Artificial Intelligence in Robotics Mapping of changing environments - Part I: Visual Navigation in Changing Environments

Tom Krajník

Artificial Intelligence Laboratory, Czech Technical University in Prague

Dec 2019

Traditional autonomy: Understanding of space

Self-localisation, motion planning and navigation

focus on metric scale, accuracy, consistency

Traditional autonomy: Understanding of space

After a few hours ...

lack of focus on robustness

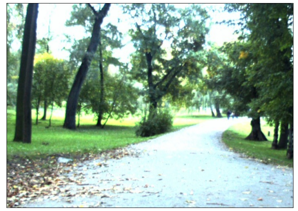
Long-term autonomous navigation

Target: 24/7 visual autonomous navigation
Environment: outdoor, forests, urban parks

Long-term operation: Environment changes

Oxford	*Churchill et al.*:	place-specific ‘experiences’
CMU	*Biswas et al.*:	static/dynamic separation
ETH	*Bürki et al.*:	map summarisation
QUT	*Sünderhauf et al.*:	appearance prediction
CTU	*Krajnik et al.*:	spatio-temporal models

Long-term operation: Environment changes



Oxford	*Churchill et al.*:	place-specific ‘experiences’
CMU	*Biswas et al.*:	static/dynamic separation
ETH	*Bürki et al.*:	map summarisation
QUT	*Sünderhauf et al.*:	appearance prediction
CTU	*Krajnik et al.*:	spatio-temporal models

Visual Navigation

Standard pipeline:

1. Extract image features,
2. find correspondences,
3. determine pose,
4. add new feats to map,
5. calculate movement.

Problems:

1. Feature deficiency,
2. environment change,
3. precision, complexity,
4. feature persistence,
5. real-time issues.

Visual Navigation

Standard pipeline:

1. Extract image features,
2. find correspondences,
3. determine pose,
4. add new feats to map,
5. calculate movement.

Problems:

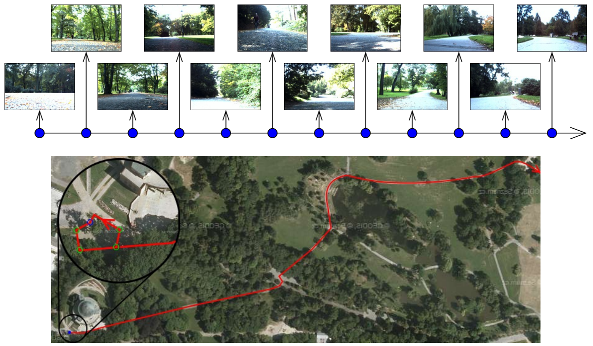
1. Feature deficiency,
2. environment change,
3. precision, complexity,
4. feature persistence,
5. real-time issues.

Teach and repeat navigation

- manually guide the robot along a given path,
- robot stores its odometry,
- robot stores image features,
- robot replays its odometry,
- while correcting its heading according to its visual memory.

Teach and repeat navigation

Image sequence indexed by position pics/along the learned path



Teach and repeat navigation

Image sequence indexed by position along the learned path

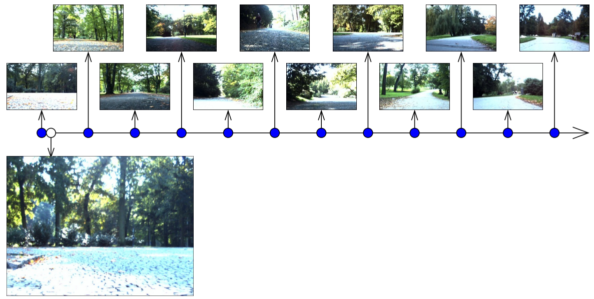


Image perceived by the robot during autonomous repeat

Teach and repeat navigation

Image sequence recorded during learning phase

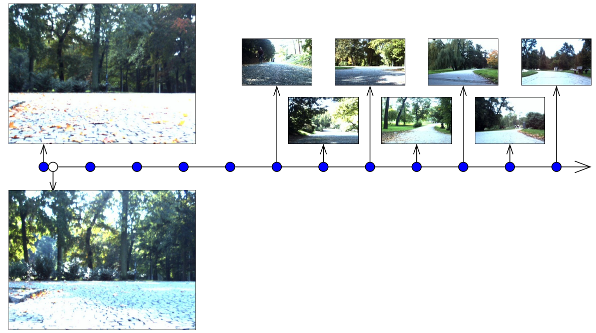


Image perceived by the robot during autonomous repeat

Teach and repeat navigation

Image sequence recorded during learning phase

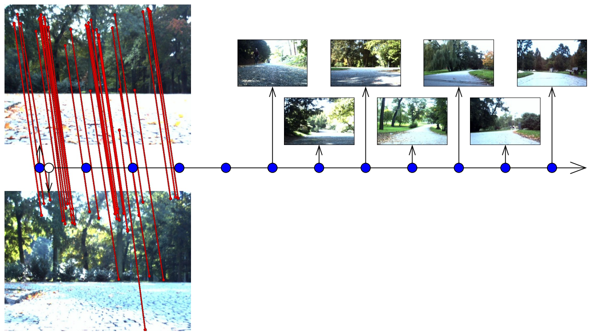


Image perceived by the robot during autonomous repeat

Teach and repeat navigation

Image sequence recorded during learning phase

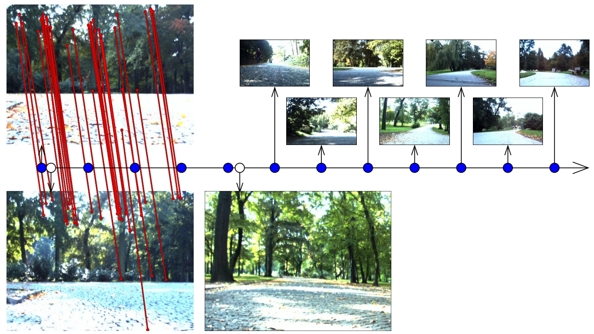


Image perceived by the robot during autonomous repeat

Teach and repeat navigation

Image sequence recorded during learning phase

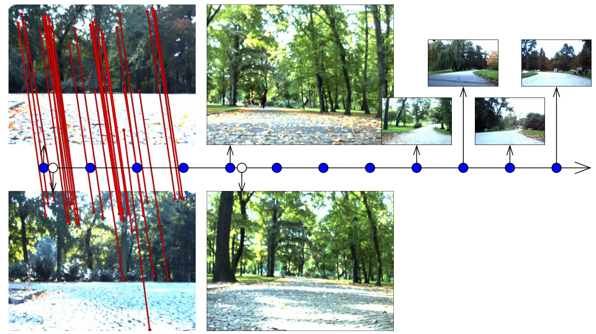


Image perceived by the robot during autonomous repeat

Teach and repeat navigation

Image sequence recorded during learning phase

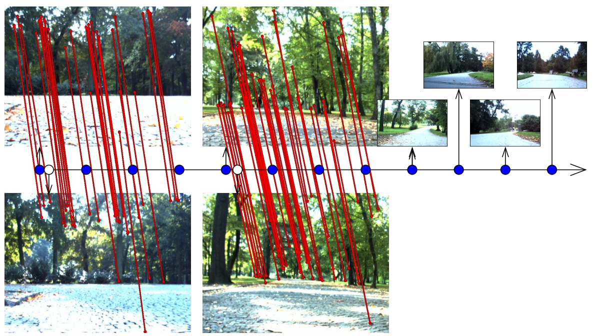


Image perceived by the robot during autonomous repeat

Teach and repeat navigation

Image sequence recorded during learning phase

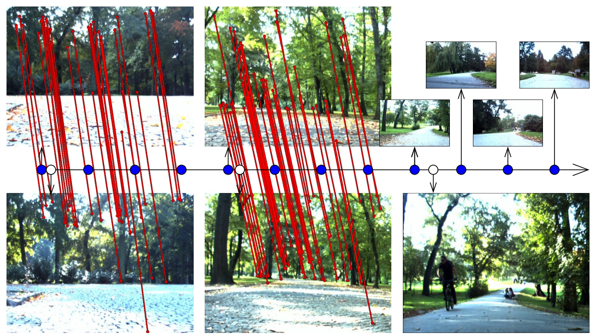


Image perceived by the robot during autonomous repeat

Teach and repeat navigation

Image sequence recorded during learning phase

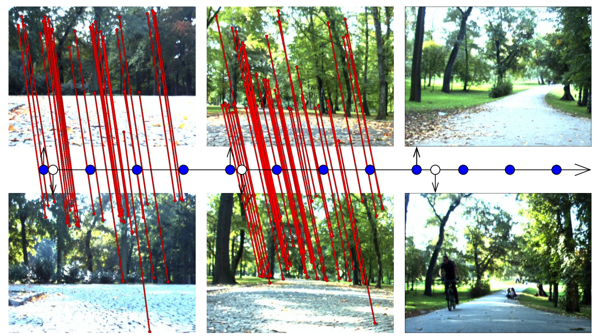


Image perceived by the robot during autonomous repeat

Teach and repeat navigation

Images stored in the local maps

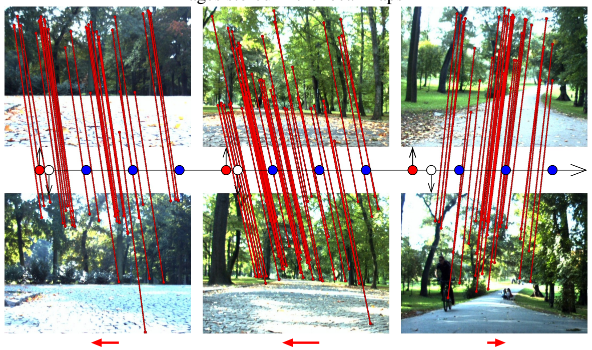


Image perceived by the robot during autonomous repeat

Teach and repeat navigation

Image sequence recorded during learning phase

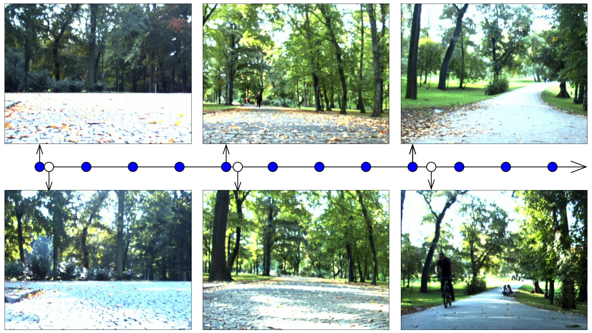


Image perceived by the robot during autonomous repeat

Not everything changes: learning stable features

Image sequence recorded during learning phase

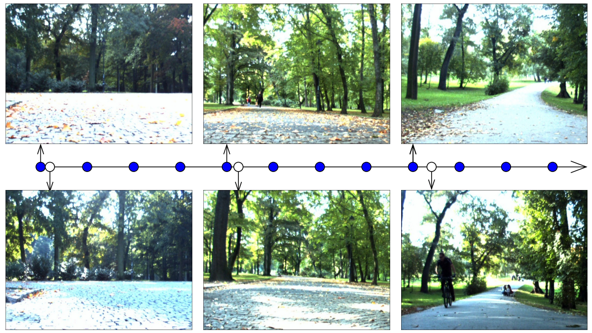


Image perceived by the robot during autonomous repeat

Not everything changes: learning stable features

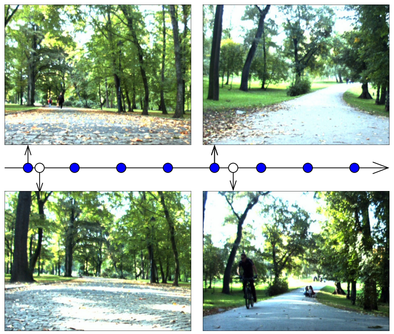
Typical features are robust to :

– viewpoint,
– scale,
– rotation,
– illumination.

Long–term teach–T&R requires robustness to :

– illumination,
– seasonal factors

Learned images



Perceived images

Not everything changes: learning stable features

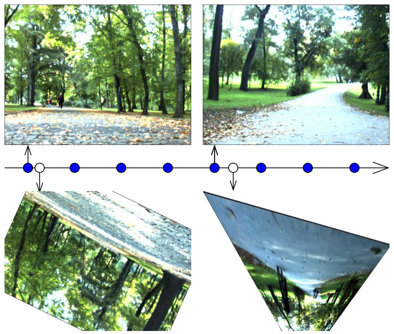
Typical features are robust to :

– viewpoint,
– scale,
– rotation,
– illumination.

Long–term teach–T&R requires robustness to :

– illumination,
– seasonal factors

Learned images



Perceived images

Not everything changes: learning stable features

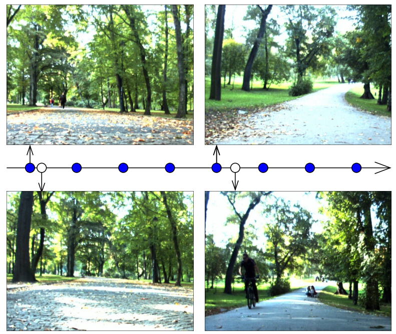
Typical features are robust to :

– viewpoint,
– scale,
– rotation,
– illumination.

Long–term teach–T&R requires robustness to :

– illumination,
– seasonal factors

Learned images



Perceived images

Not everything changes: learning stable features

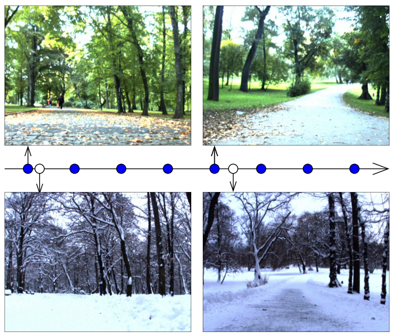
Typical features are robust to :

– viewpoint,
– scale,
– rotation,
– illumination.

Long–term teach–T&R requires robustness to :

– illumination,
– seasonal factors

Learned images



Perceived images

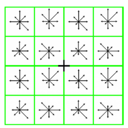
Not everything changes: learning stable features

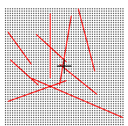
n-D vector describing local brightness gradients

- SIFT - typically best-performing, baseline method
- SURF - faster approximation of SIFT

Binary string describing brightness difference

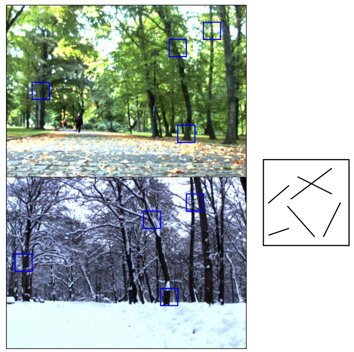
- BRIEF - binary comparisons, low viewpoint invariance,
- ORB - scale and rotation invariant BRIEF
- BRISK - scale and rotation invariant, symmetric positions
- GRIEF - BRIEF with comparison positions trained





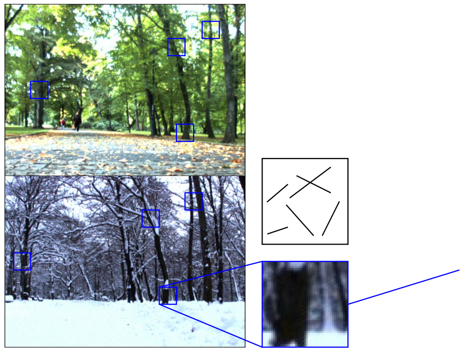
Not everything changes: learning stable features

BRIEF Binary Robust Independent Elementary Features



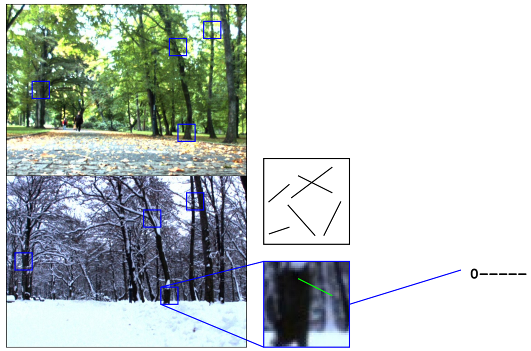
Not everything changes: learning stable features

BRIEF Binary Robust Independent Elementary Features



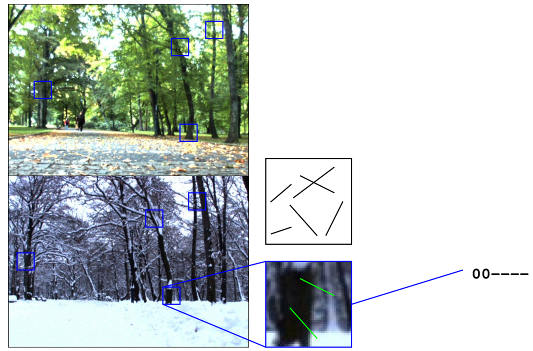
Not everything changes: learning stable features

BRIEF Binary Robust Independent Elementary Features



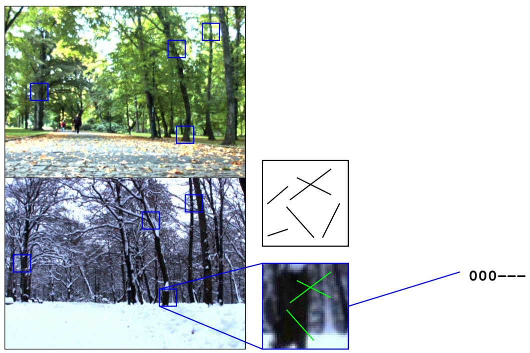
Not everything changes: learning stable features

BRIEF Binary Robust Independent Elementary Features



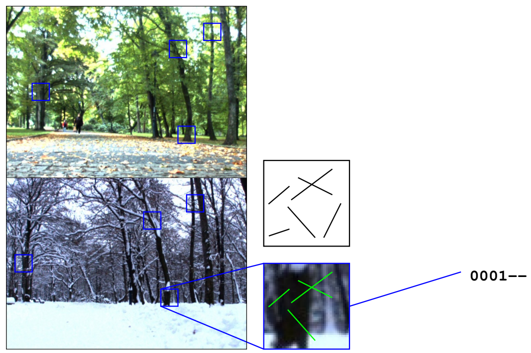
Not everything changes: learning stable features

BRIEF Binary Robust Independent Elementary Features



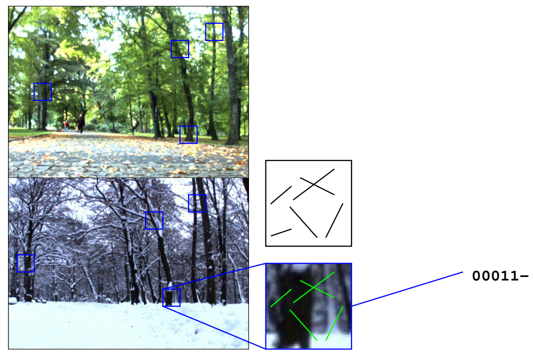
Not everything changes: learning stable features

BRIEF Binary Robust Independent Elementary Features



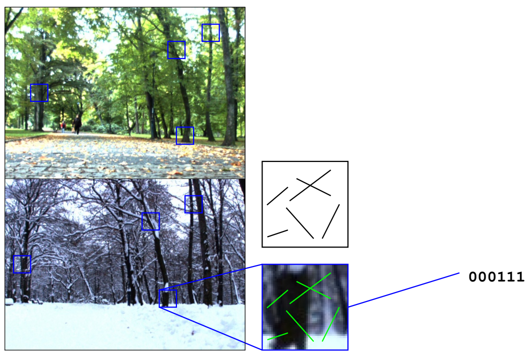
Not everything changes: learning stable features

BRIEF Binary Robust Independent Elementary Features



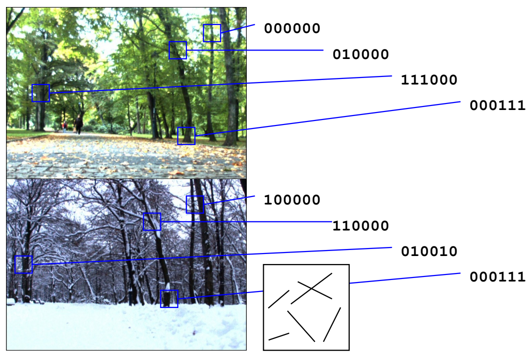
Not everything changes: learning stable features

BRIEF Binary Robust Independent Elementary Features



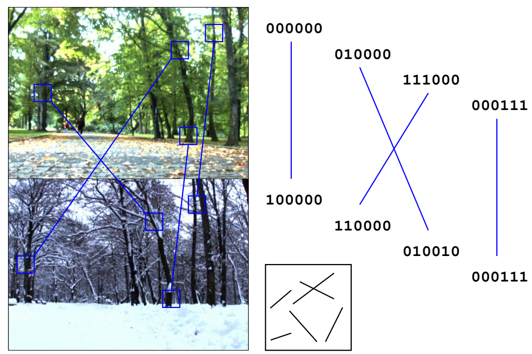
Not everything changes: learning stable features

Matching: Hamming distance of binary strings



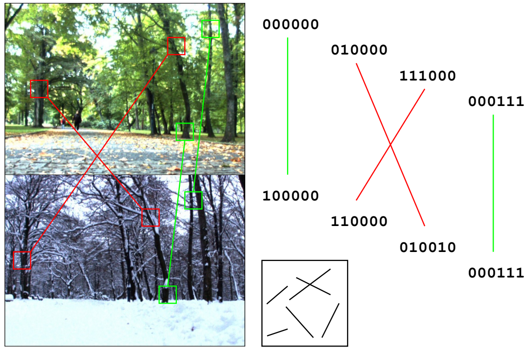
Not everything changes: learning stable features

Matching: Hamming distance of binary strings



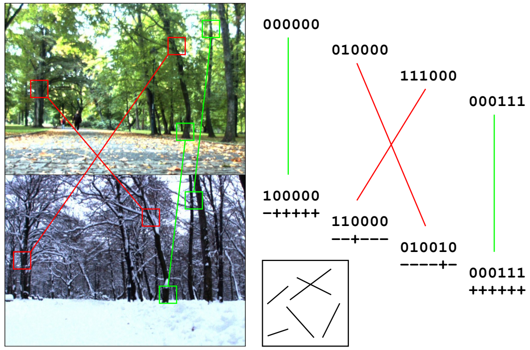
Not everything changes: learning stable features

Idea: rank the individual comparisons



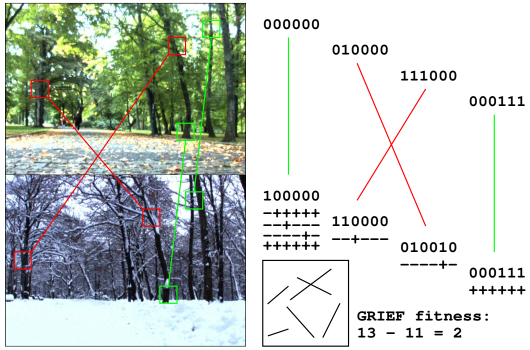
Not everything changes: learning stable features

Idea: rank the individual comparisons



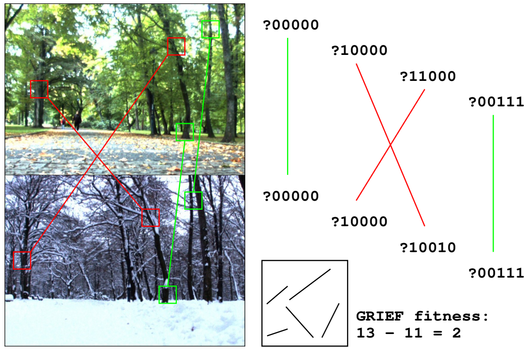
Not everything changes: learning stable features

Idea: rank the individual comparisons



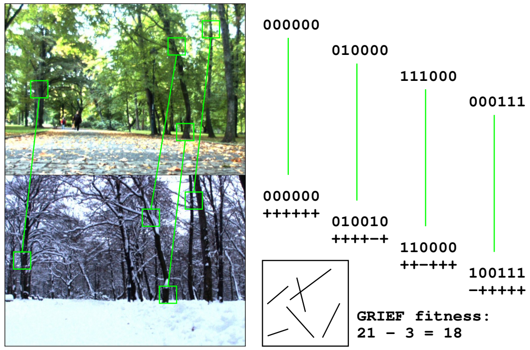
Not everything changes: learning stable features

Idea: rank the individual comparisons

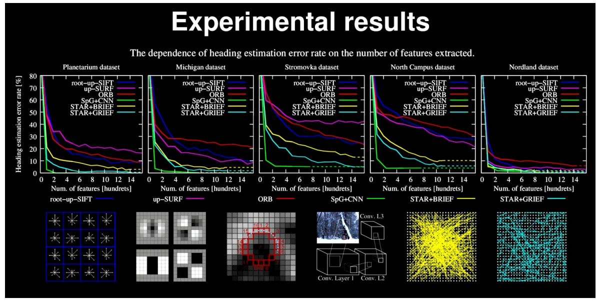


Not everything changes: learning stable features

Idea: rank the individual comparisons



Not everything changes: learning stable features



Not everything changes: learning stable features

Idea: **Not everything changes, and you can learn persistent features**

Training scheme is described in:

- T.Krajnik et al.: Image Features for Visual T&R Navigation in Changing Environments. RAS 2017.
- H.Nan et al.: Learning Place-And-Time-Dependent Binary Descriptors for Long-Term Visual Localization. ICRA 2018

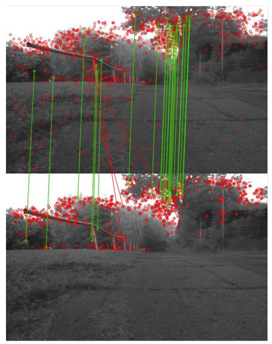
All code available at **http://github.com/gestom/grief**

Some changes are gradual, you can adapt your map

Two sources of uncertainty:
a) localisation, perception
b) environment change

Map management strategy:
- use the oldest map, robust to a)
- use the newest map, robust to b)

Some changes are gradual, you can adapt your map



Perform gradual map adaptation by ranking the map features:

- assign scores to features,
- check each feature match for geometrical consistency,
- increase score if consistent,
- decrease score if inconsistent,
- leave it not matched,
- remove the worst-scoring features,
- substitute with new ones.

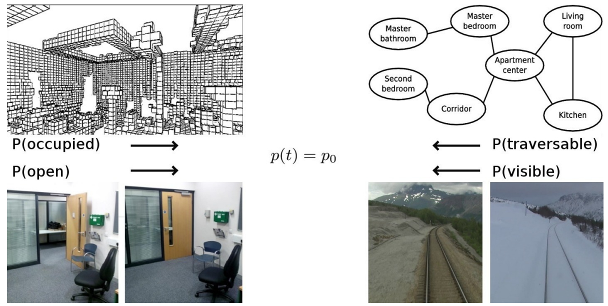
Some changes are repetitive, we can learn patterns

Oxford	*Churchill et al.*:	place-specific ‘experiences’
CMU	*Biswas et al.*:	static/dynamic separation
ETH	*Bürki et al.*:	map summarisation
QUT	*Sünderhauf et al.*:	appearance prediction

Some changes are repetitive, we can learn patterns

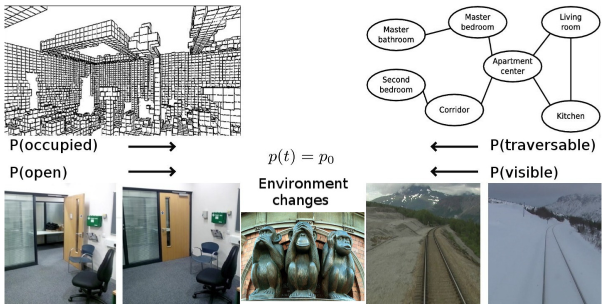
Oxford	*Churchill et al.*:	place-specific ‘experiences’
CMU	*Biswas et al.*:	static/dynamic separation
ETH	*Bürki et al.*:	map summarisation
QUT	*Sünderhauf et al.*:	appearance prediction

Some changes are repetitive, we can learn patterns



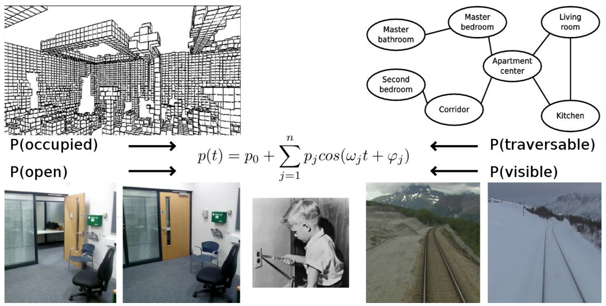
$p(t) = p_0$

Some changes are repetitive, we can learn patterns



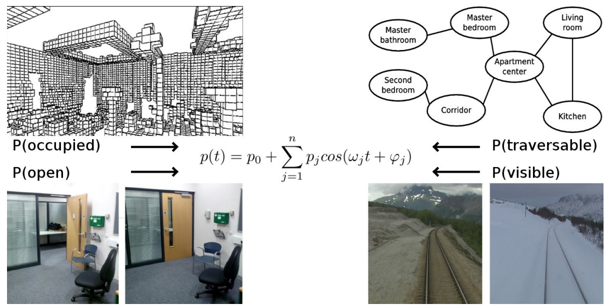
$p(t) = p_0$

Some changes are repetitive, we can learn patterns



$$p(t) = p_0 + \sum_{j=1}^{n} p_j cos(\omega_j t + \varphi_j)$$

Some changes are repetitive, we can learn patterns

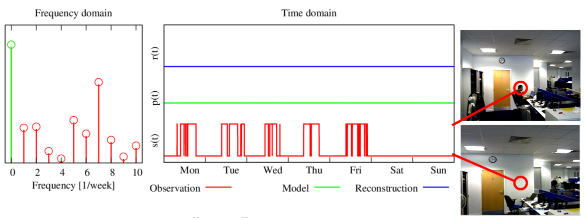


$$p(t) = p_0 + \sum_{j=1}^{n} p_j cos(\omega_j t + \varphi_j)$$

FreMEn: Frequency Map Enhancement

Continuous observation of an image feature

Static model:
$s'(t)$ matches the observations in 74% of cases

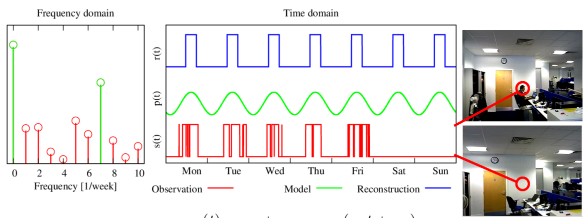


$p \quad = p_0$

FreMEn: Frequency Map Enhancement

Continuous observation of an image feature

Dynamic model with one periodic process:
$s'(t)$ matches the observations in 80% of cases

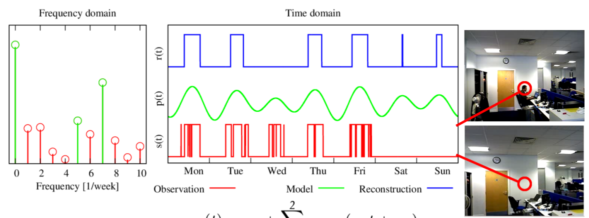


$p(t) = p_0 + \quad p_1 cos(\omega_1 t + \varphi_1)$

FreMEn: Frequency Map Enhancement

Continuous observation of an image feature

Dynamic model with two periodic processes:
$s'(t)$ matches the observations in 87% of cases

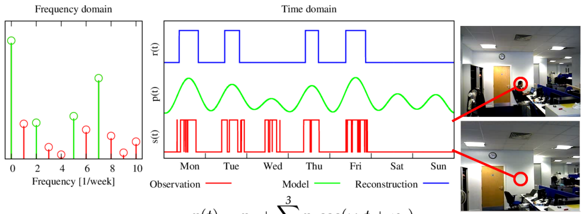


$$p(t) = p_0 + \sum_{j=1}^{2} p_j cos(\omega_j t + \varphi_j)$$

FreMEn: Frequency Map Enhancement

Continuous observation of an image feature

Dynamic model with n periodic processes:
$s'(t)$ matches the observations in $90\% - 95\%$ of cases

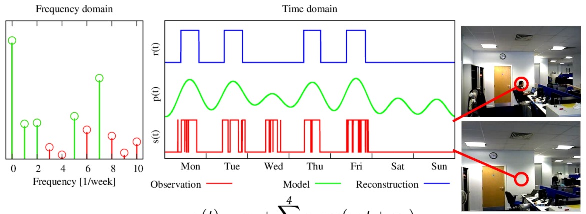


$$p(t) = p_0 + \sum_{j=1}^{3} p_j cos(\omega_j t + \varphi_j)$$

FreMEn: Frequency Map Enhancement

Continuous observation of an image feature

Dynamic model with n periodic processes:
$s'(t)$ matches the observations in $90\% - 95\%$ of cases

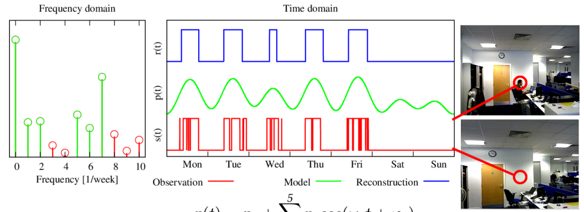


$$p(t) = p_0 + \sum_{j=1}^{4} p_j cos(\omega_j t + \varphi_j)$$

FreMEn: Frequency Map Enhancement

Continuous observation of an image feature

Dynamic model with n periodic processes:
$s'(t)$ matches the observations in $90\% - 95\%$ of cases

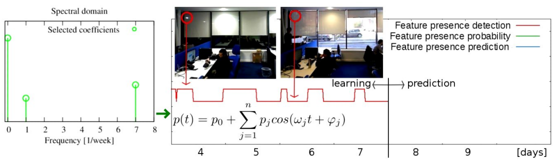


$$p(t) = p_0 + \sum_{j=1}^{5} p_j cos(\omega_j t + \varphi_j)$$

Frequency Map Enhanement principle

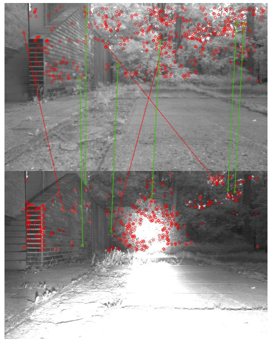
Frequency Map Enhancement (FreMEn)

Represents uncertainty of binary environment states in the frequency domain.
Can predict environment appearance at a given time.

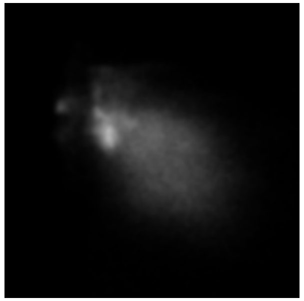


Krajnik, Fentanes, Cielniak, Dondrup, Duckett: **A frequency-based approach to long-term robotic mapping,** ICRA 2014

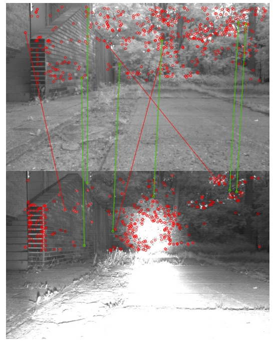
Some changes are consistent, and indicate objects



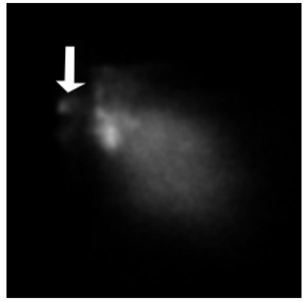
Form an accumulator



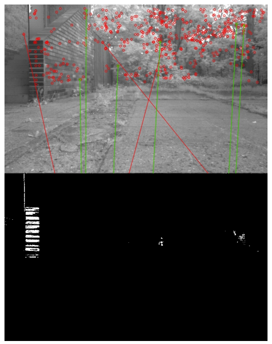
Some changes are consistent, and indicate objects



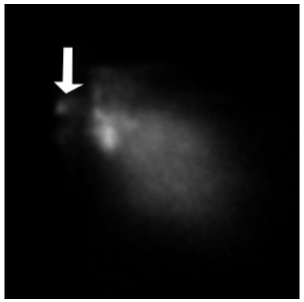
Find local maxima



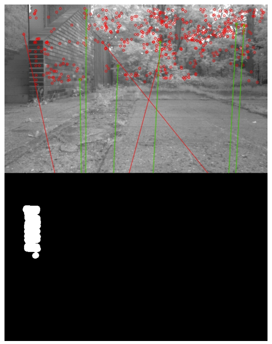
Some changes are consistent, and indicate objects



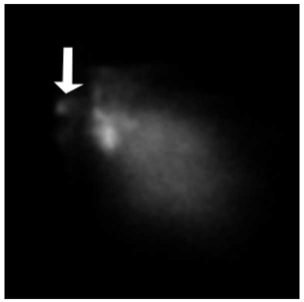
Reproject to images



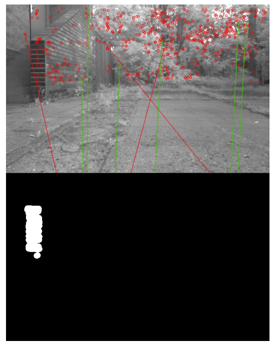
Some changes are consistent, and indicate objects



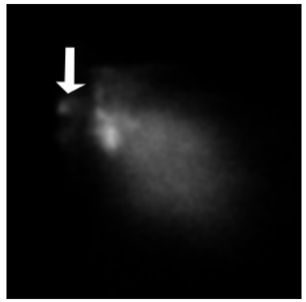
Morphological ops.



Some changes are consistent, and indicate objects



Feed masks to Mask-RCNN



What to remember from the first part of the lecture

Changes are source of valuable information [1,2].

Some changes are:

- **describable** and you can learn robust features [3],
- **gradual** and you can adapt to then [4],
- **repetitive** and you can learn their patterns [5],
- **consistent** and you can use them for auto-annotation [6],

References:
[1] Krajník et al.: Chronorobotics In IJCRAI 19 download link
[2] Kunze et al.: Artificial Intelligence for Long-term Autonomy IEEE RA-L 19
[3] Krajník et al.: Image Features for Visual T&R Navigation in Changing Environments RAS 17
[4] Halodova et al.: Predictive and adaptive maps for long-term visual navigation In IROS 19
[5] Krajník et al.: Fremen: Frequency map enhancement for LTA in changing environments IEEE T-RO 17
[6] Peconkova et al.: Unsupervised Learning of Landmarks for Vis.Nav. ... In PAIR 19

We strongly suggest to read [1] for an overview of the topic.

What to remember from the first part of the lecture

Changes are source of valuable information [1,2].

Some changes are:

- **describable** and you can learn robust features [3],
- **gradual** and you can adapt to then [4],
- **repetitive** and you can learn their patterns [5],
- **consistent** and you can use them for auto-annotation [6],

Download:

BearNav system:	http://bearnav.eu
GRIEF Image features:	http://github.com/gestom/grief
Frequency Map Enhancement:	http://fremen.uk

Artificial Intelligence in Robotics
Mapping of changing environments
-
Part II: Spatio-temporal representations of the environment dynamics

Tom Krajník
Czech Technical University in Prague

Dec 2019

Traditional autonomy: Understanding of space

Self-localisation, motion planning and navigation

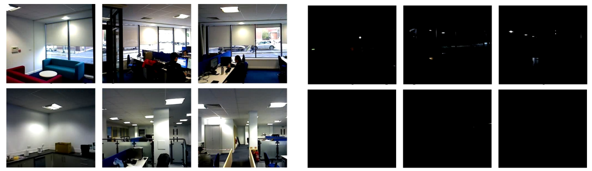
focus on metric scale, accuracy, consistency

Traditional autonomy: Understanding of space

After a few hours ...

lack of focus on robustness

Mid-term autonomy: Understanding that environment changes



Memorising and suppressing changes:

Oxford	*Churchill et al.*:	experience-based approach
Örebro	*Lowry et al.*:	condition-invariant appearance
CMU	*Biswas et al.*:	static/dynamic separation
ETH	*Bürki et al.*:	map summarisation

Mid-term autonomy: Understanding that environment changes

Memorising and suppressing changes:

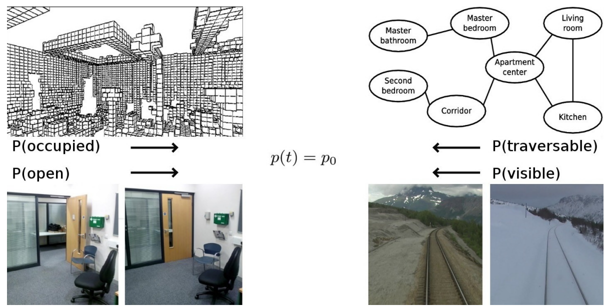
Oxford	*Churchill et al.*:	experience-based approach
Örebro	*Lowry et al.*:	condition-invariant appearance
CMU	*Biswas et al.*:	static/dynamic separation
ETH	*Bürki et al.*:	map summarisation

Long-term autonomy: Understanding how environment changes

Obtaining information from the changes observed:

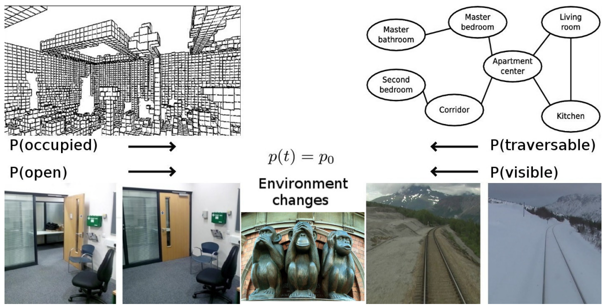
long-term operation	$\rightarrow$	observation of changes
observed changes	$\rightarrow$	spatio-temporal models
spatio-temporal model	$\rightarrow$	prediction of future states
prediction of future states	$\rightarrow$	improved long-term operation

Towards Spatio-Temporal Domain Modeling

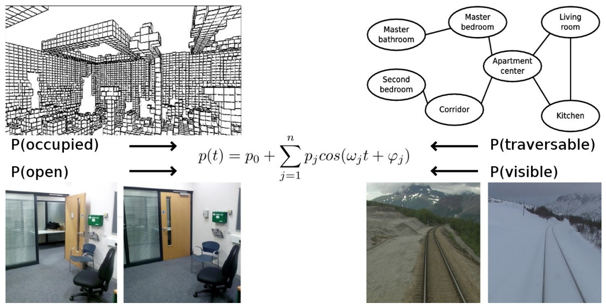


$p(t) = p_0$

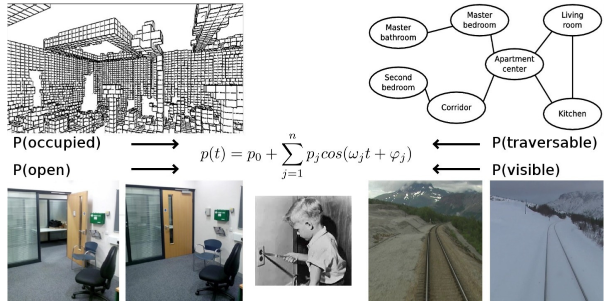
Towards Spatio-Temporal Domain Modeling



Towards Spatio-Temporal Domain Modeling



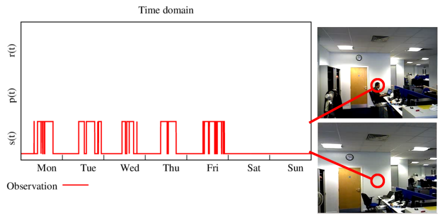
Towards Spatio-Temporal Domain Modeling



Example: Modeling a single state

Week-long model of a single feature

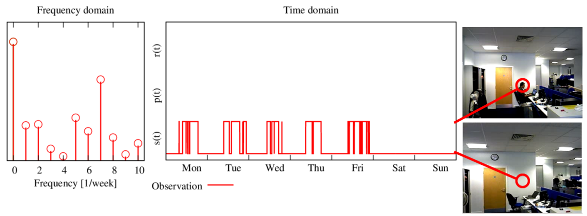
Data gathering:
Establish a binary function of time $s(t)$



Example: Modeling a single state

Week-long model of a single feature

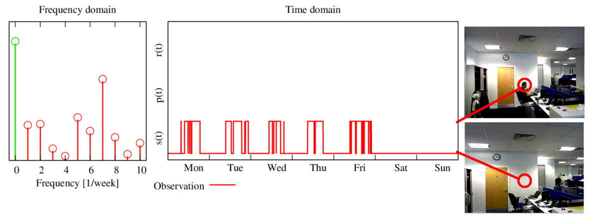
Fourier analysis:
Calculate frequency spectrum of $s(t)$



Example: Modeling a single state

Week-long model of a single feature

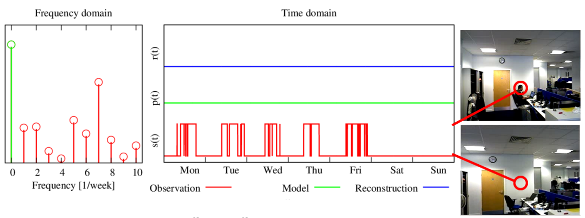
Component selection:
Select the most prominent spectral component(s)



Example: Modeling a single state

Week-long model of a single feature

Static model with one component with zero frequency:
$s'(t)$ matches the observations in 74% of cases

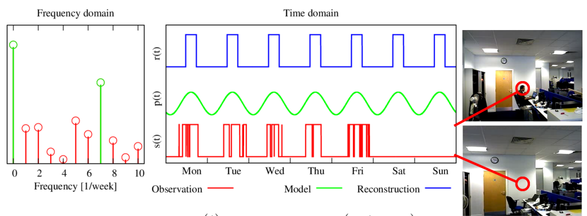


$$p \quad = p_0$$

Example: Modeling a single state

Week-long model of a single feature

Dynamic model with one periodic process:
$s'(t)$ matches the observations in 80% of cases

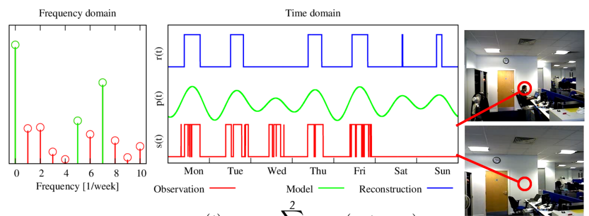


$$p(t) = p_0 + \quad p_1 cos(\omega_1 t + \varphi_1)$$

Example: Modeling a single state

Week-long model of a single feature

Dynamic model with two periodic processes:
$s'(t)$ matches the observations in 87% of cases

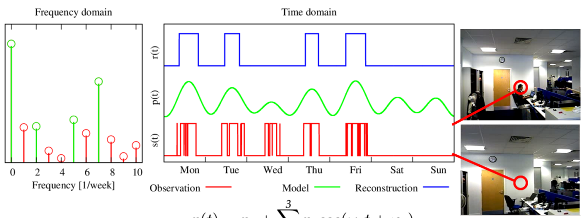


$$p(t) = p_0 + \sum_{j=1}^{2} p_j cos(\omega_j t + \varphi_j)$$

Example: Modeling a single state

Week-long model of a single feature

Dynamic model with n periodic processes:
$s'(t)$ matches the observations in $90\% - 95\%$ of cases

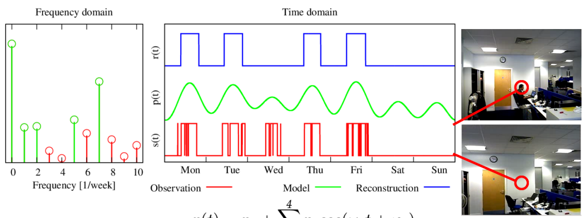


$$p(t) = p_0 + \sum_{j=1}^{3} p_j cos(\omega_j t + \varphi_j)$$

Example: Modeling a single state

Week-long model of a single feature

Dynamic model with n periodic processes:
$s'(t)$ matches the observations in $90\% - 95\%$ of cases

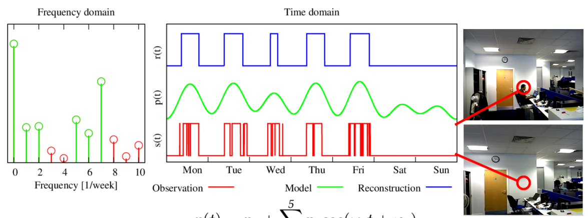


$$p(t) = p_0 + \sum_{j=1}^{4} p_j cos(\omega_j t + \varphi_j)$$

Example: Modeling a single state

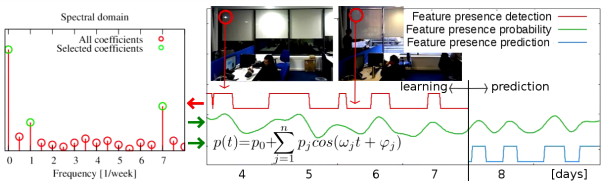
Week-long model of a single feature

Dynamic model with n periodic processes:
$s'(t)$ matches the observations in $90\% - 95\%$ of cases



$$p(t) = p_0 + \sum_{j=1}^{5} p_j cos(\omega_j t + \varphi_j)$$

Video 1: Feature-based topological localization



Frequency-enhanced feature map for visual localisation:

- The observations of image feature visibility (centre,red) are transferred to the spectral domain (left).
- The most prominent components of the model (left,green) constitute an analytic expression (centre,bottom) that represents the probability of the feature being visible at a given time (green).
- This is used to predict the feature visibility at a time when the robot performs self-localisation (blue).

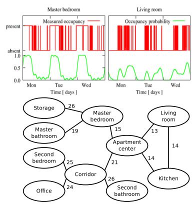
Object search scenario

Task: Find a person in shortest time possible.
Topological map, spectral-based model of room occupancies.

Spatial:
- 1 person
- 9 locations

Temporal:
- 16 weeks
- every minute

Figure: The CASAS-Aruba environment.



Object search scenario

Task: Find a person in shortest time possible.
Topological map, spectral-based model of room occupancies.

Spatial:
- 1 person
- 9 locations

Temporal:
- 16 weeks
- every minute

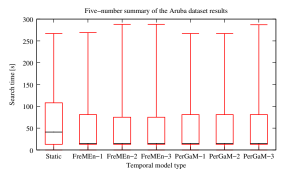
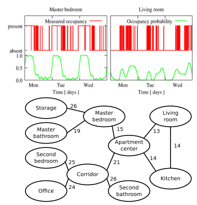


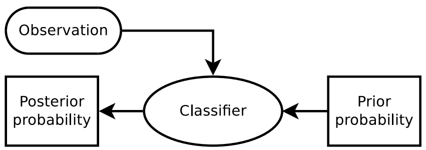
Figure: Time to find a person in the 'Aruba' flat.



Temporal context for activity recognition

Task: Classify person activity.

$$p(activity|observation) = \frac{p(observation|activity)}{p(observation)} p(activity)$$

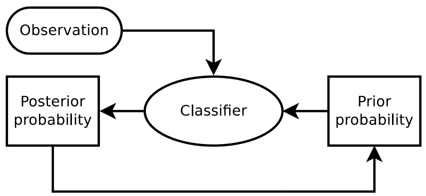


Classification pipeline.

Temporal context for activity recognition

Task: Classify person activity.

$$p(activity|observation) = \frac{p(observation|activity)}{p(observation)} p(activity)$$

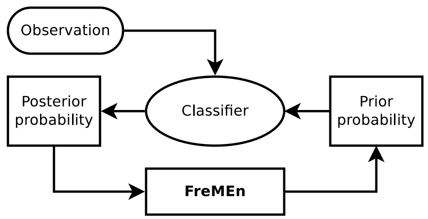


Classification pipeline + learning priors.

Temporal context for activity recognition

Task: Classify person activity.

$$p(activity|observation, t) = \frac{p(observation|activity)}{p(observation, t)} p(activity, t)$$



Classification pipeline + temporal priors.

Temporal context for activity recognition

Task: Classify person activity.
Use FreMEn-aided temporal models as priors.

$p(activity, t|observation) \sim p(observation|activity)p(activity, t)$

Household:
- 9 locations
- 12 activities

Office:
- 10 locations
- 10 activities

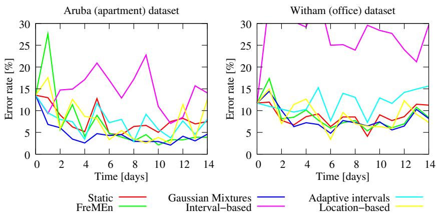


Figure: Error rate of activity recognition.

SPEctral Robotic Mapping

The approach allows for
- conversion of static models into dynamic ones,
- environment state and appearance prediction.

However, it requires regular and frequent observations,

$$S(k) = \frac{1}{N}\sum_{n=1}^{N} s(nT)e^{-2\pi jkn/N} \quad k \in \mathbb{N}$$

- which results in long, tedious and brittle learning,
- after which the model cannot be updated.

FREquency Map ENhancement

The approach allows for
- conversion of static models into dynamic ones,
- environment state and appearance prediction.

Allowing sparse and non-uniform observations,

$$S(\omega_k) = \frac{1}{N}\sum_{n=1}^{N} s(t_n)e^{-j\omega_k t_n} \quad \omega_k \in \Omega$$

- means that we can deal with irregular observations,
- and learn incrementally during operation.

Frequency Map Enhancement (FreMEn)

Can build spatio-temporal models **incrementally** from **sparse** and **irregular** observations. Allows **on-the-fly** learning.

Addition of a new measurement:

$\mu \leftarrow \frac{1}{n+1}(n\mu + s(t))$,			mean probability
$\alpha_k \leftarrow \frac{1}{n+1}(n\alpha_k + s(t)\,e^{-jt\omega_k})$	$\forall \omega_k \in \omega$,		state spectrum
$\beta_k \leftarrow \frac{1}{n+1}(n\beta_k + e^{-jt\omega_k})$	$\forall \omega_k \in \omega$,		observation spectrum
$n \leftarrow n+1$,			num of observations

Performing predictions:

$\gamma_k \leftarrow \alpha_k - \mu\,\beta_k$	predictive spectrum
$\gamma_{1..m} \leftarrow \operatorname{argmax}\lvert\gamma_k\rvert$	m components γ_k with highest abs. value
$p(t) = \mu + \sum_{j=1}^{m}\lvert\gamma_j\rvert\cos(\omega_j t + \arg(\gamma_j))$	actual prediction

Topological path planning

Decide the best time to navigate to a particular location.
Topological map with FreMEn edge traversability.

Spatial:
- 14 nodes
- 26 edges

Temporal:
- two months
- $\sim$ 10$\times$ per day

Nav. success rate:
- Static: 60%
- FreMEn: 90%

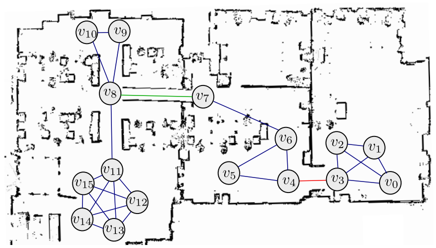


Figure: Topological/metric map overlay of the LCAS offices.

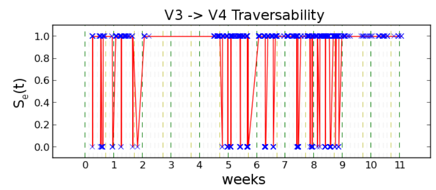
Topological path planning

Decide the best time to navigate to a particular location.
Topological map with FreMEn edge traversability.

Spatial:
- 14 nodes
- 26 edges

Temporal:
- two months
- $\sim$ 10$\times$ per day

Nav. success rate:
- Static: 60%
- FreMEn: 90%



Edge V3->V4: Navigation success over time.

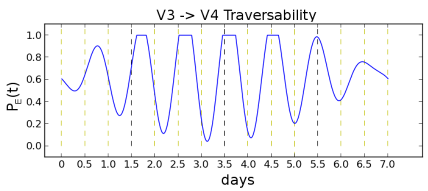
Topological path planning

Decide the best time to navigate to a particular location.
Topological map with FreMEn edge traversability.

Spatial:
- 14 nodes
- 26 edges

Temporal:
- two months
- $\sim$ 10$\times$ per day

Nav. success rate:
- Static: 60%
- FreMEn: 90%



Edge V3->V4: FreMEn model for one week.

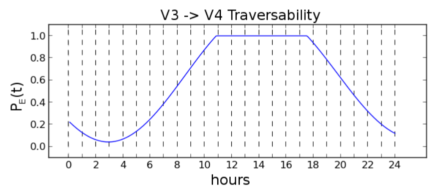
Topological path planning

Decide the best time to navigate to a particular location.
Topological map with FreMEn edge traversability.

Spatial:
- 14 nodes
- 26 edges

Temporal:
- two months
- $\sim$ 10$\times$ per day

Nav. success rate:
- Static: 60%
- FreMEn: 90%

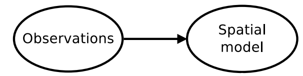


Edge V3->V4: FreMEn model for Thursday

Towards Spatio-Temporal Exploration

Create accurate spatial models.

Mapping pipeline:

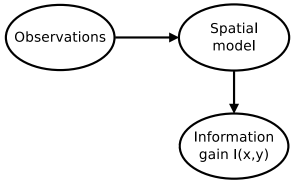


Observations gathered during routine operation

Towards Spatio-Temporal Exploration

Create accurate spatial models.

Spatial exploration pipeline:

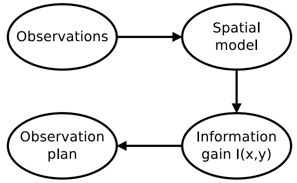


Robot decides **where** to perform observations

Towards Spatio-Temporal Exploration

Create accurate spatial models.

Spatial exploration pipeline:

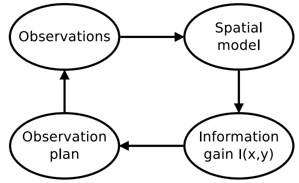


Robot decides **where** to perform observations

Towards Spatio-Temporal Exploration

Create accurate spatial models.

Spatial exploration pipeline:

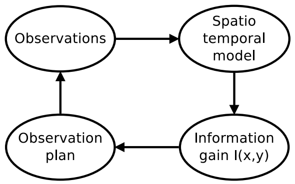


Robot decides **where** to perform observations

Towards Spatio-Temporal Exploration

Create and maintain accurate spatial-temporal models.

Spatio-temporal exploration pipeline:

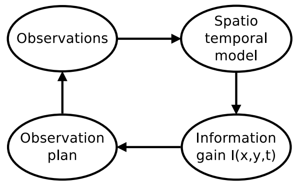


Robot decides **where** and **when** to perform observations

Towards Spatio-Temporal Exploration

Create and maintain accurate spatial-temporal models.

Spatio-temporal exploration pipeline:

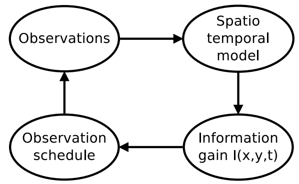


Robot decides **where** and **when** to perform observations

Towards Spatio-Temporal Exploration

Create and maintain accurate spatial-temporal models.

Spatio-temporal exploration pipeline:

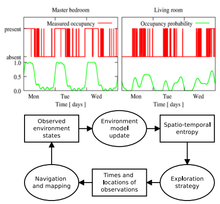


Robot decides **where** and **when** to perform observations

Information-theoretic spatio-temporal exploration

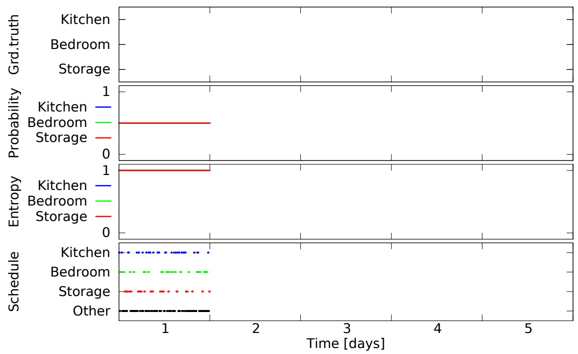
Create and maintain accurate spatio-temporal models.
Decide **where** and **when** to perform observations
Probability p(t) → Entropy H(t) → Prob. of observation o(t)
'Next Best Time and Location'

Figure: The CASAS-Aruba environment.



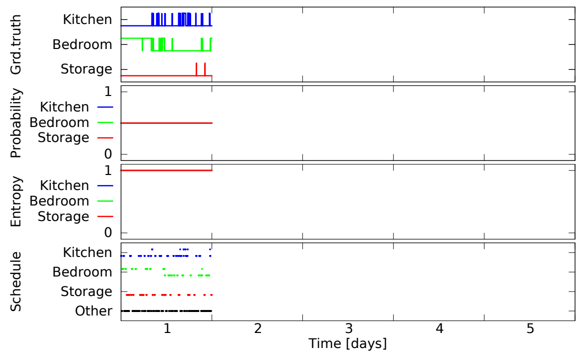
Spatio-temporal exploration

Decide **where** and **when** to go to make observations.
Spatio-temporal entropy + information-gain-based methods.



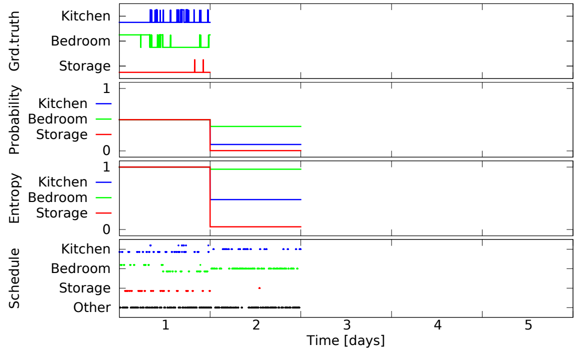
Spatio-temporal exploration

Decide **where** and **when** to go to make observations.
Spatio-temporal entropy + information-gain-based methods.



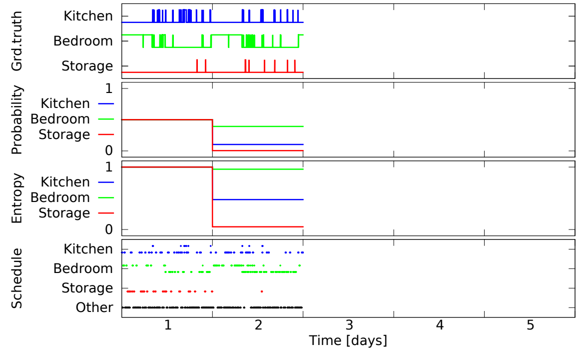
Spatio-temporal exploration

Decide **where** and **when** to go to make observations.
Spatio-temporal entropy + information-gain-based methods.



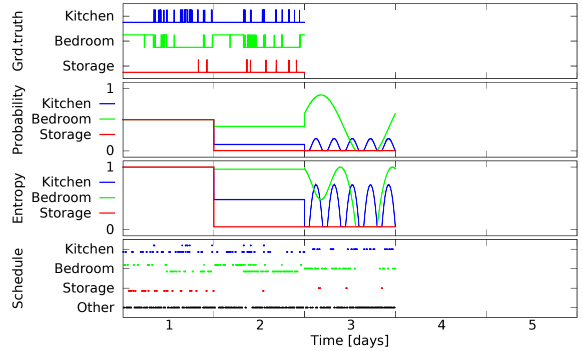
Spatio-temporal exploration

Decide **where** and **when** to go to make observations.
Spatio-temporal entropy + information-gain-based methods.



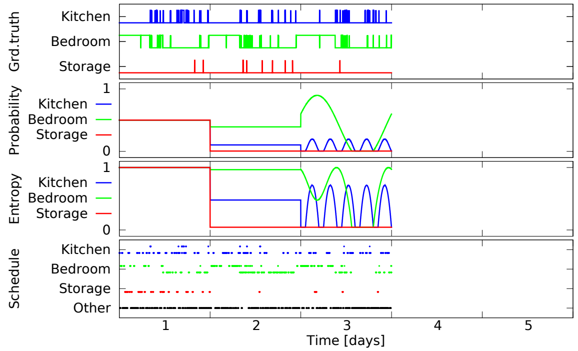
Spatio-temporal exploration

Decide **where** and **when** to go to make observations.
Spatio-temporal entropy + information-gain-based methods.



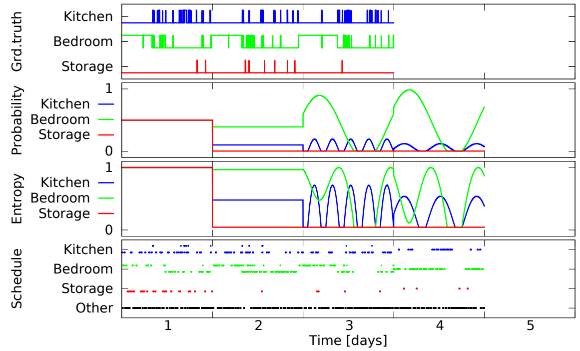
Spatio-temporal exploration

Decide **where** and **when** to go to make observations.
Spatio-temporal entropy + information-gain-based methods.



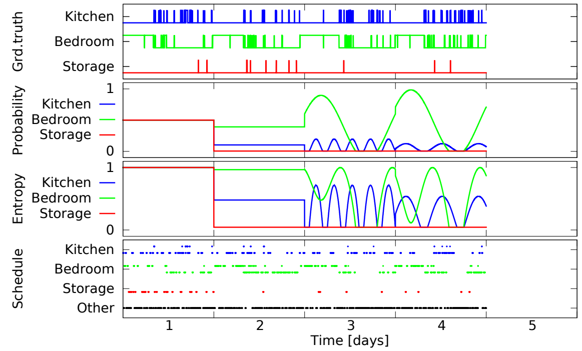
Spatio-temporal exploration

Decide **where** and **when** to go to make observations.
Spatio-temporal entropy + information-gain-based methods.



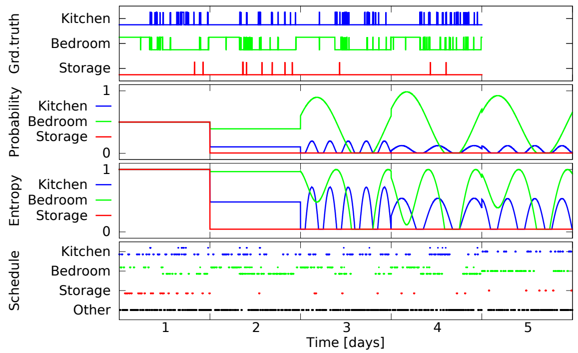
Spatio-temporal exploration

Decide **where** and **when** to go to make observations.
Spatio-temporal entropy + information-gain-based methods.



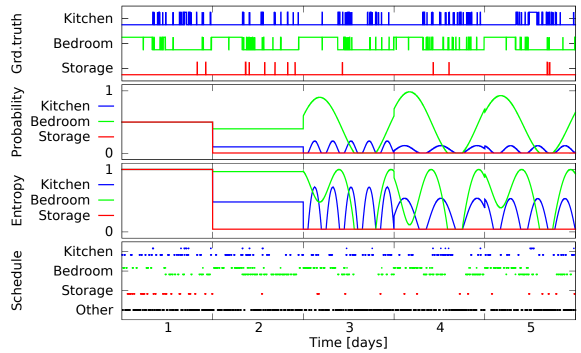
Spatio-temporal exploration

Decide **where** and **when** to go to make observations.
Spatio-temporal entropy + information-gain-based methods.



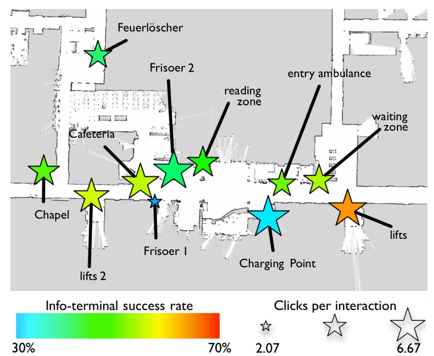
Spatio-temporal exploration

Decide **where** and **when** to go to make observations.
Spatio-temporal entropy + information-gain-based methods.



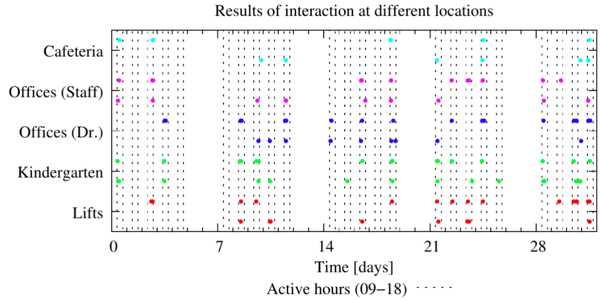
Mobile Infoterminal - exploration/exploitation

Decide the best time and location to provide an info-terminal service in a hospital. Maximise number of interactions.



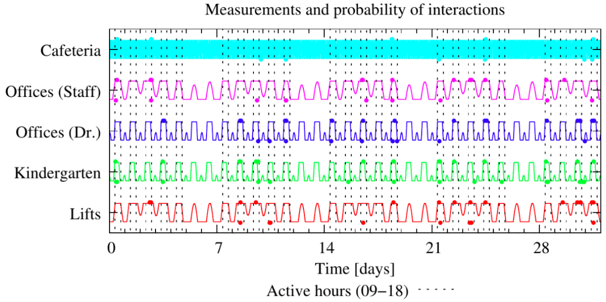
Mobile Infoterminal - exploration/exploitation

Decide the best time and location to provide an info-terminal service in a hospital. Maximise number of interactions.



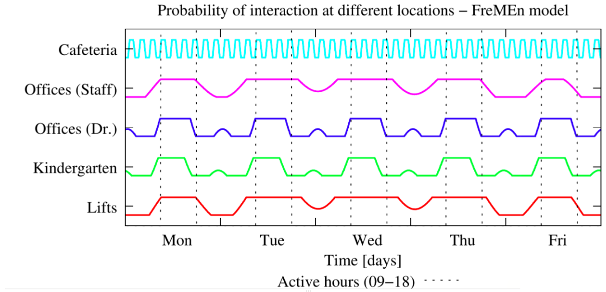
Mobile Infoterminal - exploration/exploitation

Decide the best time and location to provide an info-terminal service in a hospital. Maximise number of interactions.



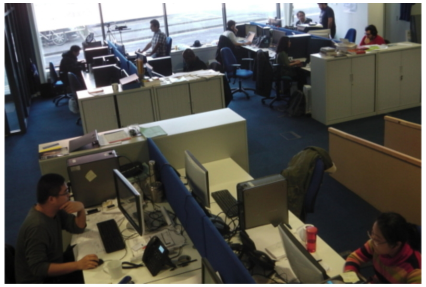
Mobile Infoterminal - exploration/exploitation

Decide the best time and location to provide an info-terminal service in a hospital. Maximise number of interactions.



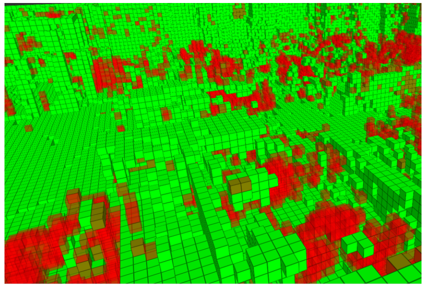
4D Spatio-Temporal Exploration

Spatio-temporal Information-driven Next Best View.
FreMEn 3D grid + spatio-temporal entropy + next best path



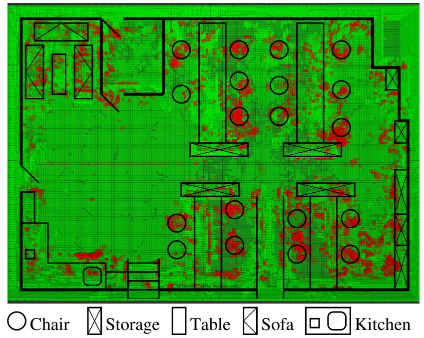
4D Spatio-Temporal Exploration

Spatio-temporal Information-driven Next Best View.
FreMEn 3D grid + spatio-temporal entropy + next best path

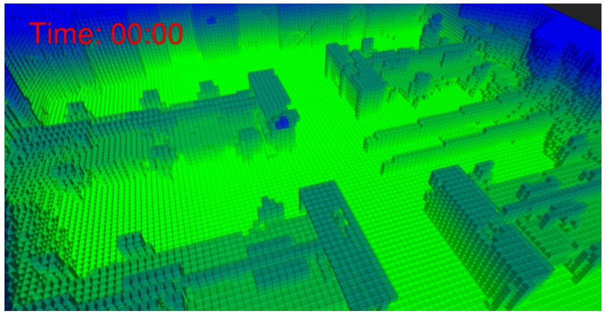


4D Spatio-Temporal Exploration

Spatio-temporal Information-driven Next Best View.
FreMEn 3D grid + spatio-temporal entropy + next best path



Video 2: 4D maps build by a FreMEn-based exploration system



To predict the grid state for a particular time, each cell contains a temporal model. Approx. 10^6 cells resulted in memory issues.

From discrete to continuous models

FreMEn is powefull, but it can model only:

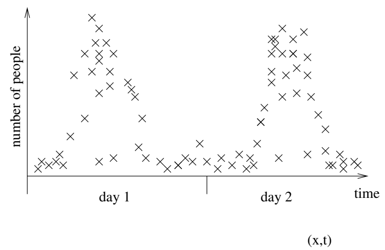
- Bernoulli distributions, i.e. probabilities of binary states,
- discrete models, which might be memory inefficient,
- events with durations comparable to the period length,
- independent components

Warped Hypertime can represent:

- arbitrary distributions, e.g. number of people, robot velocity,
- memory-efficient, continous models,
- arbitrarily long events and changes,
- respects spatio-temporal continuity

Warped Hypertime - continuous models

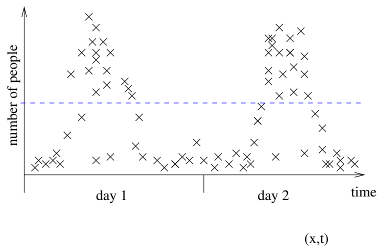
Example: modelling the number of people within a given area



Observe data over time,

Warped Hypertime - continuous models

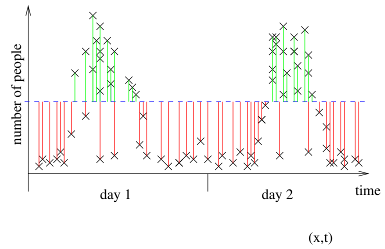
Example: modelling the number of people within a given area



establish some time-unavare model,

Warped Hypertime - continuous models

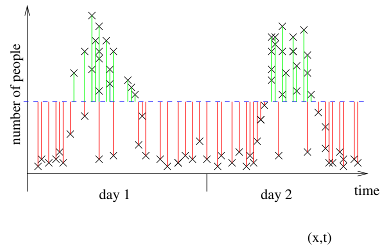
Example: modelling the number of people within a given area



establish model error over time $\epsilon(t)$,

Warped Hypertime - continuous models

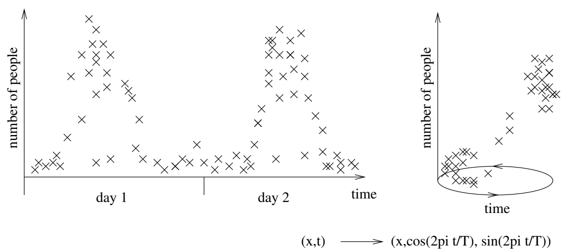
Example: modelling the number of people within a given area



use FreMEn to find periodicity T in $\epsilon(t)$,

Warped Hypertime - continuous models

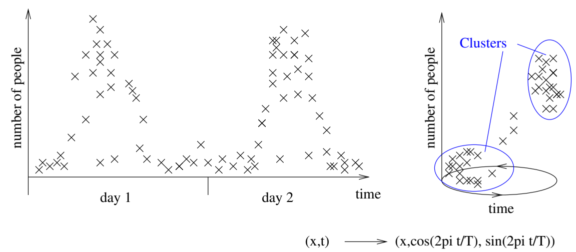
Example: modelling the number of people within a given area



project 1D time in ‘2D warped hypertime’

Warped Hypertime - continuous models

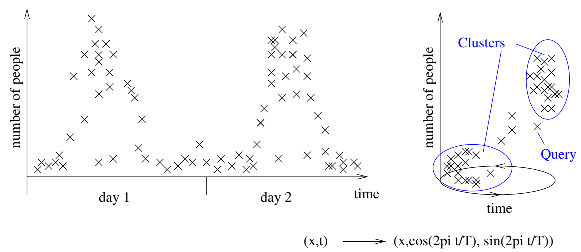
Example: modelling the number of people within a given area



create a model of the data distribution (k-means, EM-GMM),

Warped Hypertime - continuous models

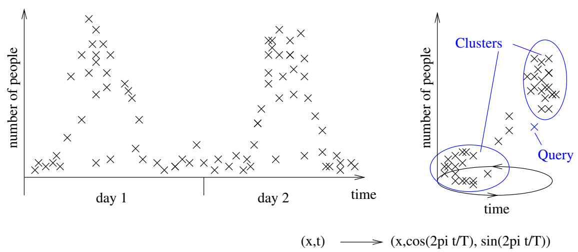
Example: modelling the number of people within a given area



repeat (add 2 temporal dimensions per observed periodicity).

Warped Hypertime - continuous models

Example: modelling the number of people within a given area



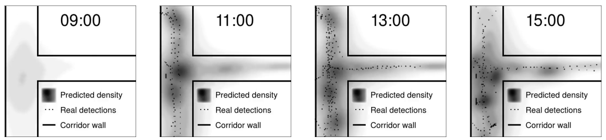
Predict probability of future distribution of x.

Video 3: Predicting future presence of people

Predicting future density of pedestrians

- pedestrian tracker detects people at x, y at time t,
- Warped Hypertime adds 4 temporal dimensions
- $[x, y] \rightarrow [x, y, \cos(2\pi\frac{t}{T_0}), \sin(2\pi\frac{t}{T_0}), \cos(2\pi\frac{t}{T_1}), \sin(2\pi\frac{t}{T_1})]$
- 9 clusters characterise spatio-temporal density of people

Prediction of people density

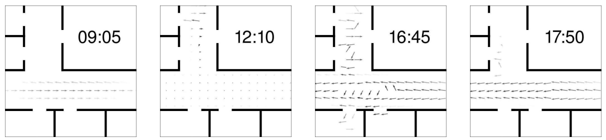


Video 4: Predicting future pedestrian flows

Prediction of pedestrian flows

- pedestrian tracker provides x, y, ϕ, v at time t,
- Warped Hypertime adds 4 temporal dimensions
- $[x, y, \phi, v, \cos(2\pi\frac{t}{T_0}), \sin(2\pi\frac{t}{T_0}), \cos(2\pi\frac{t}{T_1}), \sin(2\pi\frac{t}{T_1})]$
- 13 clusters characterise intensity, velocity and directions

Prediction of pedestrian flows



What to remember from the lecture

- explicit representation of the temporal domain improves the efficiency of robot operation in long-term scenarios [1]
- one can convert static representations into models that represent how the environment changes over time,
- by modeling the uncertainty in the spectral domain [2],
- by modeling the time in a multi-dimensional space [3].

References:

- [1] Krajník et al.: Chronorobotics: ... In IJCRAI 2020
- [2] Krajník et al.: FreMEn: ... IEEE T-RO 2017
- [3] Krajník et al.: Warped Hypertime ... IEEE RAL 2019

We strongly suggest to read [1] for an overview of the topic.