
<td>75.5</td>
<td>73.8</td>
</tr>
<tr>
<td>Sm</td>
<td>56.9</td>
<td>64.4</td>
<td>58.3</td>
<td>54.1</td>
<td>36.2</td>
<td>49.2</td>
</tr>
<tr>
<td>Bu</td>
<td>60.6</td>
<td>62.8</td>
<td>61.0</td>
<td>41.7</td>
<td>38.9</td>
<td>41.1</td>
</tr>
<tr>
<td>Ve</td>
<td>43.7</td>
<td>80.1</td>
<td>48.1</td>
<td>35.9</td>
<td>34.6</td>
<td>35.6</td>
</tr>
</tbody>
</table>

Legend for class shortcuts: Grass, Tarmac/Paved, Dirt Path, Textured Wall, Smooth Wall, Bush/Foliage, Vehicle
Conclusions

Robotics is a natural domain for the integration of multiple sensing modalities:

- Many sensors covering the same, contiguous workspace.

In fact, some robotics applications **demand** sensor integration. E.g.:

- Driving day and night.
- Close-range, long-range reconstruction.

A lot of the machinery developed in ML and statistics find application here:

- Probabilistic methods.
- Discriminative and generative classification models.
- Regression models.

**Significant differences in constraints wrt other, more traditional domains:**

- Often require real time operation.
- Action selection (or the meaning of uncertainties).

**A lot of scope for information transfer between modalities:**

- Cue integration. (At what level?)
- Transfer learning.
- Domain adaptation.
Thank you...
Spatial vs. Temporal Smoothing

- (a) and (c) present results obtained using temporal smoothing only.
- (b) and (d) present results with both temporal and spatial edge information included.