Incorporating Manhattan and Piecewise Planar Priors in RGB Monocular SLAM

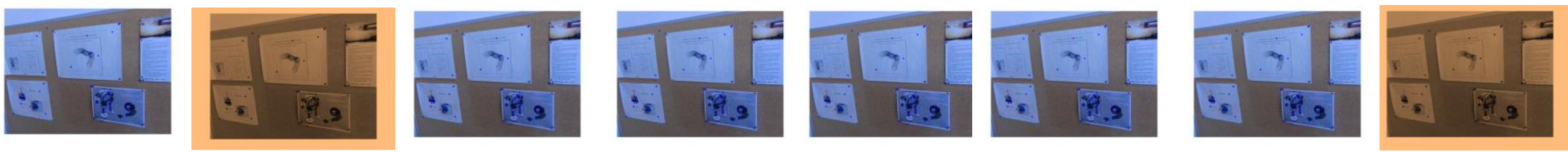
Javier Civera
Work done jointly with Alejo Concha,
Wajahat Hussain and Luis Montano

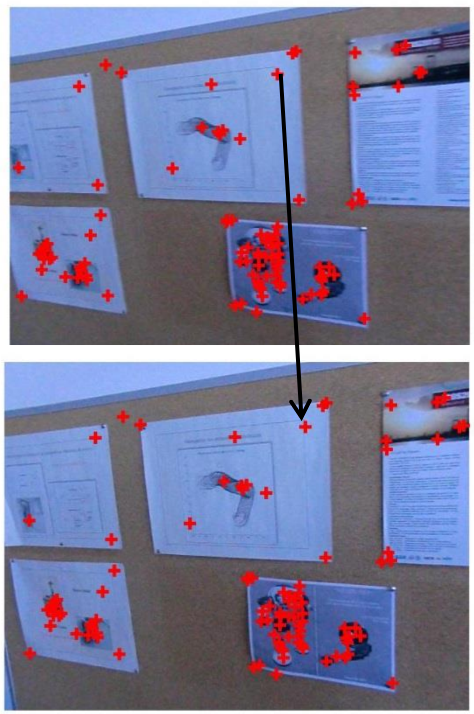
Universidad Zaragoza 1542

instituto de investigación en ingeniería de Aragón

PTAM: Parallel Tracking and Mapping

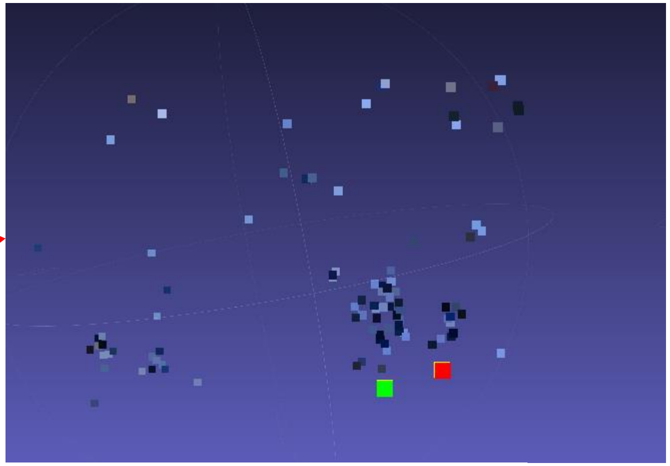
	High Texture			Low Texture		
	Accuracy	Density	Cost	Accuracy	Density	Cost
PTAM	✓	✗	✓	✓	✗	✓





Keypoints:

- Repetitive.
- Accurate.
- High color gradients.



$$\left\{ \{(R,t)_1..(R,t)_n\}, \{p_1..p_m\} \right\} = \underset{\left\{\{(R,t)\},\{p\}\right\}}{\arg\min} \sum_{i=1}^{n} \sum_{j=1}^{m} e_{ij}$$

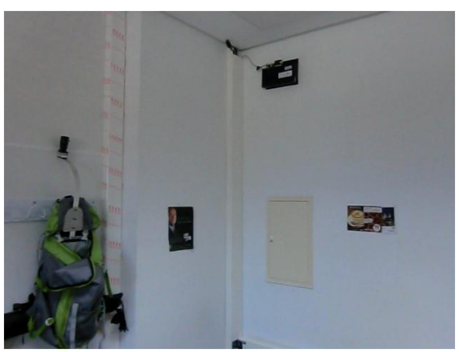
$$e_{ij} = u_j - g(p_j, R_i, t_i)$$

Klein, G., Murray, D., Parallel tracking and mapping for small ar workspaces. In Proc. of ISMAR 2007, pp. 225–234, 2007

PMVS: Patch-based Multi-View Stereo

	High Texture			Low Texture		
	Accuracy	Density	Cost	Accuracy	Density	Cost
PMVS	✓	✓	✗	✓	✗	✗






PMVS: Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(8):13621376, 2010.

DTAM: Dense Tracking and Mapping

	High Texture			Low Texture		
	Accuracy	Density	Cost	Accuracy	Density	Cost
DTAM	✓	✓	✓	✗	✓	✓

$$E_{\xi} = \int_{\Omega} \left\{ g(\mathbf{u})\|\nabla\xi(\mathbf{u})\|_{\epsilon} + \lambda \mathbf{C}\left(\mathbf{u},\xi(\mathbf{u})\right)\right\} d\mathbf{u}$$

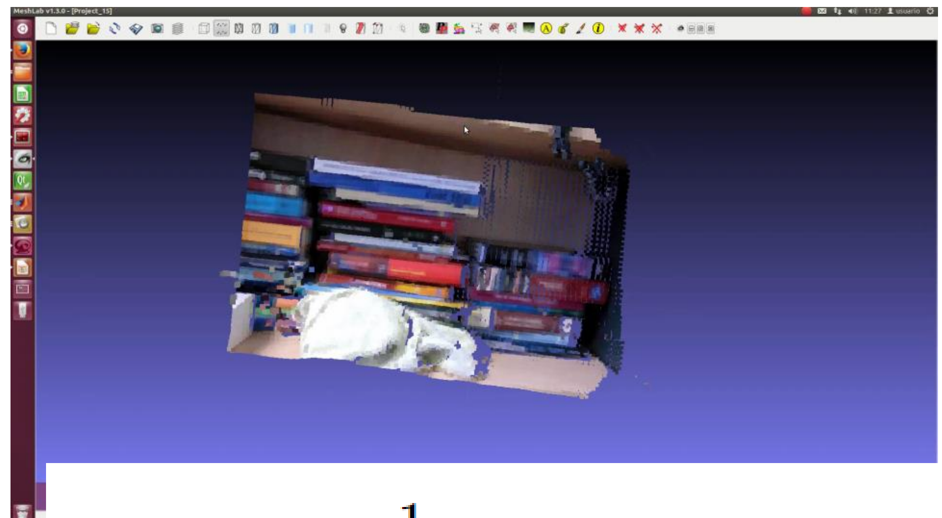
$$\mathbf{C}(\mathbf{u},\mathbf{e}) = \frac{1}{|\mathbf{I}_s|}\sum_{\mathbf{m}\in\mathbf{I}_s}\|\rho(\mathbf{I}_m,\mathbf{u},\mathbf{e})\|_1$$

$$\rho(\mathbf{I}_m,\mathbf{u},\mathbf{e}) = \mathbf{I}_r(\mathbf{u}) - \mathbf{I}_m(\mathbf{T}_{mr}(\mathbf{e},\mathbf{u}))$$

DTAM: Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In Computer Vision (ICCV), 2011 IEEE International Conference on, pages 23202327. IEEE, 2011.

DTAM: Dense Tracking and Mapping

	High Texture			Low Texture		
	Accuracy	Density	Cost	Accuracy	Density	Cost
DTAM	✓	✓	✓	✗	✓	✓



$$E_{\boldsymbol{\xi}} = \int_{\Omega} \left\{ g(\mathbf{u}) \|\nabla \boldsymbol{\xi}(\mathbf{u})\|_{\epsilon} + \lambda \mathbf{C}\left(\mathbf{u}, \boldsymbol{\xi}(\mathbf{u})\right) \right\} \mathrm{d}\mathbf{u}$$

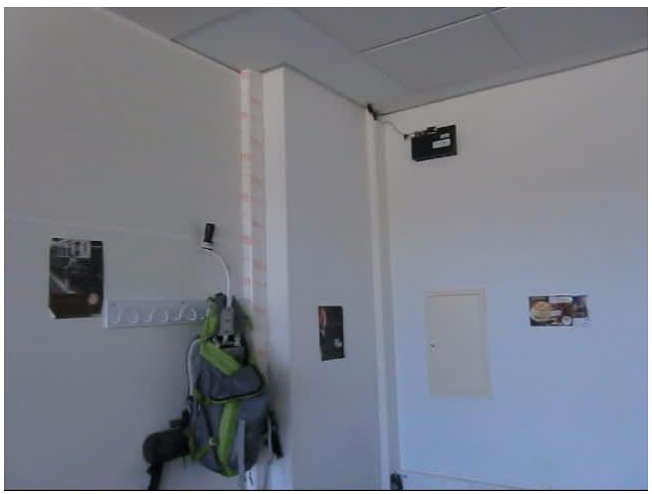
$$\mathbf{C}(\mathbf{u}, \mathbf{e}) = \frac{1}{|\mathbf{I}_{\mathrm{s}}|} \sum_{\mathbf{m} \in \mathbf{I}_{\mathrm{s}}} \|\rho(\mathbf{I}_{\mathrm{m}}, \mathbf{u}, \mathbf{e})\|_1$$

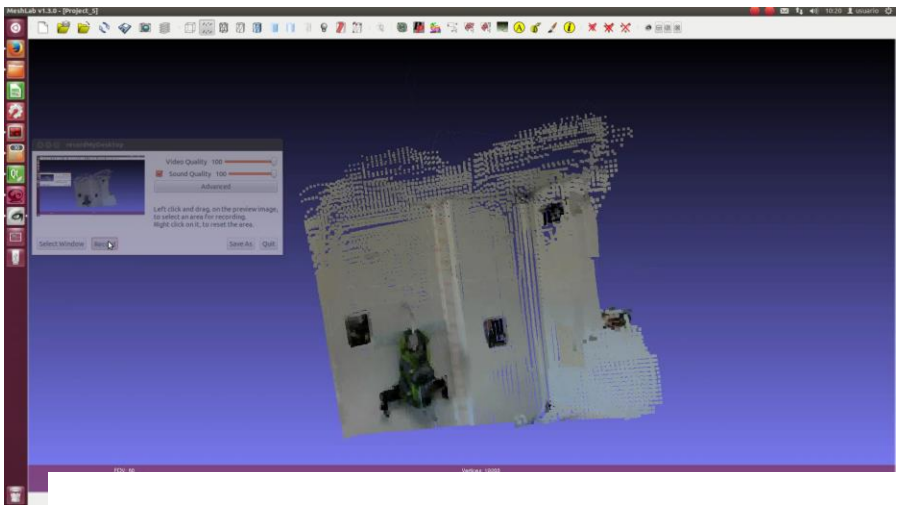
$$\rho(\mathbf{I}_{\mathrm{m}}, \mathbf{u}, \mathbf{e}) = \mathbf{I}_{\mathrm{r}}(\mathbf{u}) - \mathbf{I}_{\mathrm{m}}(\mathbf{T}_{\mathrm{mr}}(\mathbf{e}, \mathbf{u}))$$

DTAM: Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In Computer Vision (ICCV), 2011 IEEE International Conference on, pages 23202327. IEEE, 2011.

DTAM: Dense Tracking and Mapping

	High Texture			Low Texture		
	Accuracy	Density	Cost	Accuracy	Density	Cost
DTAM	✓	✓	✓	✗	✓	✓





$$E_{\boldsymbol{\xi}} = \int_{\Omega} \Big\{ g(\mathbf{u}) \|\nabla \boldsymbol{\xi}(\mathbf{u})\|_{\epsilon} + \lambda \mathbf{C}\left(\mathbf{u}, \boldsymbol{\xi}(\mathbf{u})\right) \Big\} d\mathbf{u}$$

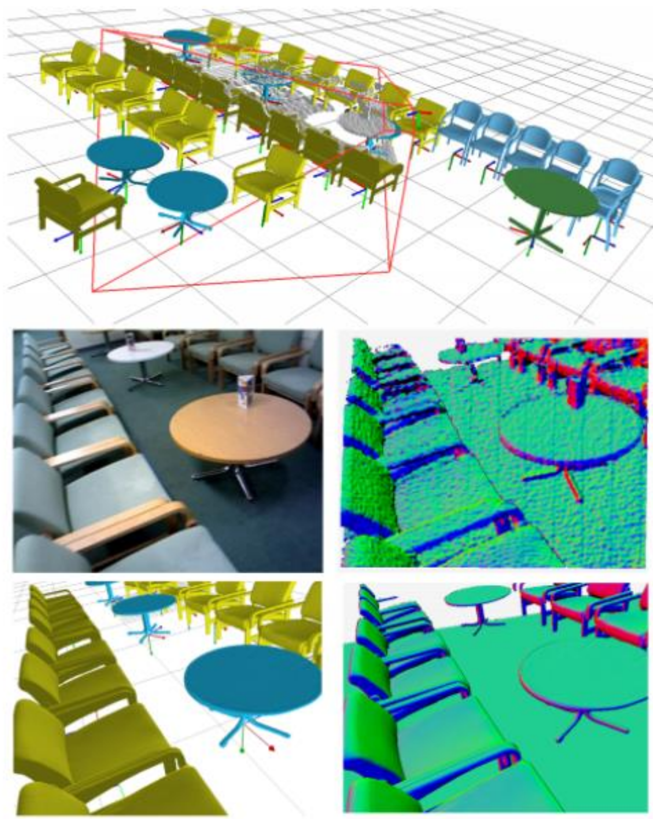
$$\mathbf{C}(\mathbf{u}, \mathbf{e}) = \frac{1}{|\mathbf{I}_{\mathrm{s}}|} \sum_{\mathbf{m} \in \mathbf{I}_{\mathrm{s}}} \|\rho(\mathbf{I}_{\mathrm{m}}, \mathbf{u}, \mathbf{e})\|_1$$

$$\rho(\mathbf{I}_{\mathrm{m}}, \mathbf{u}, \mathbf{e}) = \mathbf{I}_{\mathrm{r}}(\mathbf{u}) - \mathbf{I}_{\mathrm{m}}(\mathbf{T}_{\mathrm{mr}}(\mathbf{e}, \mathbf{u}))$$

DTAM: Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In Computer Vision (ICCV), 2011 IEEE International Conference on, pages 23202327. IEEE, 2011.

SLAM++: SLAM at the level of objects

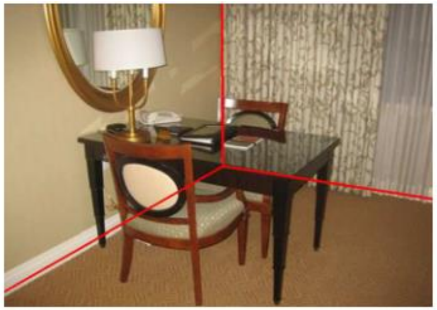
	High Texture			Low Texture		
	Accuracy	Density	Cost	Accuracy	Density	Cost
SLAM ++	✓	✗	✓	✓	✗	✓



SLAM++ : Renato F Salas-Moreno, Richard A Newcombe, Hauke Strasdat, Paul HJ Kelly, and Andrew J Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In CVPR, 2013.

Layout understanding

	High Texture			Low Texture		
	Accuracy	Density	Cost	Accuracy	Density	Cost
Layout	✗	✓	✓	✗	✓	✓



	High Texture			Low Texture		
	Accuracy	Density	Cost	Accuracy	Density	Cost
PTAM	✓	✗	✓	✓	✗	✓
PMVS	✓	✓	✗	✓	✗	✗
DTAM	✓	✓	✓	✗	✓	✓
SLAM++	✓	✗	✓	✓	✗	✓
Layout	✗	✓	✓	✗	✓	✓
Layout + DTAM	✓	✓	✓	✗	✓	✓
Superpixels	✗	✓	✗	✓	✓	✓
Superpixels + DTAM	✓	✓	✓	✓	✓	✓

Varsha Hedau, Derek Hoiem, and David Forsyth. Recovering the spatial layout of cluttered rooms. ICCV 2009

Superpixels

	High Texture			Low Texture		
	Accuracy	Density	Cost	Accuracy	Density	Cost
PTAM	✓	✗	✓	✓	✗	✓
PMVS	✓	✓	✗	✓	✗	✗
DTAM	✓	✓	✓	✗	✓	✓
SLAM++	✓	✗	✓	✓	✗	✓
Layout	✗	✓	✓	✗	✓	✓
Layout + DTAM	✓	✓	✓	✗	✓	✓
Superpixels	✗	✓	✗	✓	✓	✓
Superpixels + DTAM	✓	✓	✓	✓	✓	✓

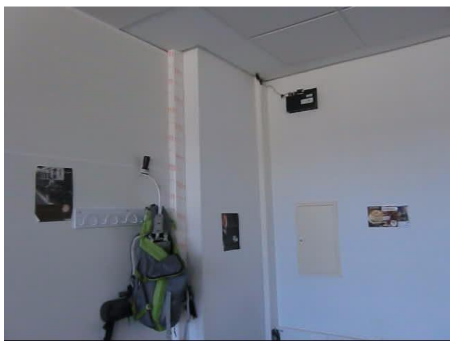
* Assuming planarity

** Assuming GPU

Pedro F Felzenszwalb and Daniel P Huttenlocher. Ecient graph-based image segmentation. International Journal of Computer Vision, 59(2):167181, 2004.

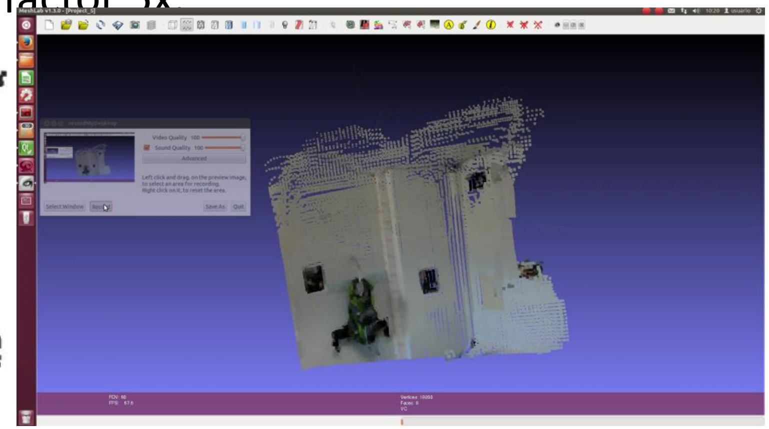
Superpixels/Layout in low-textured areas

- State-of-the-art methods are limited in large untextured areas.
- We propose to use mid-level features (superpixels) and high-level features (layout) to overcome such limitations.
- The only assumption is a priori over some areas to be planar.
- Our results show that the median error is reduced in a factor 5x.

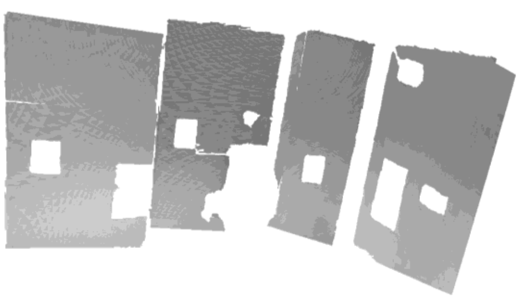


Video (input)

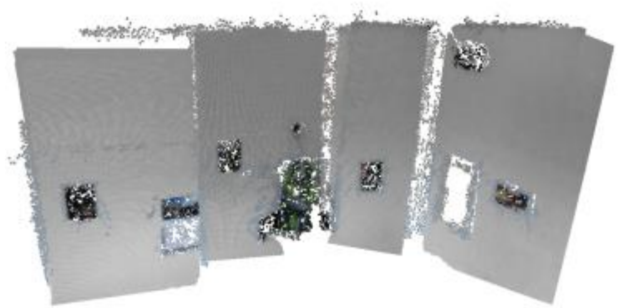
PMVS



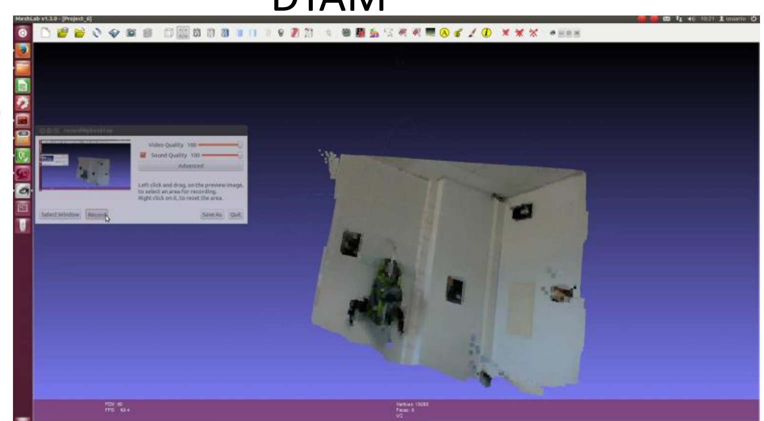
DTAM



Superpixels



PMVS + Superpixels

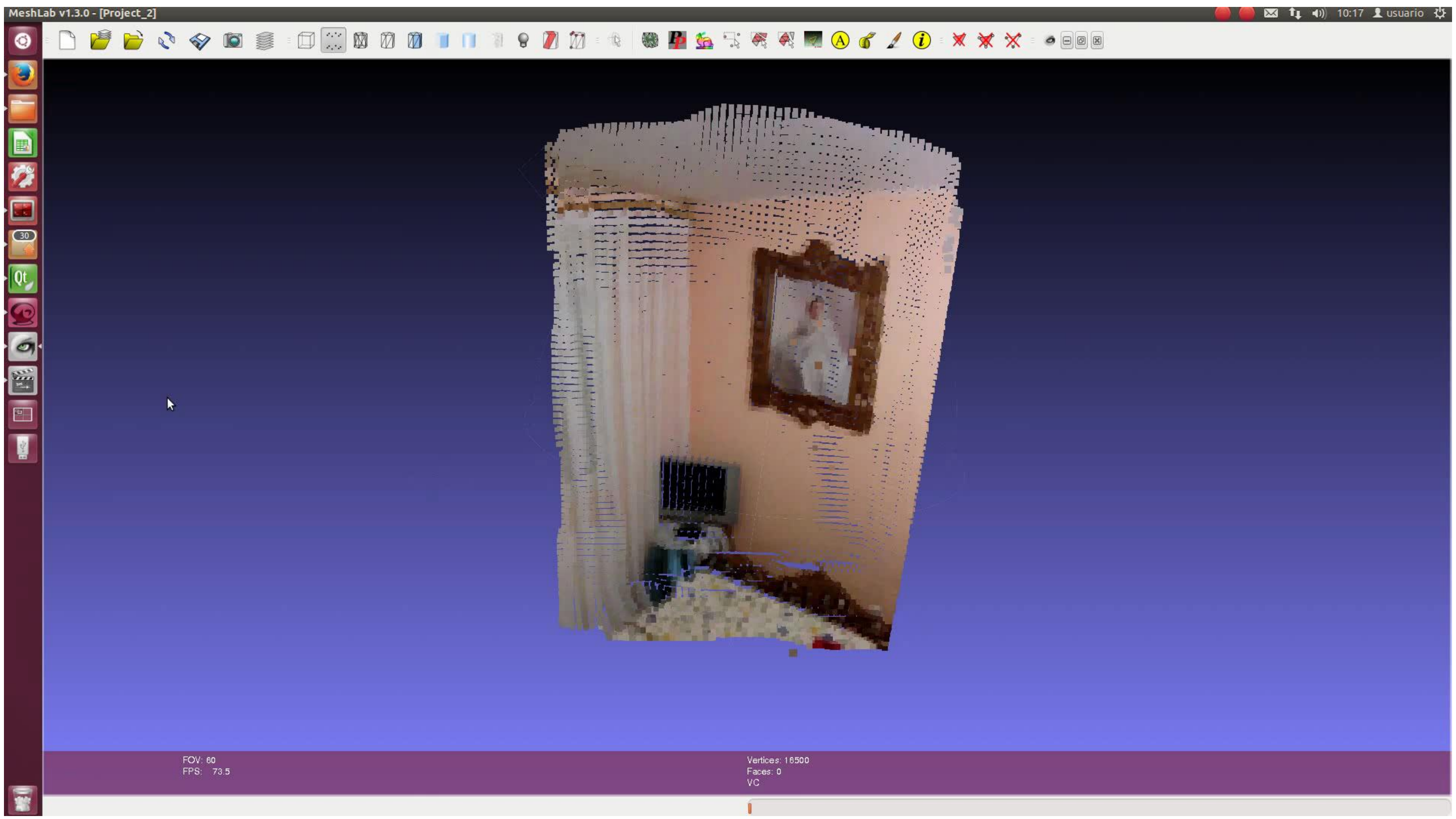


DTAM + Superpixels

Superpixels in low-textured areas

DTAM SOLUTION

	High Texture			Low Texture		
	Accuracy	Density	Cost	Accuracy	Density	Cost
DTAM	✓	✓	✓	✓	✗	✓

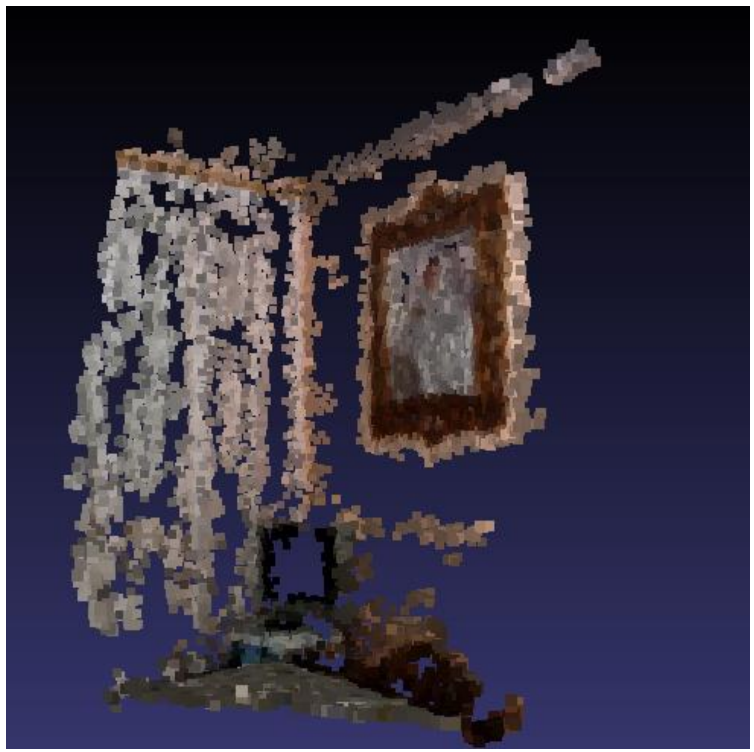


DTAM: Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In Computer Vision (ICCV), 2011 IEEE International Conference on, pages 23202327. IEEE, 2011.

Superpixels in low-textured areas

PMVS SOLUTION

	High Texture			Low Texture		
	Accuracy	Density	Cost	Accuracy	Density	Cost
PMVS	✓	✓	✗	✓	✗	✗

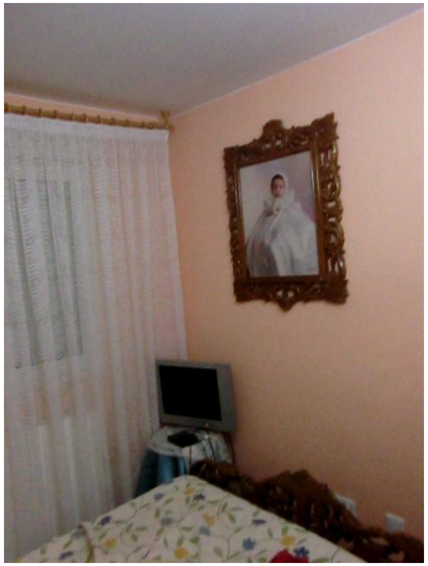


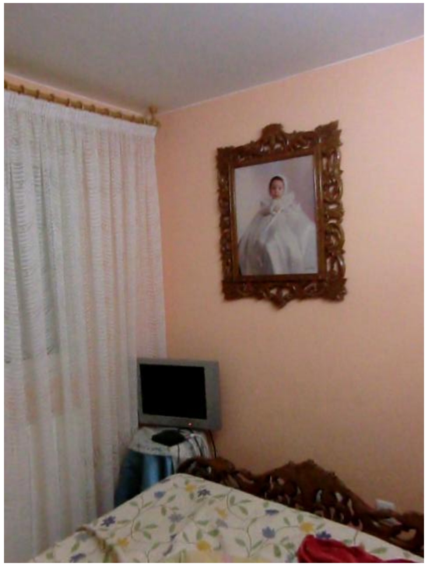
PMVS: Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(8):13621376, 2010.

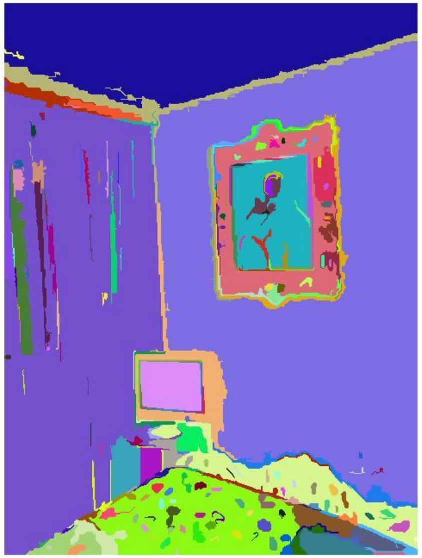
Superpixels in low-textured areas

SUPERPIXELS SOLUTION

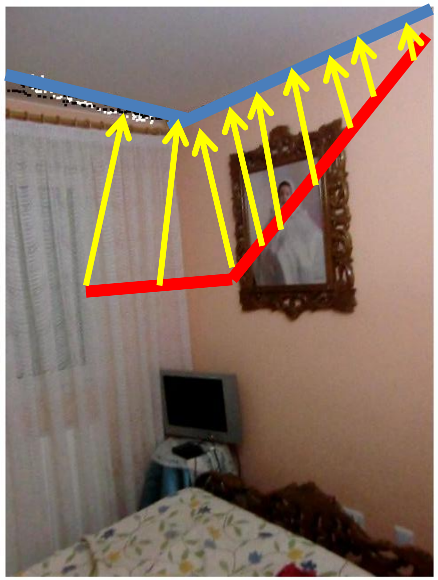
	High Texture			Low Texture		
	Accuracy	Density	Cost	Accuracy	Density	Cost
Superpixels	✗	✓	✗	✓	✓	✓







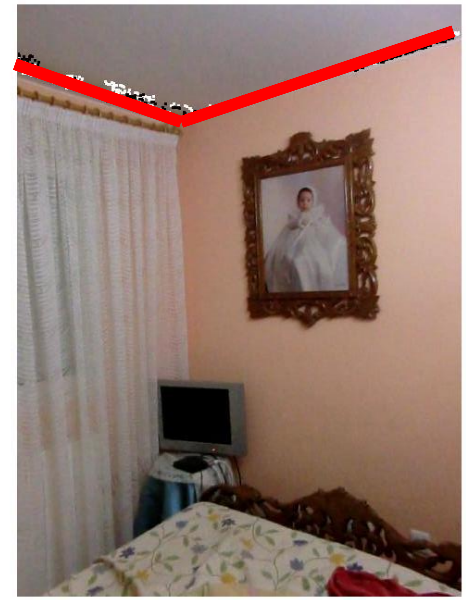
Superpixels in low-textured areas



SUPERPIXELS SOLUTION

Homography:

$\longleftarrow h(n, d) \longrightarrow$

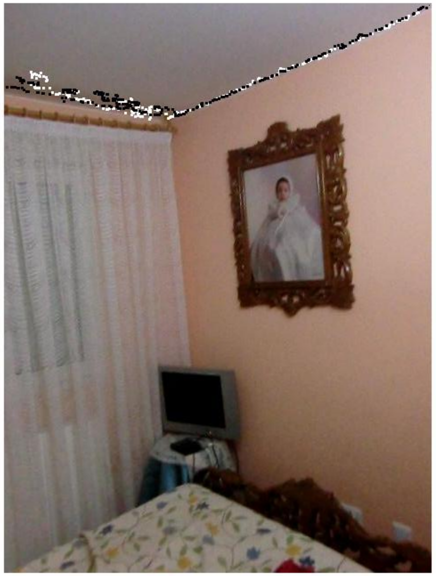


$$\epsilon_{s^h_{k,c}} = ||\mathbf{u}^l_{s^h_{k,c}} - \mathbf{h}\left(\mathbf{u}^j_{s^h_{k,c}}, \mathbf{s}_k, \mathbf{c}_j, \mathbf{c}_l\right)||$$

Estimation of the parameters of the homography

- Montecarlo approach to initialize using a robust cost function.
- Contours overlapping.
- Levenberg-Marquardt to optimize using a robust cost function

Superpixels in low-textured areas



SUPERPIXELS SOLUTION

Homography:

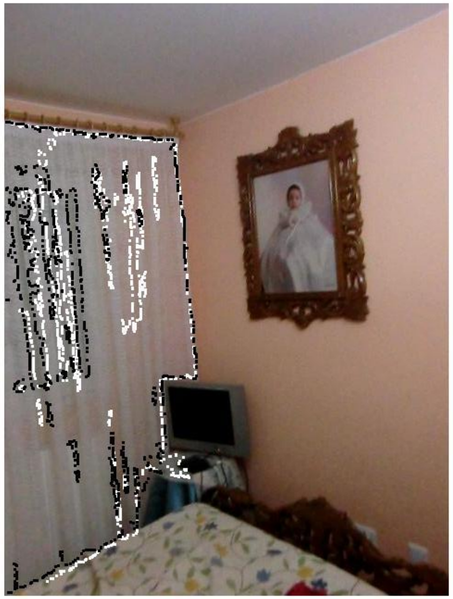
$$\leftarrow h(n, d) \rightarrow$$

$$\epsilon_{s^h_{k,c}} = ||\mathbf{u}^l_{s^h_{k,c}} - \mathbf{h}\left(\mathbf{u}^j_{s^h_{k,c}}, \mathbf{s}_k, \mathbf{c}_j, \mathbf{c}_l\right)||$$

Estimation of the parameters of the homography

- Montecarlo approach to initialize using a robust cost function.
- Contours overlapping.
- Levenberg-Marquardt to optimize using a robust cost function

Superpixels in low-textured areas



SUPERPIXELS SOLUTION

Homography:

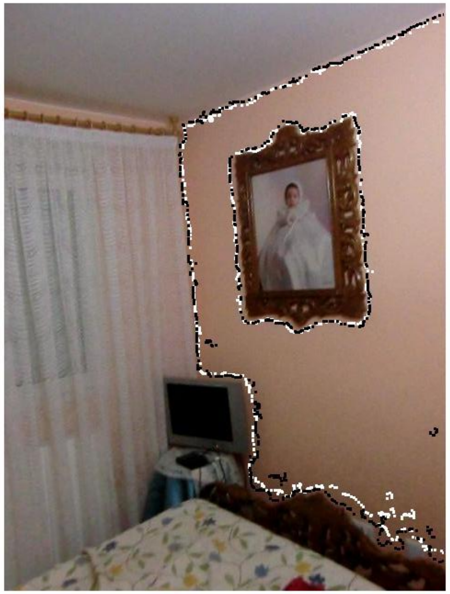
$$\leftarrow h(n,d) \rightarrow$$

$$\epsilon_{s^h_{k,c}} = ||\mathbf{u}^l_{s^h_{k,c}} - \mathbf{h}\left(\mathbf{u}^j_{s^h_{k,c}}, \mathbf{s}_k, \mathbf{c}_j, \mathbf{c}_l\right)||$$

Estimation of the parameters of the homography

- Montecarlo approach to initialize using a robust cost function.
- Contours overlapping.
- Levenberg-Marquardt to optimize using a robust cost function

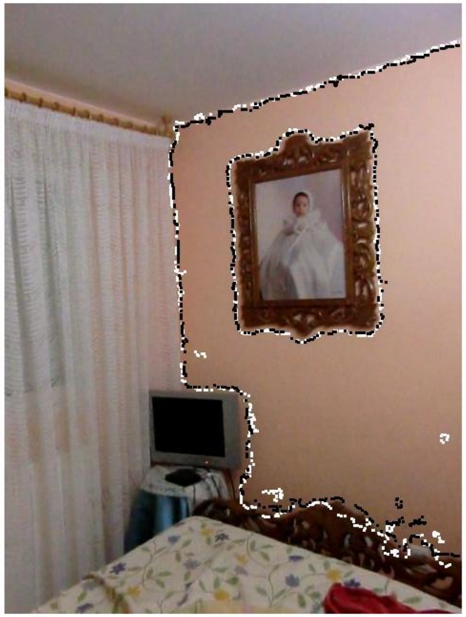
Superpixels in low-textured areas



SUPERPIXELS SOLUTION

Homography:

$\longleftarrow \; h(n,d) \; \longrightarrow$



$$\epsilon_{s^h_{k,c}} = \left\| \mathbf{u}^l_{s^h_{k,c}} - \mathbf{h}\left(\mathbf{u}^j_{s^h_{k,c}}, \mathbf{s}_k, \mathbf{c}_j, \mathbf{c}_l\right) \right\|$$

Estimation of the parameters of the homography

- Montecarlo approach to initialize using a robust cost function.
- Contours overlapping.
- Levenberg-Marquardt to optimize using a robust cost function

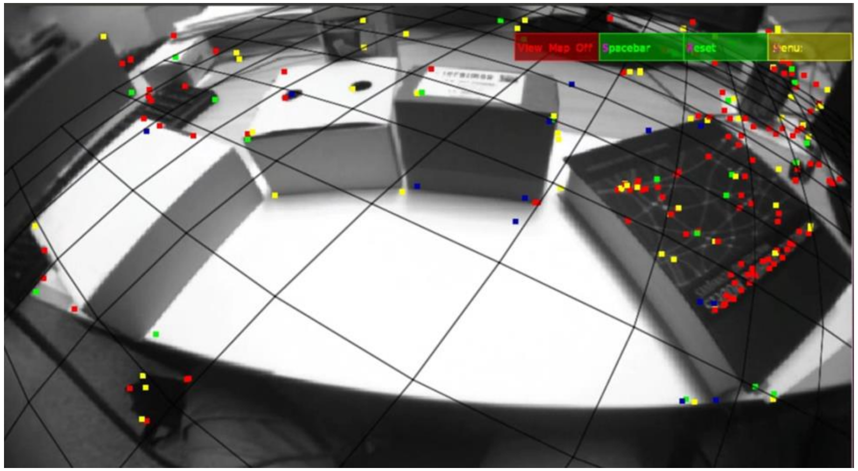
Superpixels in low-textured areas

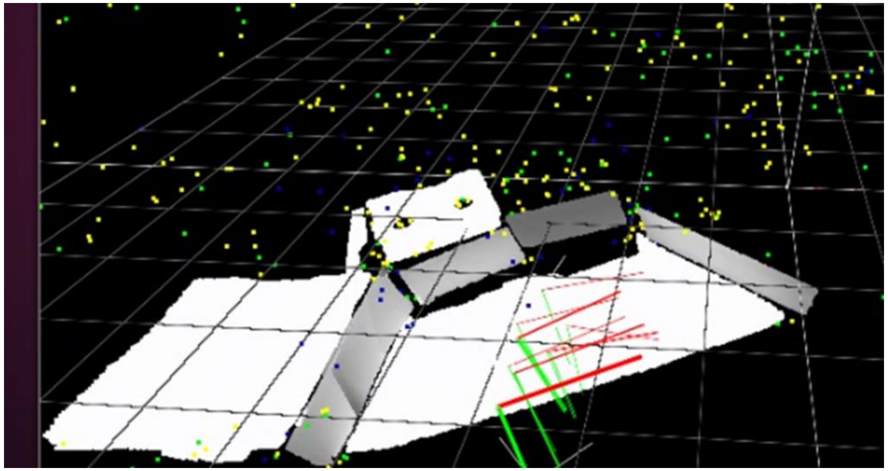
SUPERPIXELS SOLUTION

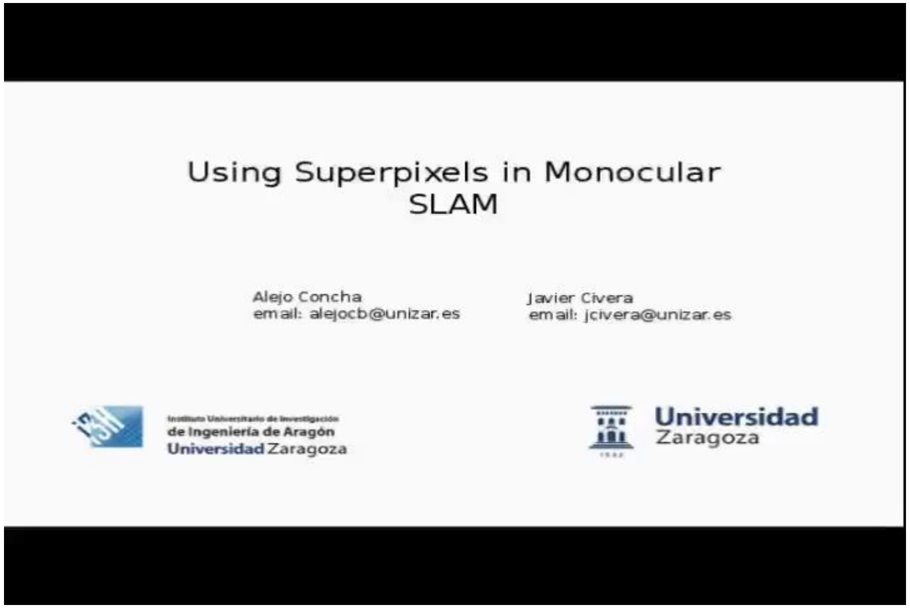
	High Texture			Low Texture		
	Accuracy	Density	Cost	Accuracy	Density	Cost
Superpixels	✗	✓	✗	✓	✓	✓

Alejo Concha and Javier Civera. Using Superpixels in Monocular SLAM. ICRA 2014

Using Superpixels in Monocular SLAM







Alejo Concha and Javier Civera. Using Superpixels in Monocular SLAM. ICRA 2014

DTAM + Superpixels/Layout

SOLUTION USING SUPERPIXELS AND DTAM

	High Texture			Low Texture		
	Accuracy	Density	Cost	Accuracy	Density	Cost
DTAM	✓	✓	✓	✓	✓	✓

$$E_\rho = \int (\lambda_1 \mathbf{C}(\mathbf{u}, \rho(\mathbf{u})) + \mathbf{G}(\mathbf{u}, \rho(\mathbf{u})) + \\ + \frac{\lambda_2}{2} \mathbf{M}(\mathbf{u}, \rho(\mathbf{u}), \rho_{\mathbf{p}}(\mathbf{u})) \partial \mathbf{u}$$

$$\mathbf{C}(\mathbf{u}, \rho(\mathbf{u})) = \frac{1}{|I_s|} \sum_{j=1, j\neq r}^{m} ||\epsilon(\mathbf{I}_j, \mathbf{I}_r, \mathbf{u}, \rho)||_1$$

$$\epsilon(\mathbf{I}_j, \mathbf{I}_r, \mathbf{u}, \rho) = \mathbf{I}_r(\mathbf{u}) - \mathbf{I}_j(\mathbf{T}_{rj}(\mathbf{u}, \rho))$$

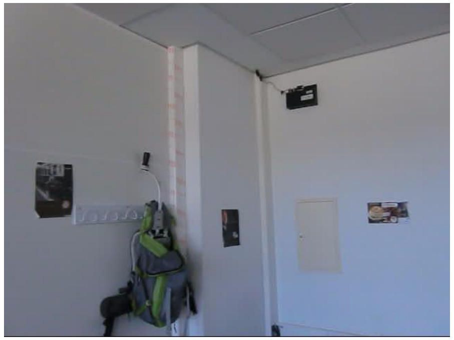
$$\mathbf{G}(\mathbf{u}^{\mathbf{r}}, \rho(\mathbf{u})) = \mathbf{g}(\mathbf{u}^{\mathbf{r}}) ||\nabla \rho(\mathbf{u})||_\epsilon$$

$$\mathbf{M}(\mathbf{u}, \rho(\mathbf{u}), \rho_{\mathbf{p}}(\mathbf{u})) = ||\rho(\mathbf{u}) - \rho_{\mathbf{p}}(\mathbf{u})||_{\mathbf{2}}^{\mathbf{2}}$$

Depth prior comes from multiview superpixels or scene layout understanding

Alejo Concha, Wajahat Hussain, Luis Montano and Javier Civera. Manhattan and Piece-Wise Planar Regularization for Dense Mapping with a Monocular Camera. RSS 2014

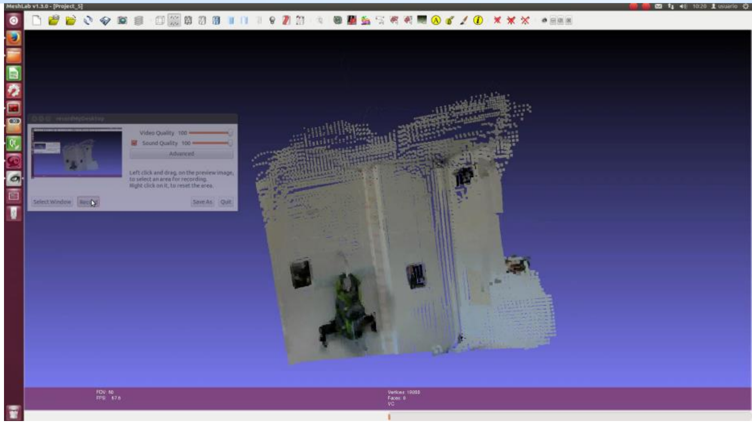
DTAM + Superpixels



Video (input)

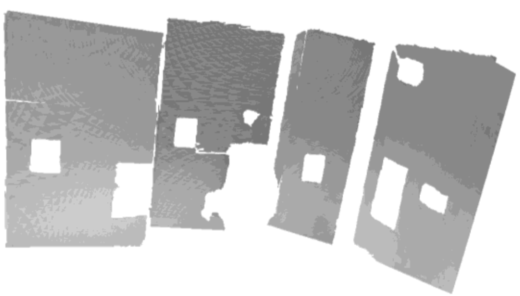
PMVS

Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(8):13621376, 2010.



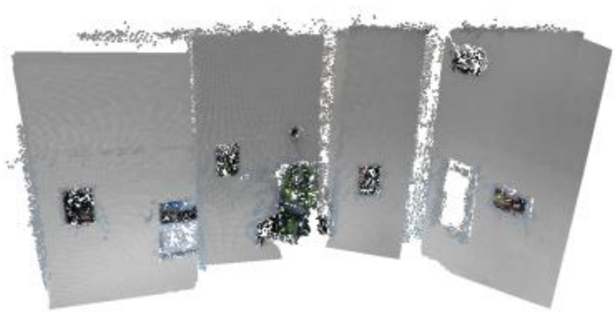
DTAM

Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In Computer Vision (ICCV), 2011 IEEE International Conference on, pages 23202327. IEEE, 2011.

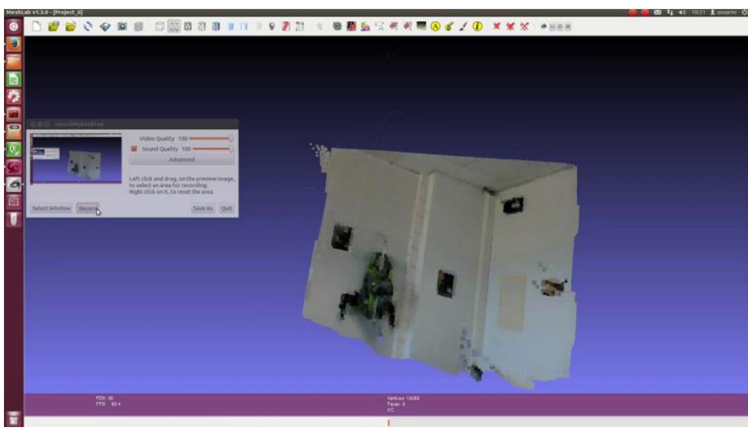


Superpixels

Alejo Concha and Javier Civera. Using Superpixels in Monocular SLAM. ICRA 2014



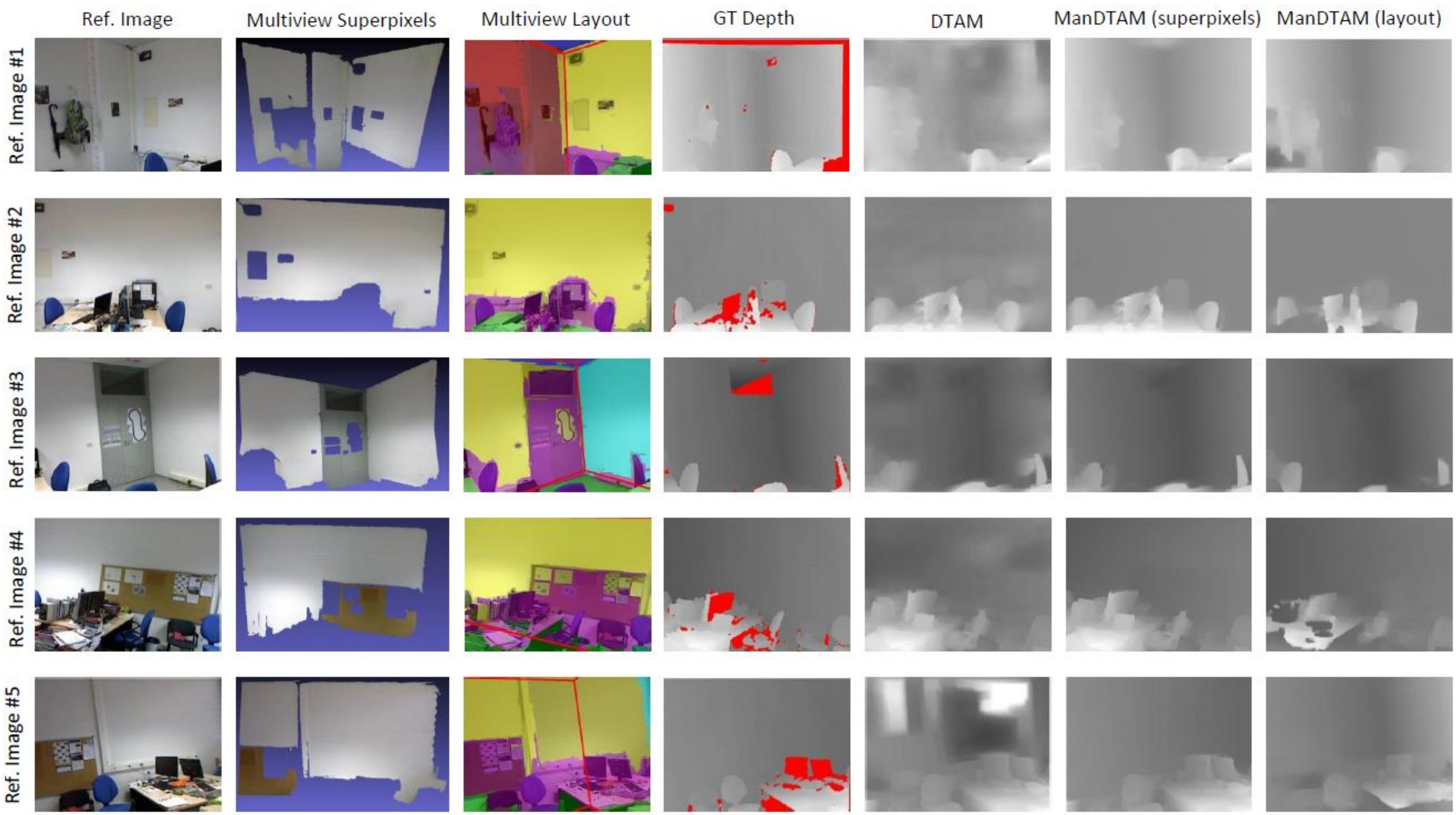
PMVS + Superpixels



DTAM + Superpixels

Alejo Concha, Wajahat Hussain, Luis Montano and Javier Civera. Manhattan and Piece-Wise Planar Regularization for Dense Mapping with a Monocular Camera. RSS 2014

DTAM + Layout



Alejo Concha, Wajahat Hussain, Luis Montano and Javier Civera. Manhattan and Piece-Wise Planar Regularization for Dense Mapping with a Monocular Camera. RSS 2014

Conclusions

- Low-level features (salient points and lines) are unable to reconstruct large and and textureless areas.
- Mid-level features (superpixels) and high-level understanding (layout) allow to model such areas; but might be less accurate than low-level features in textured ones.
- We fuse standard low-level point features with mid-level ones and scene understanding improve the accuracy of dense 3D maps from RGB cameras (in our experiments, the median error is reduced 5x).
- We are the first in using such features in dense RGB mapping; and we believe it is a promising line of research.

Robust Loop Closing Over Time

Yasir Latif

joint work with

Cesar Cadena and Jose Neira

Universidad Zaragoza

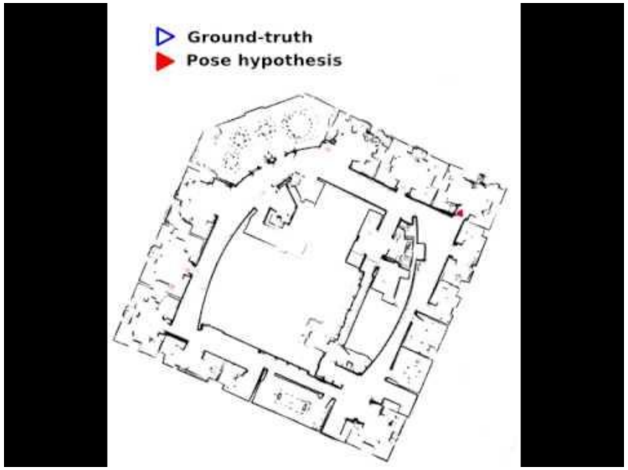
1542

Place Recognition / Detecting Loop Closures

Feature space

Sensor space

- Laser [FLIRT Points] [Correlative Scan Matching]
- Vision [DBoW] [FAP-MAP]



Universidad Zaragoza

Perceptual Aliasing

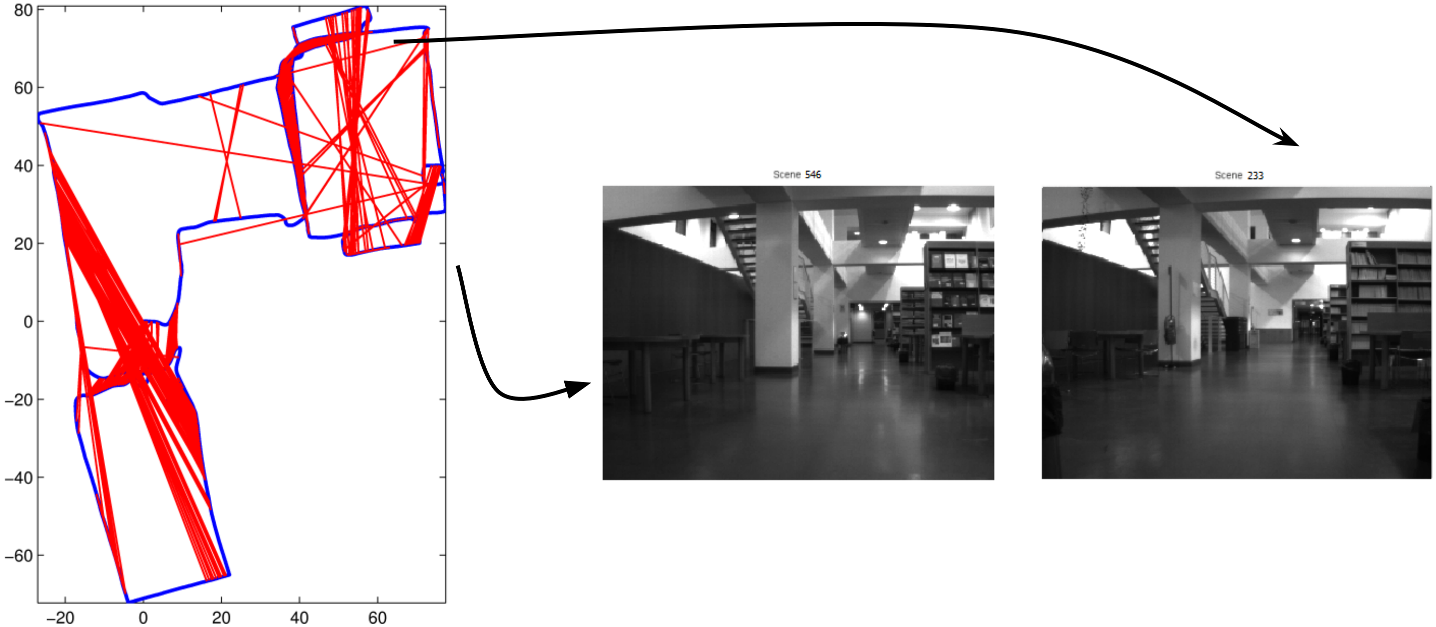
In sensor space, different places might look the same

Scene 546

Scene 233

Universidad Zaragoza 1542

Perceptual Aliasing

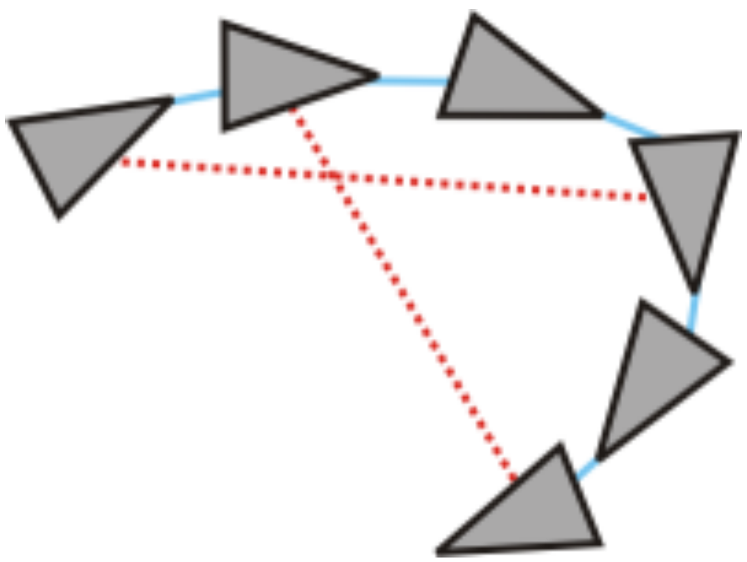


Universidad Zaragoza
1542

Graph SLAM

Least Squares formulation of SLAM

input	odometry and loop closures
output	best guess of robot locations
outliers	a disaster!



Robust Loop Closing

- Place recognition algorithms are not perfect
  - perceptual aliasing leads to false positives
  - map corruption rather than improvement
- Robust loop closing
  - Utilize global information
  - Ideally, never make a mistake
  - Recover from mistakes
- Main Idea
  - Odometry is reliable
  - Loop closures should agree with odometry and among themselves (consistency)
  - Consistency determined via chi-square tests

Universidad Zaragoza
1542

χ^2-test (Chi-squared test)

$$D_l^2(\mathbf{x}) = \boxed{r_{ij}(\mathbf{x})^T \Omega_{ij} r_{ij}(\mathbf{x})} < \chi^2_{\alpha,d_l}, \;\; (i,j) \in R_i$$

Parameters

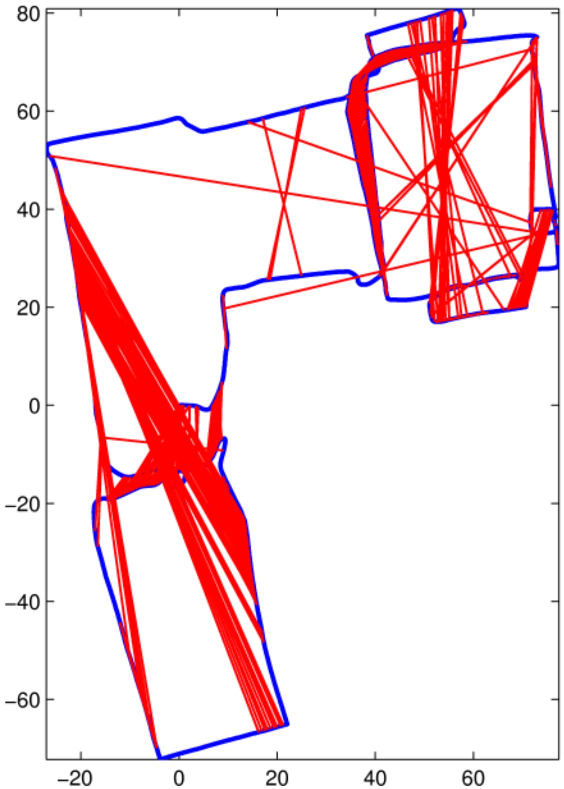
α: confidence level (usually 95%)

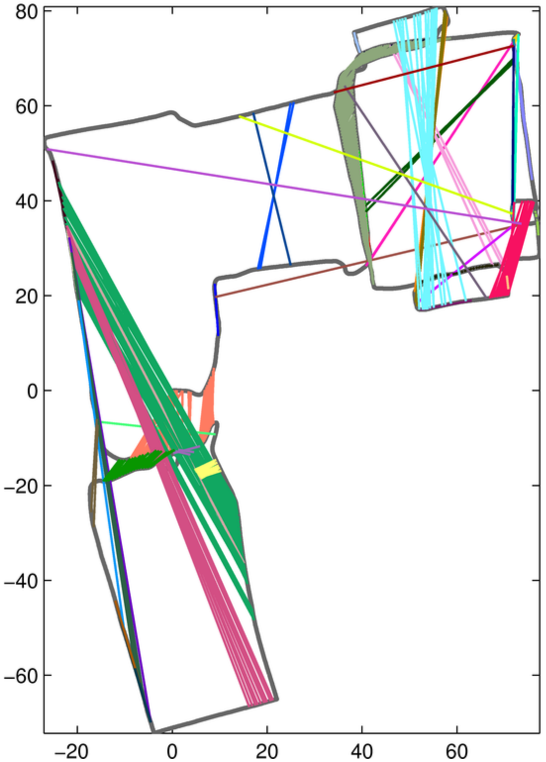
d: degrees of freedom

Simply put: with a confident α, we can say that this residual comes from a distribution with the given covariance.

Realizing Reversing Recovering [RRR]

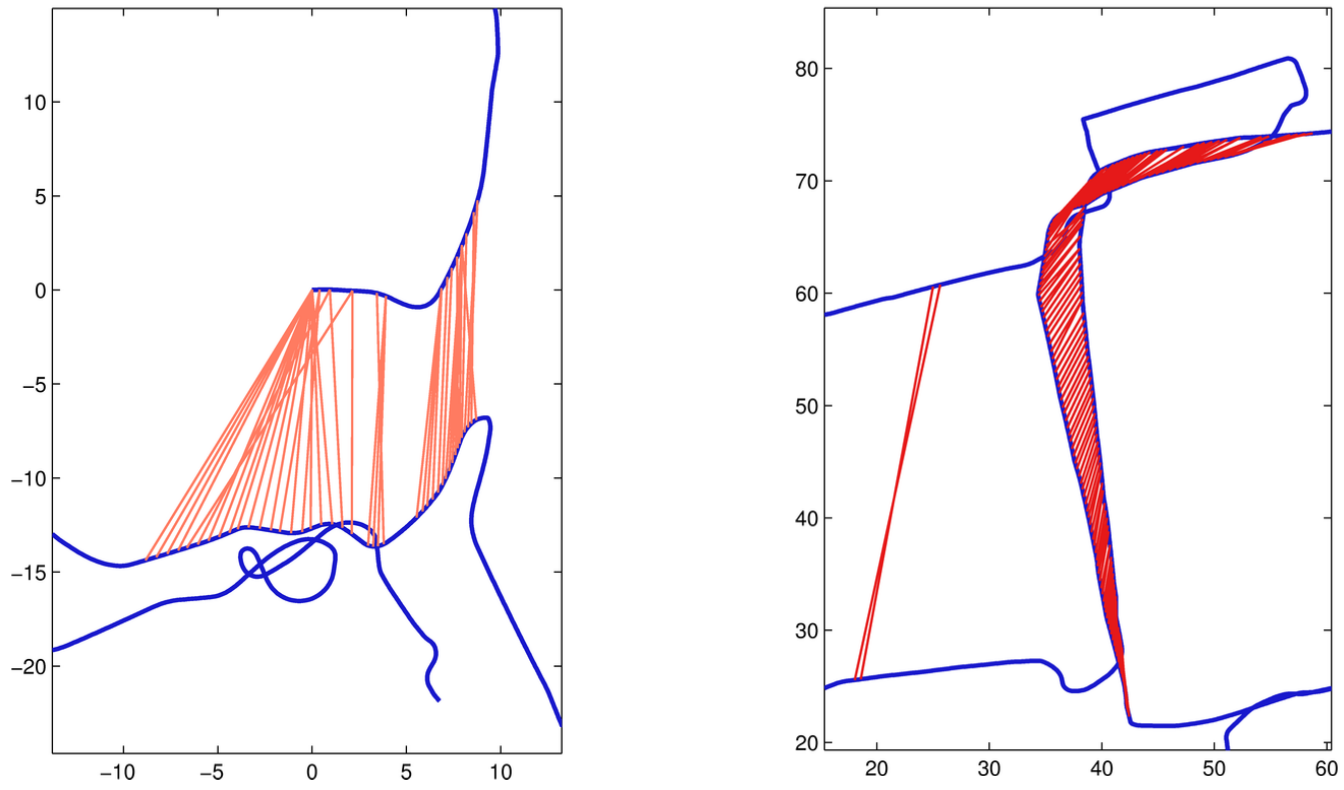
Reason on clusters





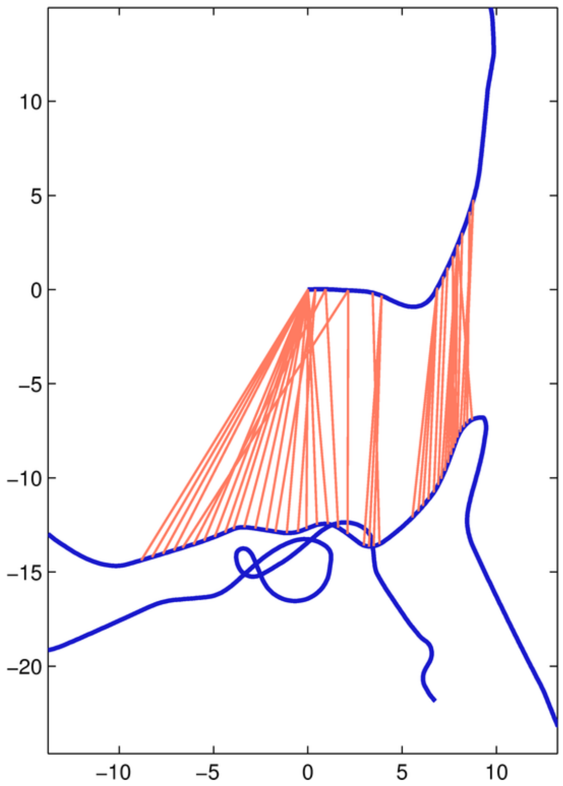
Universidad Zaragoza
1542

Intra- and Inter-cluster consistency



Universidad Zaragoza 1542

Intracluster Consistency

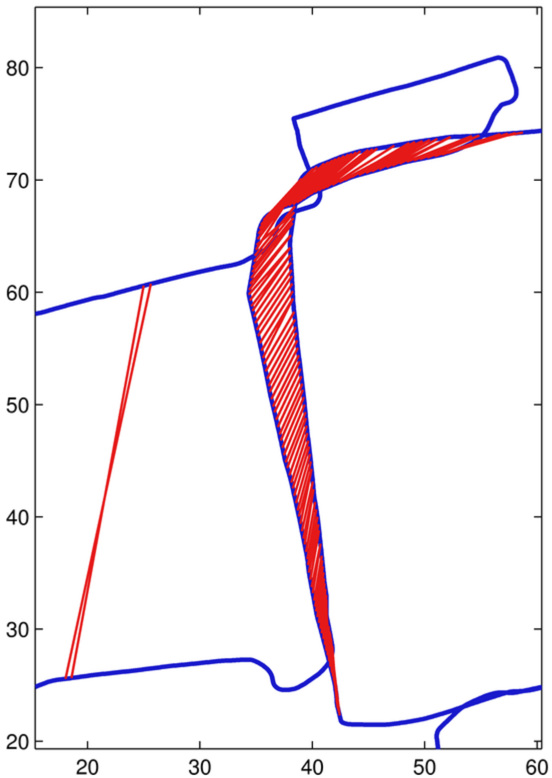


Algorithm 1 Intra_Cluster_Consistency

Input: $poses$, $slinks$, $cluster$ of $rlinks$
Output: $cluster$
add $poses, slinks$ to PoseGraph
PoseGraphIC $\leftarrow$ PoseGraph
add $cluster$ to PoseGraphIC
optimize PoseGraphIC
if $D_G^2 < \chi^2_{\alpha,d_G}$ **then**
    for each $rlink_l \in cluster$ **do**
        if $D_l^2 < \chi^2_{\alpha,d_l}$ **then**
            Accept $rlink_l$
        else
            Reject $rlink_l$
        end if
    end for
else
    Reject $cluster$
end if

Universidad Zaragoza
1542

Inter-cluster consistency



Algorithm 2 Inter_Cluster_Consistency

Input: $goodSet$, $candidateSet$, PoseGraph
Output: $goodSet$, $rejectSet$

    PoseGraphJC ← PoseGraph
    add $(goodSet, candidateSet)$ to PoseGraphJC
    $rejectSet \leftarrow \{\}$
    optimize PoseGraphJC
    if $D_C^2 < \chi^2_{\alpha,d_C} \wedge D_G^2 < \chi^2_{\alpha,d_G}$ **then**
        $goodSet \leftarrow \{goodSet, candidateSet\}$
    else
        find the $cluster_i \in candidateSet$ with largest CI
        remove $cluster_i$ from $candidateSet$
        $rejectSet \leftarrow cluster_i$
        if ¬isempty $candidateSet$ **then**
            $(goodSet, rSet) \leftarrow$
            $Inter\_Cluster\_Consistency(goodSet, candidateSet)$
            $rejectSet \leftarrow \{rejectSet, rSet\}$
        end if
    end if

Universidad Zaragoza
1542

RRR

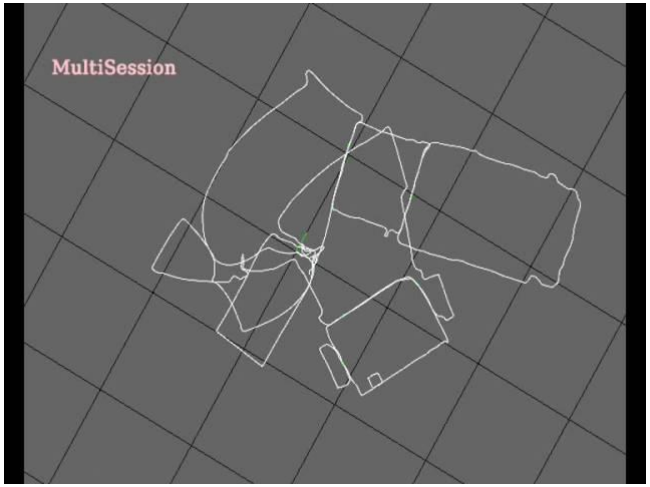
Algorithm 3 RRR

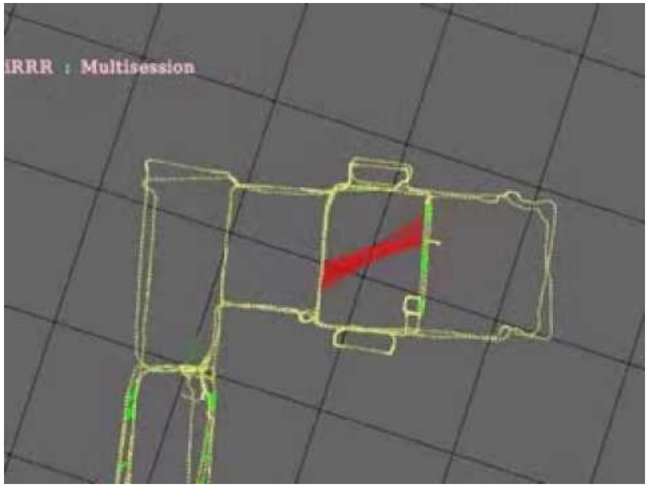
Input: $poses$, $slinks$, $\mathcal{R}$ set of clusters containing $rlinks$
Output: $goodSet$ of $rlinks$

  add $poses, slinks$ to PoseGraph
  $goodSet \leftarrow \{\}$
  $rejectSet \leftarrow \{\}$
  loop
    PoseGraphPR $\leftarrow$ PoseGraph
    $currentSet \leftarrow R \backslash \{goodSet \cup rejectSet\}$
    $candidateSet \leftarrow \{\}$
    add $currentSet$ to PoseGraphPR
    optimize PoseGraphPR
    for each $cluster_i \in currentSet$ **do**
      if $\exists D_l^2 < \chi^2_{\alpha,d_l} \mid rlink_j \in cluster_i$ **then**
        $candidateSet \leftarrow \{candidateSet, cluster_i\}$
      end if
    end for
    if isempty$(candidateSet)$ **then**
      STOP
    else
      $s = goodSet.size$
      $(goodSet, rSet) \leftarrow$
      $Inter\_Cluster\_Consistency(goodSet, candidateSet)$
      if $goodSet.size > s$ **then**
        $rejectSet \leftarrow \{\}$
      else
        $rejectSet \leftarrow \{rejectSet, rSet\}$
      end if
    end if
  end loop

Results

a correct set of loop closures leads to an acceptable solution of the pose-graph

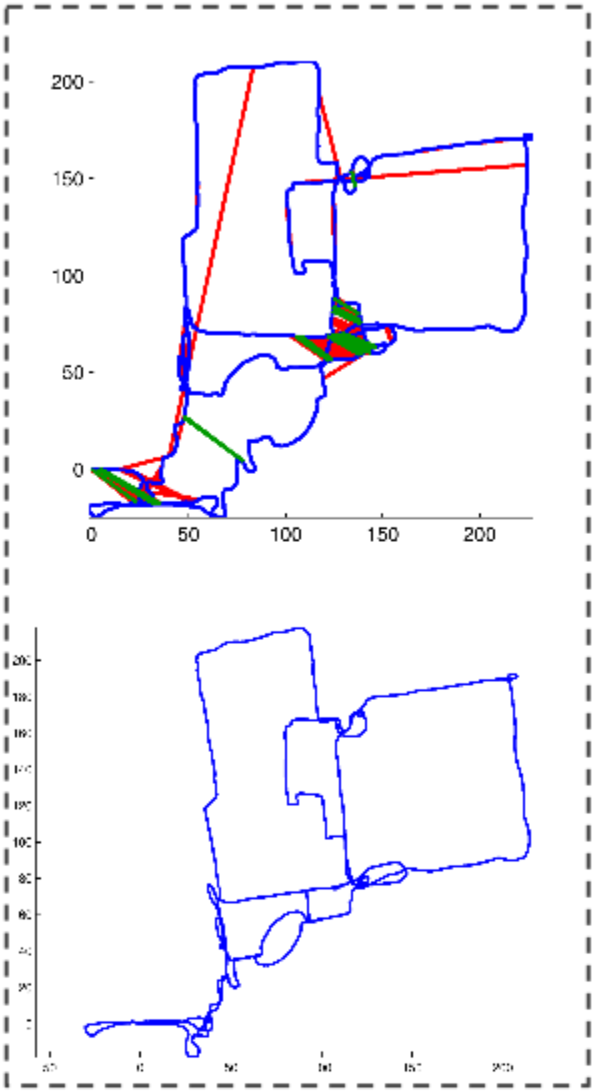




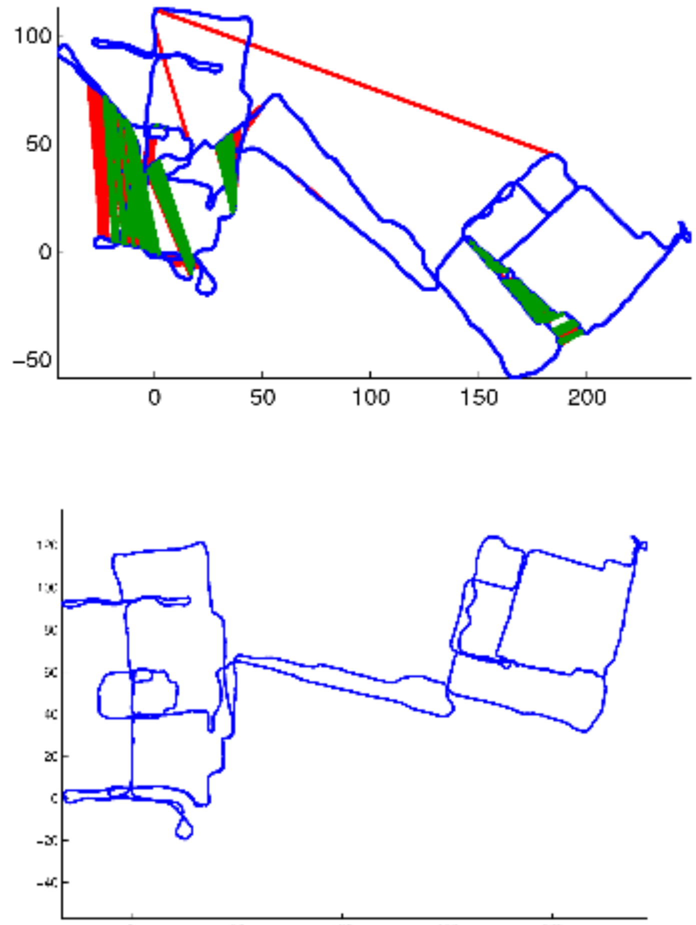
Universidad Zaragoza

More results

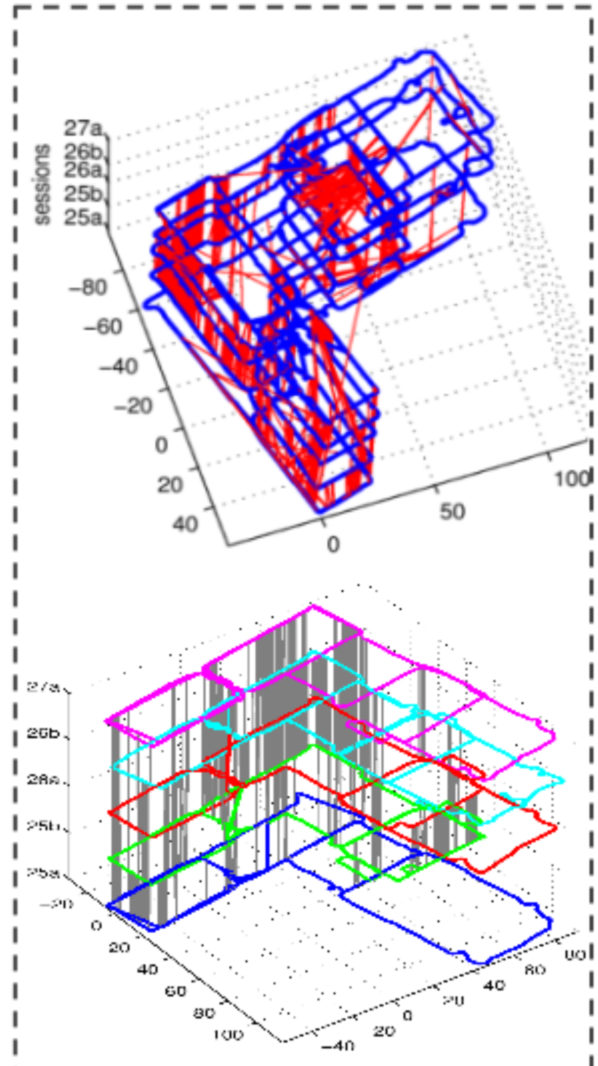
Outdoors (Bovisa Campus)



Mix outdoor-Indoor



Indoors - Multi-session



Competing approaches

Dynamic Covariance Scaling[DCS]

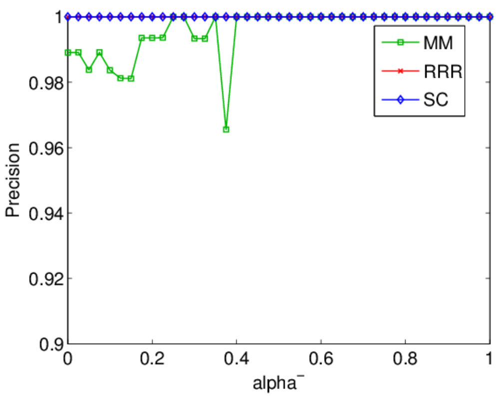
Adaptively updates the covariance matrix

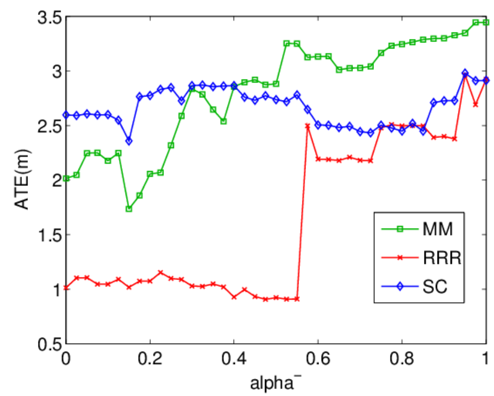
Switchable Constraints[SC]

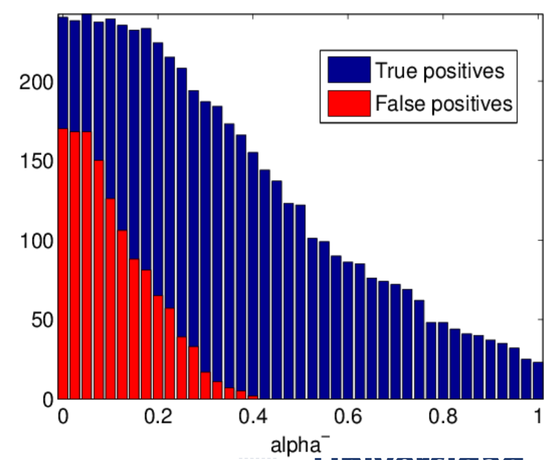
Regularization based on residual error

Max-Mixtures[MM]

Enable/Disable loop closures based on a prior error distribution







Universidad Zaragoza

References

[Flirt Points] : Tipaldi, Gian Diego, and Kai Oliver Arras. "FLIRT-interest regions for 2D range data." Robotics and Automation (ICRA), IEEE International Conference on, 2010.

[Correlative scan matching] : Olson, Edwin B. "Real-time correlative scan matching." Robotics and Automation, IEEE International Conference on, 2009.

[DBow]: Galvez-Lopez, Dorian, and Juan D. Tardos. "Bags of binary words for fast place recognition in image sequences." Robotics, IEEE Transactions on 28.5 (2012): 1188-1197.

[FAB-MAP]: Cummins, Mark, and Paul Newman. "FAB-MAP: Probabilistic localization and mapping in the space of appearance." The International Journal of Robotics Research 27.6 (2008): 647-665.

[RRR]: Yasir Latif, César Cadena, and José Neira. "Robust loop closing over time for pose graph SLAM." The International Journal of Robotics Research 32.14 (2013): 1611-1626.

[DCS]: Agarwal, Pratik, et al. "Robust map optimization using dynamic covariance scaling." Robotics and Automation (ICRA), 2013 IEEE International Conference on. IEEE, 2013.

[MM]: Olson, Edwin, and Pratik Agarwal. "Inference on networks of mixtures for robust robot mapping." The International Journal of Robotics Research 32.7 (2013): 826-840.

[SC]: Sunderhauf, N., and Peter Protzel. "Switchable constraints for robust pose graph slam." Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. IEEE, 2012.

Universidad Zaragoza
1542

Scary formulas

$$D^2(\mathbf{x}) = \boxed{\sum_{(i,j)\in S} d_{ij}(\mathbf{x})^2} + \boxed{\sum_{(i,j)\in R} d_{ij}(\mathbf{x})^2}$$

$$D_G^2(\mathbf{x}) = \boxed{\sum_{(i,j)\in R_i} r_{ij}(\mathbf{x})^T \Omega_{ij} r_{ij}(\mathbf{x})} + \boxed{\sum_{(i,j)\in S} d_{ij}(\mathbf{x})^2} < \chi^2_{\alpha,d_G}$$

$$D_l^2(\mathbf{x}) = \boxed{r_{ij}(\mathbf{x})^T \Omega_{ij} r_{ij}(\mathbf{x})} < \chi^2_{\alpha,d_l}, \ \ (i,j) \in R_i$$

$$D_C^2(\mathbf{x}) = \boxed{\sum_{c=1}^{|C|} \sum_{(i,j)\in R_c} r_{ij}(\mathbf{x})^T \Omega_{ij} r_{ij}(\mathbf{x})} < \chi^2_{\alpha,d_C}$$

$$D_G^2(\mathbf{x}) = D_C^2(\mathbf{x}) + \boxed{\sum_{(i,j)\in S} r_{ij}(\mathbf{x})^T \Omega_{ij} r_{ij}(\mathbf{x})} < \chi^2_{\alpha,d_G}$$

Universidad Zaragoza 1542

Active SLAM : autonomously constructing and refining the environment representation with mobile robots

Universidad Zaragoza

Henry Carrillo

Acknowledgment

- **Joint work with (chronological order):**
  - Prof. José A. Castellanos – University of Zaragoza
  - Prof. Ian Reid – University of Oxford / Adelaide
  - Prof. José Neira – University of Zaragoza
    - Yasir Latif
  - Prof. Udo Frese – DFKI / Univ. Bremen
    - Oliver Birbach
  - Prof. Vijay Kumar – University of Pennsylvania
    - Philip Dames

"IF I HAVE SEEN FURTHER, IT IS BY STANDING ON THE SHOULDERS OF GIANTS."

- **Funding:**

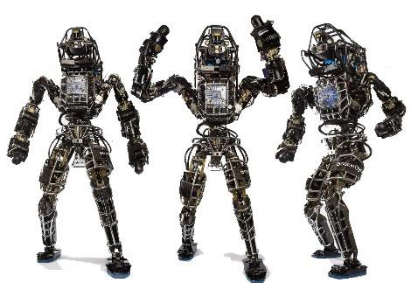
  - This work has been supported by the MINECO-FEDER projects DPI2009-13710 and DPI2012-36070, research grants BES-2010-033116 and EEBB-2011-44287, and also by DGA-FSE (grupo T04).

Motivation(I) :: How mobile robotics can be at your service?

- Mobile robots can **replace humans** in repetitive, hazardous or boring task, freeing them to enjoy life!
- **In factories ::**

- **Boring tasks ::**

- **In hazardous environments ::**



- **Freeing humans…**

Motivation(II) ::
Mobile robots need to be autonomous

- Three **necessary but not sufficient** ingredients for robot's autonomy are ::
  - Mapping – (How does the *world* look like?)
  - Localization – (Where "in the *world*" am I?)
  - Path and trajectory planning – (How do I get there?)
- Providing the robots with (i) the ability to **learn maps of environments** and (ii) enable them to **use the maps**.
  - Mapping+Localization == SLAM
    - **Is SLAM solved?**
  - Mapping+Localization+Path == Active SLAM
    - **Not solved….yet!**


Künstl Intell (2010) 24: 255–257
DOI 10.1007/s13218-010-0047-x

INTERVIEW

Interview: Is SLAM Solved?

Udo Frese

Published online: 29 July 2010
© Springer-Verlag 2010

Sebastian Thrun studied computer science at University of Hildesheim and University of Bonn, where he also obtained his Dr. rer. nat. degree. Until 2003 he was associate professor at Carnegie Mellon University. Since 2003 he is associate professor of computer science and electrical engineering at Stanford University (since 2007 full professor) and since 2004 director of the Stanford Artificial Intelligence Laboratory (SAIL). In 2007 he was elected into the German Academy of Sciences Leopoldina and the National Academy of Engineering.

Sebastian Thrun played a leading role in establishing a probabilistic paradigm in robotics. In 2005 he became known to the general public as head of the Stanford Racing Team that won the DARPA grand challenge, a 212 km race for driverless vehicles.

He contributed two novel probabilistic formulations to SLAM using Monte-Carlo samples (particle filter) and a so-called information matrix respectively. This constituted a radical change since at that time the field was exclusively taking a covariance matrix based view.

Sebastian Thrun is now distinguished engineer at Google, Inc. where he co-developed Google Street View.

José Neira studied computer science at the Universidad de los Andes, Bogotá. He obtained his Ph.D. degree from the University of Zaragoza, Spain in 1993. He is currently an Associate Professor with the Department of Computer Science and Systems Engineering, University of Zaragoza.

José Neira has been working in the SLAM field since that topic appeared in the early 1990th. Among his contributions are Joint Compatibility Branch and Bound (JCBB), a probabilistic algorithm for data-association, i.e. for determining to which feature an observation corresponds to. This algorithm is still considered the gold-standard for data-association.

In 2008 José Neira co-edited the special issue on Visual SLAM of the IEEE Transactions on Robotics.

U. Frese (✉)
DFKI GmbH, Bremen, Germany
e-mail: udo.frese@dfki.de

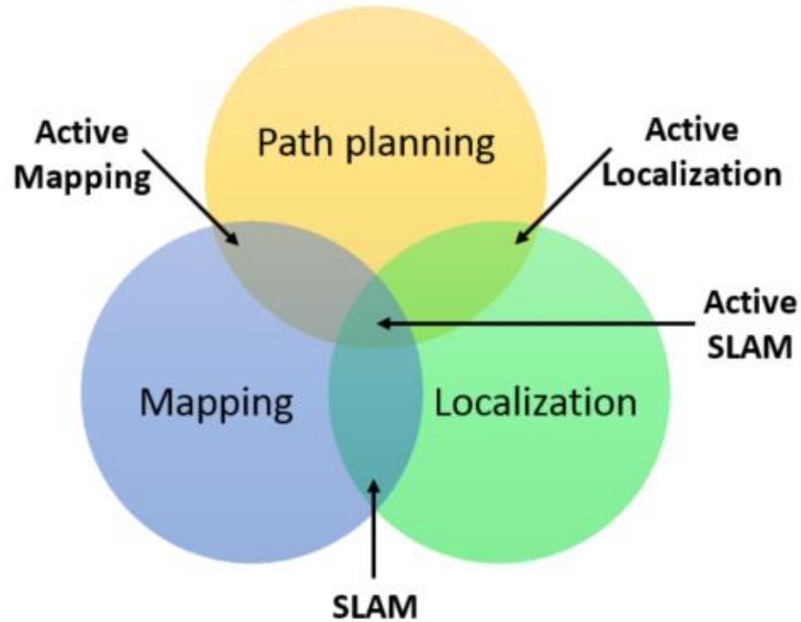
Springer

Active SLAM (I)

- SLAM does not define the path the robot has to follow.
  - Usually: random or predefined.

- Active SLAM => To integrate path planning into a SLAM process.
  - To **explore** more area or **refine** existing one.
    - Reduce uncertainty.
    - Navigate safely.

- First algorithms ::
  - 1° Alg. [Feder, Leonard](99)
    - Refining the map
    - Active perception [Bajacksy](86)
  - [Makarenko](02)
    - Entropy for exploration
  - [Huang, Dissanayake](05,06)
    - Control theory and MPC
    - Coined the term



Universidad Zaragoza

Active SLAM (II)

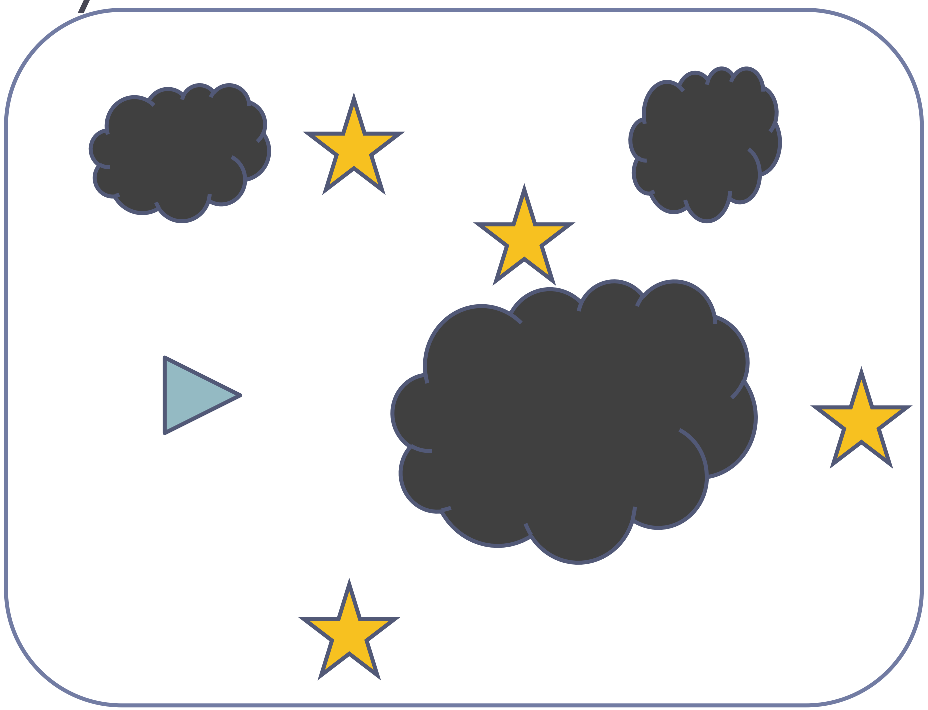
- <u>**General Active SLAM Pseudo-code:**</u>

Require: *A priori* partially known map.

- Select a set of trajectories π^{s}
- Assign a score to each trajectory

$$\mathcal{J} = \sum_i \alpha_i \mathcal{U}_i + \sum_i \beta_i \mathcal{T}_i$$

  - Uncertainty map+robot: $\mathcal{U}_i$
  - Trajectory cost: $\mathcal{T}_i$
- Execute the trajectory with the optimum $\mathcal{J}$.



Active SLAM (II)

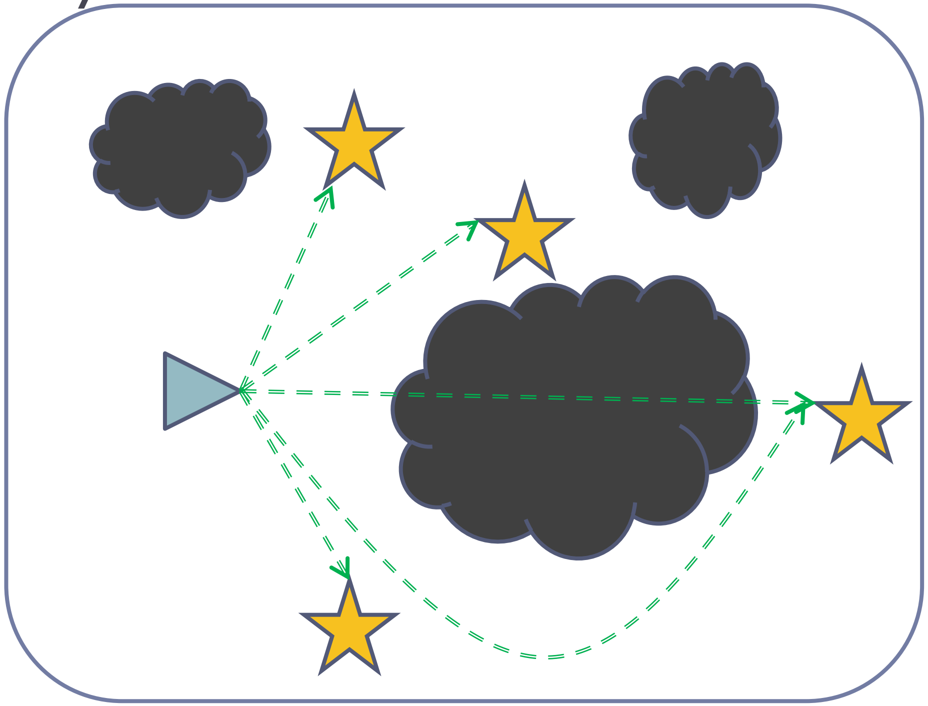
- <u>**General Active SLAM Pseudo-code:**</u>

Require: *A priori* partially known map.

- Select a set of trajectories π^{s}
- Assign a score to each trajectory

$$\mathcal{J} = \sum_i \alpha_i \mathcal{U}_i + \sum_i \beta_i \mathcal{T}_i$$

  - Uncertainty map+robot: $\mathcal{U}_i$
  - Trajectory cost: $\mathcal{T}_i$
- Execute the trajectory with the optimum $\mathcal{J}$.



Active SLAM (II)

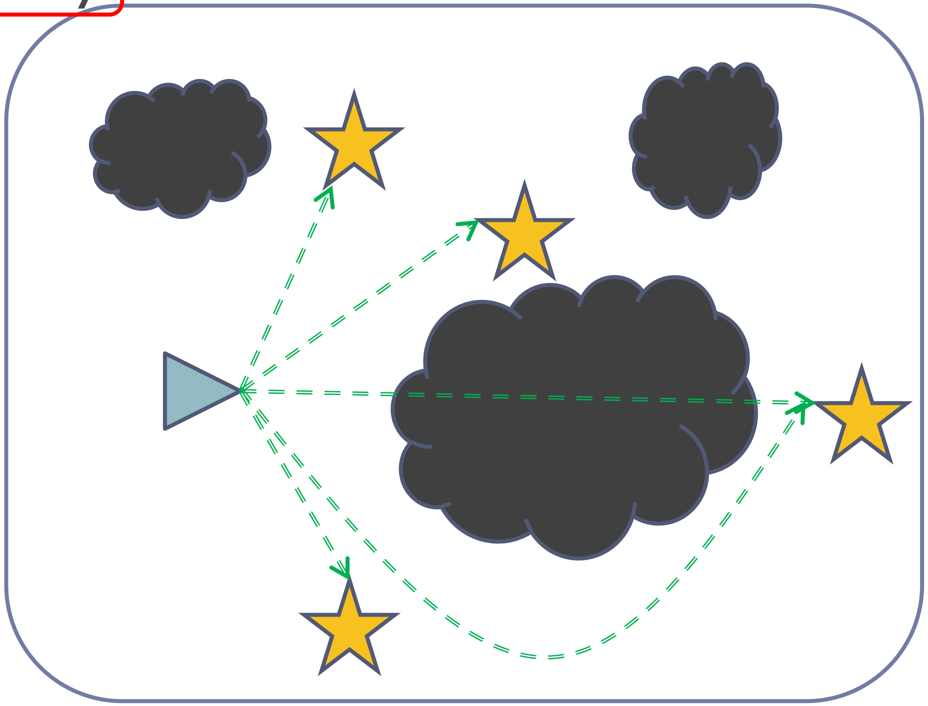
- **General Active SLAM Pseudo-code:**

Require: *A priori* partially known map.

- Select a set of trajectories π^{s}
- Assign a score to each trajectory

$$\mathcal{J} = \sum_i \alpha_i \mathcal{U}_i + \sum_i \beta_i \mathcal{T}_i$$

  - Uncertainty map+robot: $\mathcal{U}_i$
  - Trajectory cost: $\mathcal{T}_i$
- Execute the trajectory with the optimum $\mathcal{J}$.



J1	J2	J3	J4	J5
1	1,5	1,9	0,8	3

Active SLAM (II)

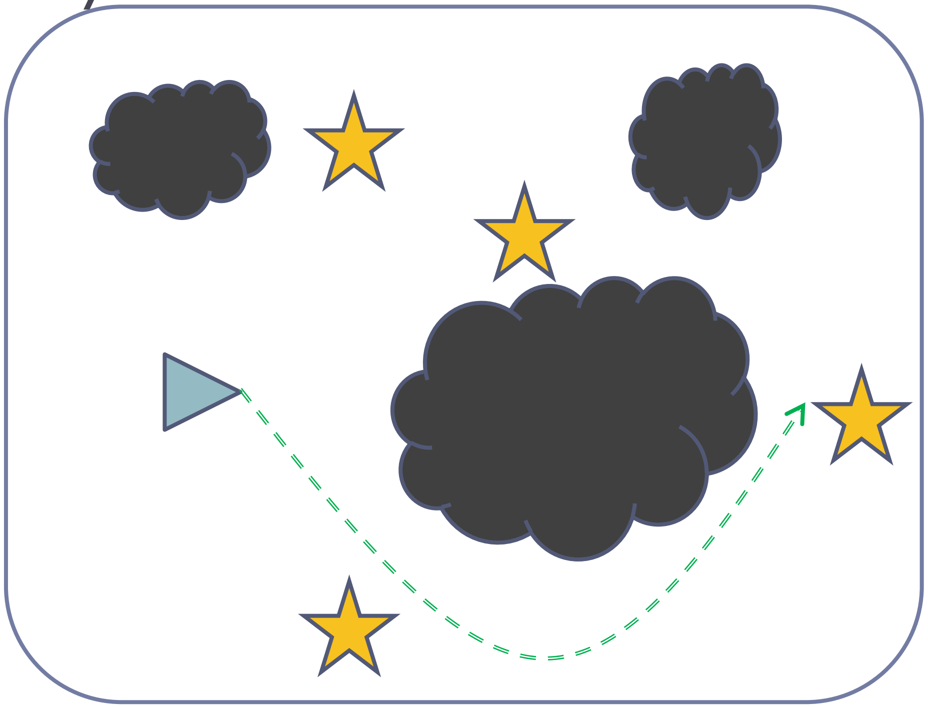
- **General Active SLAM Pseudo-code:**

Require: *A priori* partially known map.

- Select a set of trajectories π^s
- Assign a score to each trajectory

$$\mathcal{J} = \sum_i \alpha_i \mathcal{U}_i + \sum_i \beta_i \mathcal{T}_i$$

  - Uncertainty map+robot: $\mathcal{U}_i$
  - Trajectory cost: $\mathcal{T}_i$
- Execute the trajectory with the optimum $\mathcal{J}$.



J1	J2	J3	J4	J5
1	1,5	1,9	0,8	3

Uncertainty Criteria for Active SLAM (I)

- A key part of an active SLAM algorithm is to measure the **uncertainty map+robot** $\mathcal{U}_i$ associated to a trajectory $\boldsymbol{\pi}_i$.
- Measurement of Uncertainty => • Theory of Optimal Experiment Design (TOED). A framework.
- The idea is to quantify the uncertainty associated to a trajectory:

$$\boldsymbol{\phi}: \boldsymbol{Cov}(\boldsymbol{\pi}_i) \longrightarrow \mathbb{R}$$

  - Easy way to compare designs (*i.e.* $\boldsymbol{\pi}_1$).
  - **This function is the so-called uncertainty criterion.**

Appx. of D-opt	Appx. of A-opt	
$\det(\Sigma) = \prod_{k=1,\dots,l} \lambda_k$	$\mathrm{trace}(\Sigma) = \sum_{k=1,\dots,l} \lambda_k$	$\max(\lambda_1, \dots, \lambda_k)$
Determinant (D-opt)	Trace (A-opt)	Max (E-opt)

Uncertainty Criteria for Active SLAM (II)

- Previous works ([Sim and Roy, 2005], [Mihaylova, 2003]) report **A-opt** as the best criterion and that **D-opt gives null values**.
  - **A-opt**, widely used, **as 2012**: [Kollar2008] [Martinez-Cantin2008] [Meger2008] [Leung2006].
  - **Although D-opt is commonly used in the TOED because it is optimal…also implicitly in Shannon's Entropy**

$$\det(\Sigma) = \prod_{k=1,\dots,l} \lambda_k$$

Determinant (D-opt)

$$\mathrm{trace}(\Sigma) = \sum_{k=1,\dots,l} \lambda_k$$

Trace (A-opt)

$$\max(\lambda_1, \dots, \lambda_k)$$

Max (E-opt)

Uncertainty Criteria for Active SLAM (II)

- Previous works ([Sim and Roy, 2005], [Mihaylova, 2003]) report **A-opt** as the best criterion and that **D-opt gives null values**.
  - **A-opt**, widely used, **as 2012**: [Kollar2008] [Martinez-Cantin2008] [Meger2008] [Leung2006].
  - **Although D-opt is commonly used in the TOED because it is optimal…also implicitly in Shannon's Entropy**

$$\cancel{\det(\Sigma) = \prod_{k=1,\dots,l} \lambda_k}$$

Determinant (D-opt)

$$\mathrm{trace}(\Sigma) = \sum_{k=1,\dots,l} \lambda_k$$

Trace (A-opt)

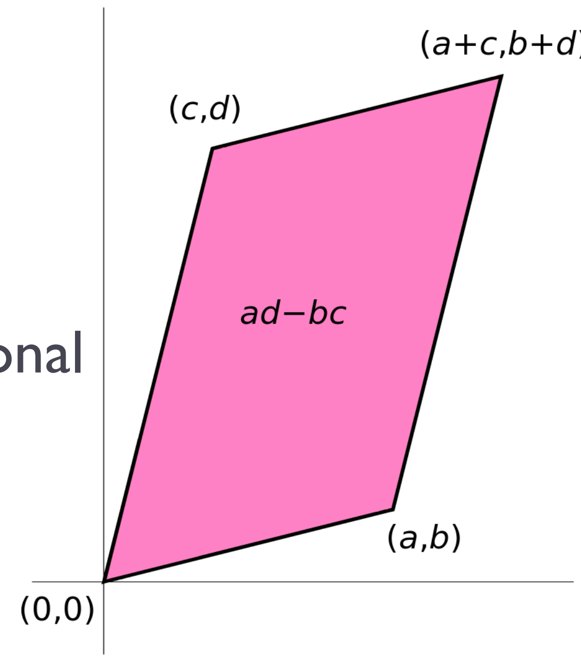
$$\max(\lambda_1, \dots, \lambda_k)$$

Max (E-opt)

Determinant

- An associated value of a squared matrix.
  - Map a matrix to a scalar.
- Properties for a <u>**n x n**</u> matrix:
  - Geometrically: The hyper-volume of the parallelepiped defined in the n-dimensional space.
  - <u>***homogeneous of grade n.***</u>
  - If, $f: u \rightarrow v$

$$f(\alpha u) = \alpha^n f(v)$$



Uncertainty Criteria for Active SLAM (III)

- **It is indeed possible to use D-opt in the Active SLAM context:**
  - The structure of the problem needs to be taken into account (*i.e. The covariance matrix varies with time*).
  - It is not informative to compare the determinant of a matrix ***l*** x ***l*** with a ***m* x *m*. => *det(l x l) is homogeneous of grade l.***
  - The computation of the determinant of a highly correlated matrix (e.g. SLAM) is prone to round-off errors. **Processing it in the logarithm space**
- **Proposed D-opt for a *l x l* covariance matrix:**

$$\exp(\log([\det(\Sigma(\pi))]^{1/l})) = \exp\left(l^{-1}\sum_{k=1}^{l}\log(\lambda_k)\right)$$

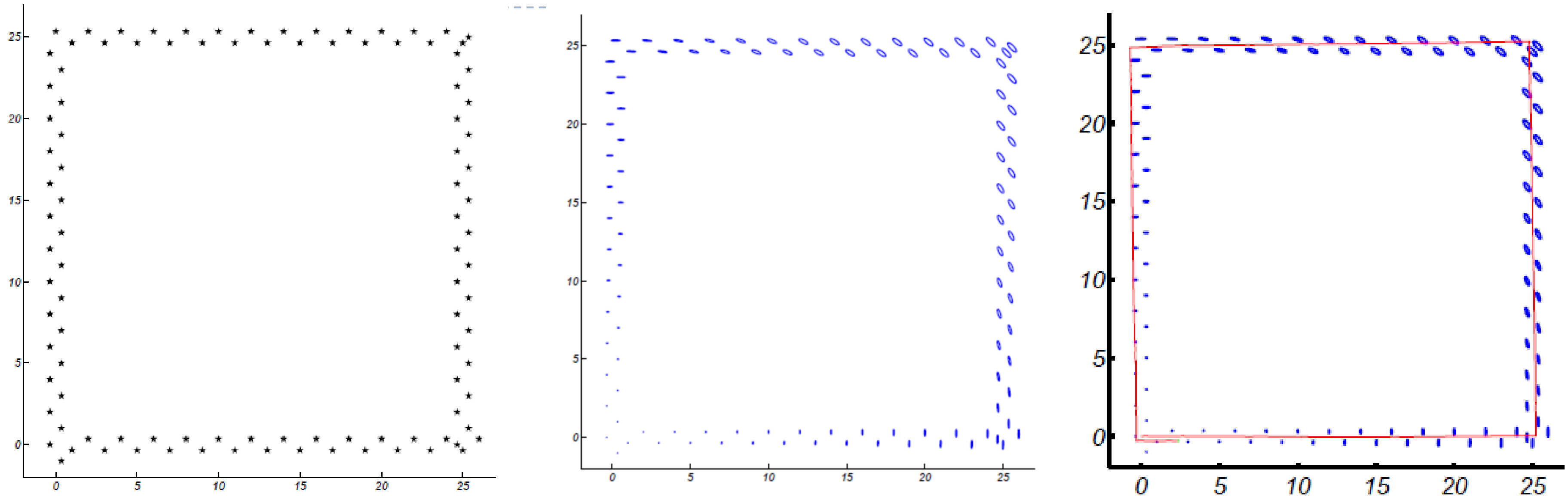
- Stems from [Kiefer, 1974] : $\phi_p(\xi) = [l^{-1}\mathrm{trace}(\Sigma^p(\xi))]^{1/p}$

First experiment

- First experiment: on the computation
  - Is it possible to compute D-opt from a robot doing SLAM?
  - Execute a SLAM algorithm (e.g. EKF-SLAM, Graph SLAM).
  - Compute in each step: A-opt, E-opt and D-opt,.

- Simulated Robot indoor environment : MRPT/C++
- Real Robot indoor environment : Pioneer 3 DX - *Ad-hoc*
- Real Robot indoor environment : DLR *dataset*
- Real Robot outdoor environment :Victoria Park *dataset*

1E - Simulated Robot indoor environment (I)



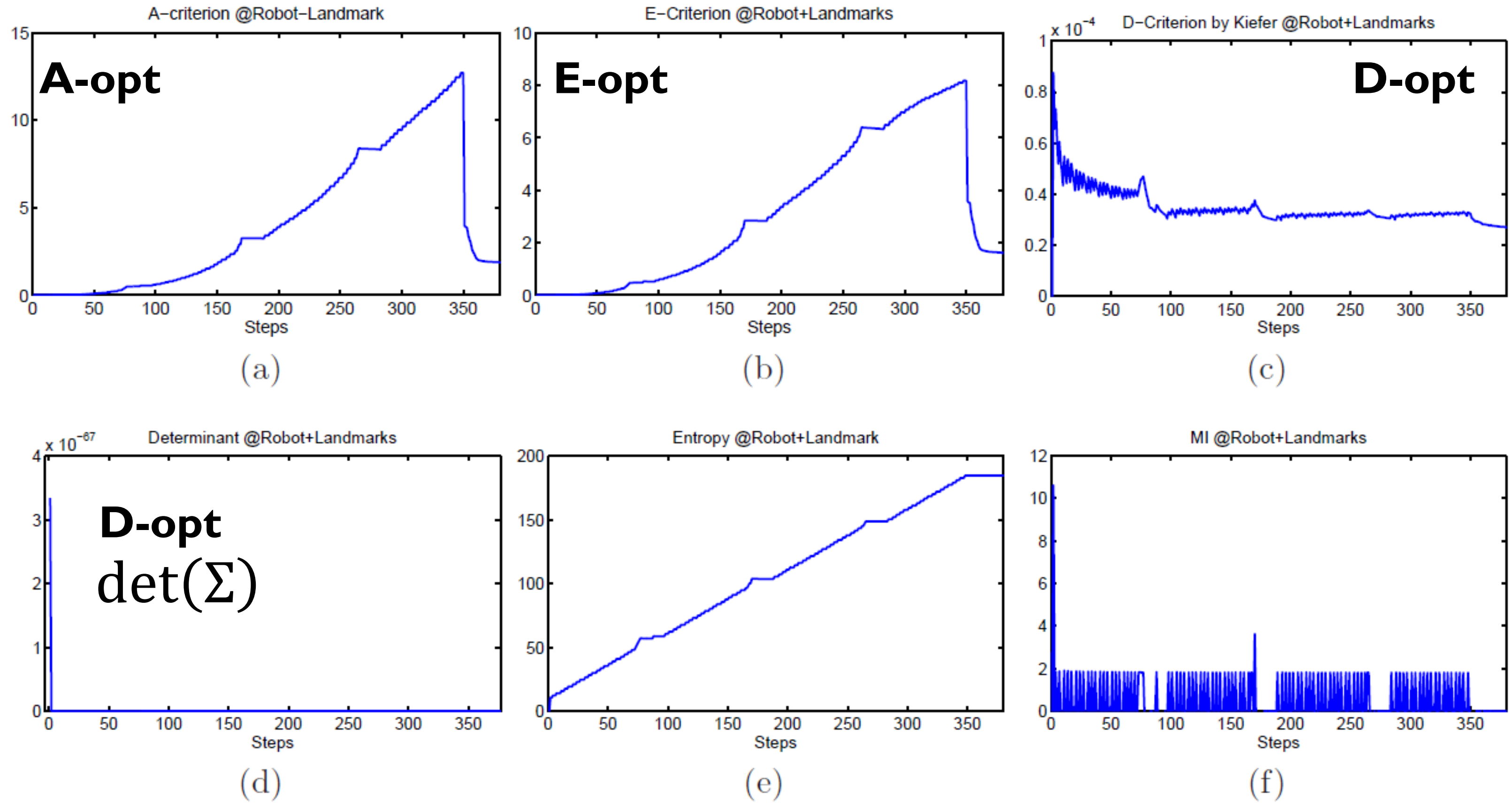
Scenario:

- Area of 25x25 m
- 2D EKF-SLAM
- Sensor: Odometry + Camera (360° - 3m range)
- 180 *landmarks* - DA Known.
- Gaussian errors:

  Odometry + Camera

Universidad Zaragoza 1542

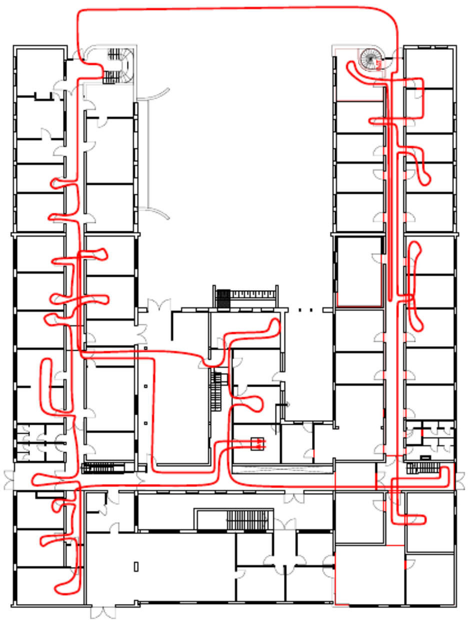
1E-Simulated Robot indoor environment (II) Qualitative results

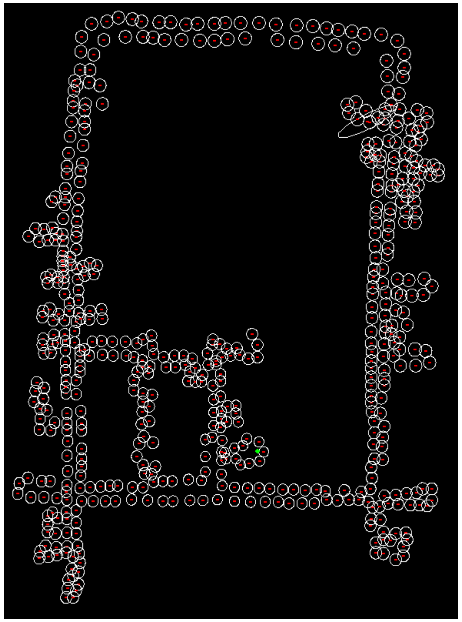


(a)-(f) A-opt, E-opt, D-opt, determinant, entropy and MI.

Universidad Zaragoza

1E-Real Robot indoor environment @ DLR

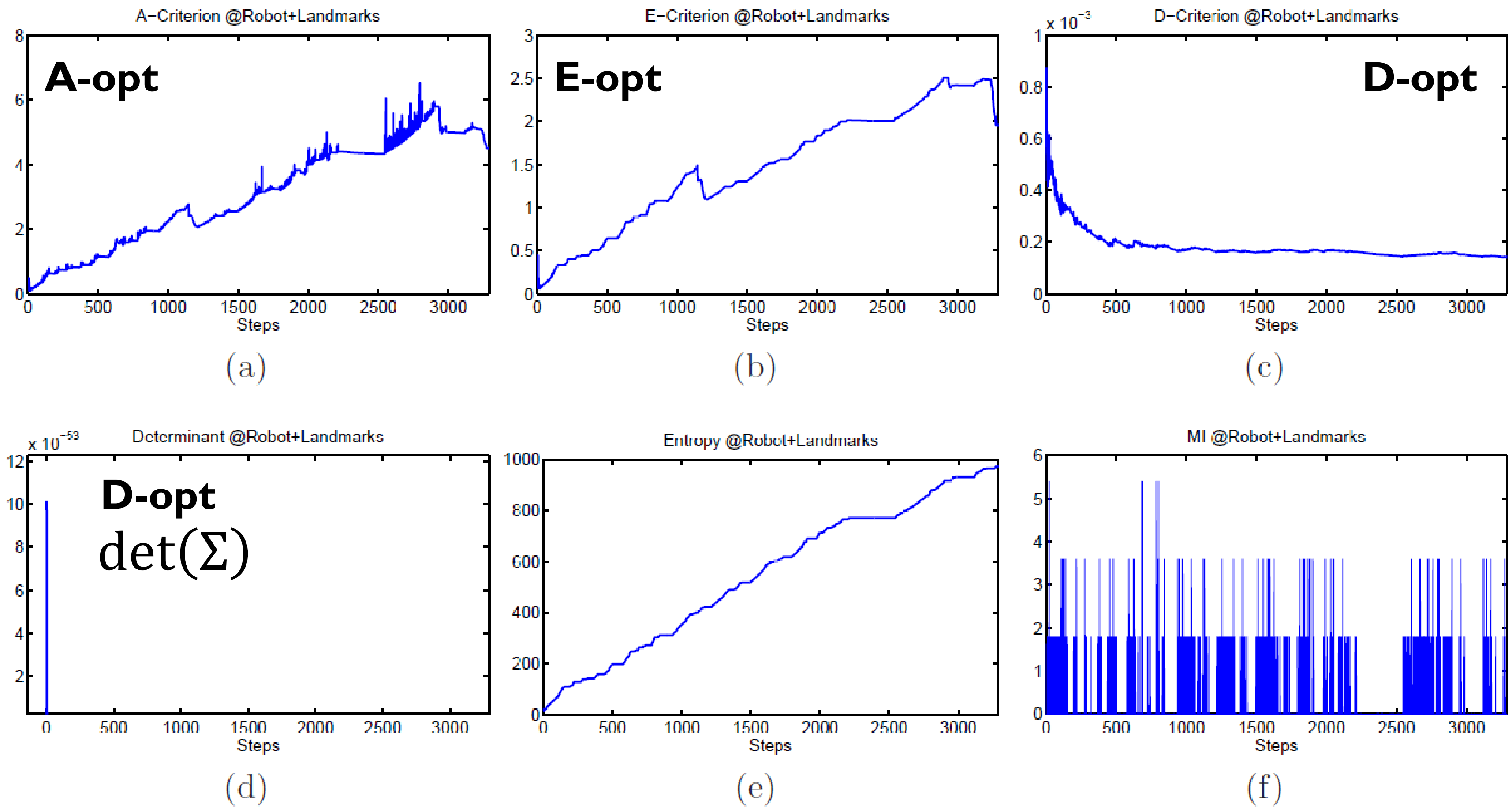




Scenario:

- Area 60x40 m
- Sensor: Odometry + Camera
- 2D EKF-SLAM
- 576 *landmarks* – DA known.

1E-Real Robot indoor environment @ DLR Qualitative results



(a)-(f) A-opt, E-opt, D-opt, determinant, entropy and MI.

Universidad Zaragoza

First experiment – Quantitative analysis

- Average correlation between the uncertainty criteria:

	A-opt	E-opt	D-opt
A-opt	1	0,9872	0,6003
E-opt	0,9872	1	0,5903
D-opt	0,6003	0,5903	1

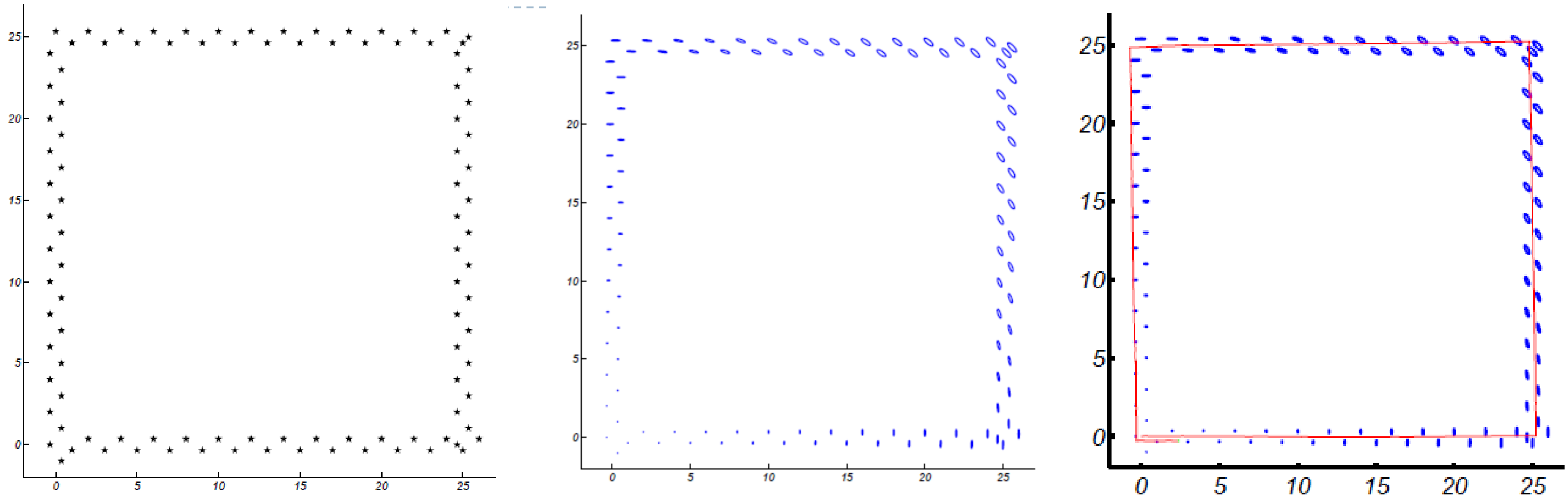
- Variance: A-E (0,0002) / A-D (0,0540) / D-E (0,0481).
- A-opt y E-opt => High correlation.
  - E-opt is guided by a single eigenvalue.
- A-opt y D-opt => Medium correlation.
  - D-opt take into account more components than A-opt.

Second Experiment

- Second experiment: Active SLAM
  - What is the effect of the uncertainty criteria in active SLAM?
  - Active SLAM => Unitary horizon (*greedy*).
  - Uncertainty criteria => A-opt, D-opt and Entropy.
  - Effect => MSE y χ^2.

- Simulated Robot with unitary horizon: MRPT / C++

2E-Simulated Robot indoor environment (I)

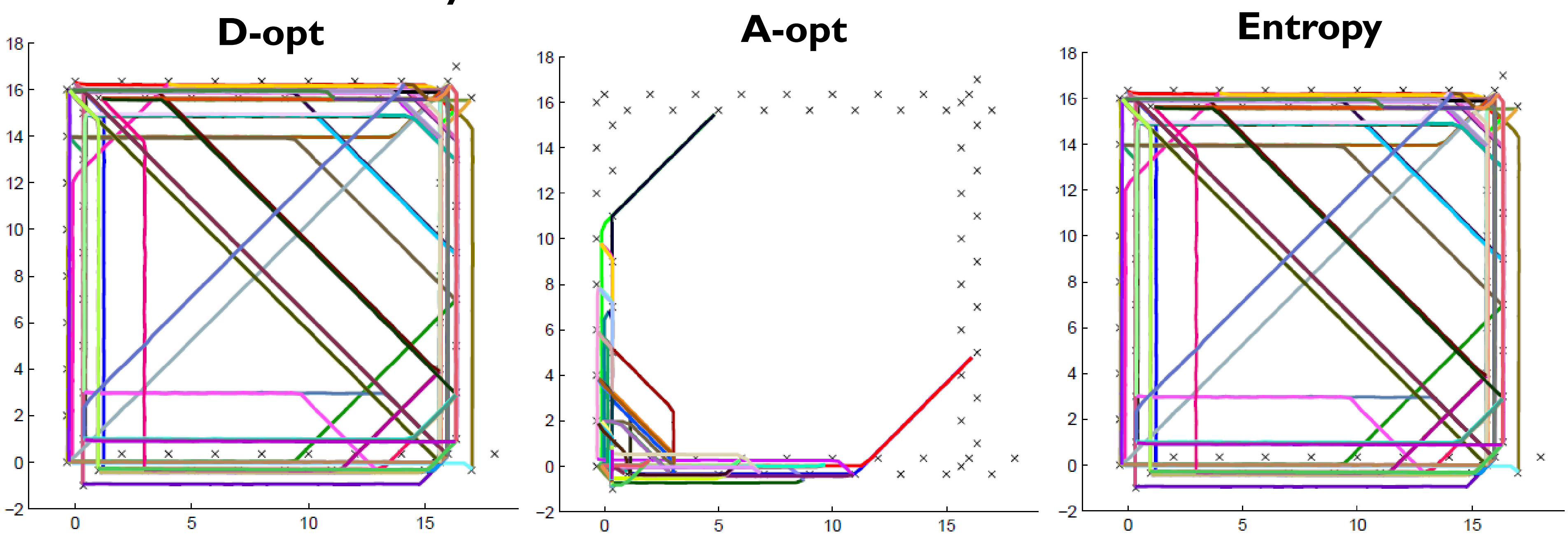


Scenario:

- Area of 20x20m and 30x30m
- 2D EKF-SLAM
- Sensor: Odometry + Camera (360° - 3m range)
- Gaussian errors: Odometry + sensors.
- Path planner: Discrete (A*) and continuous (Attract-Repel).

2E-Simulated Robot indoor environment (II)

- Qualitative analysis



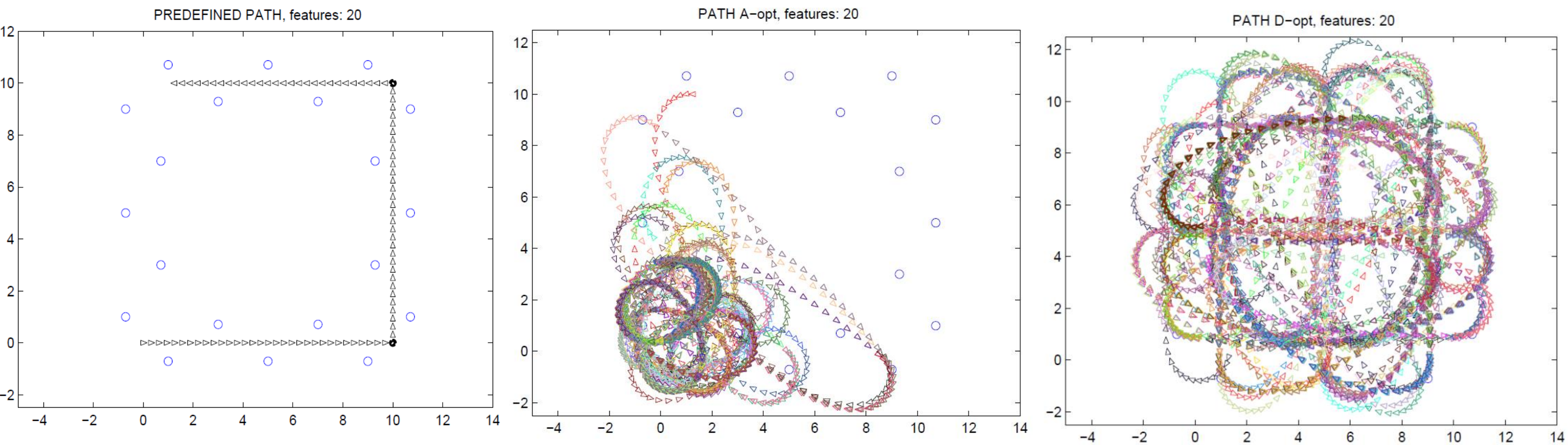
- Resulting paths for each uncertainty criterion: (a) *D-opt*, (b) *A-opt* y (c) Entropy. Each colour represents an executed path. 20 x 20 m map. ***VIDEO***

Universidad Zaragoza

2E-Simulated Robot indoor environment (III)

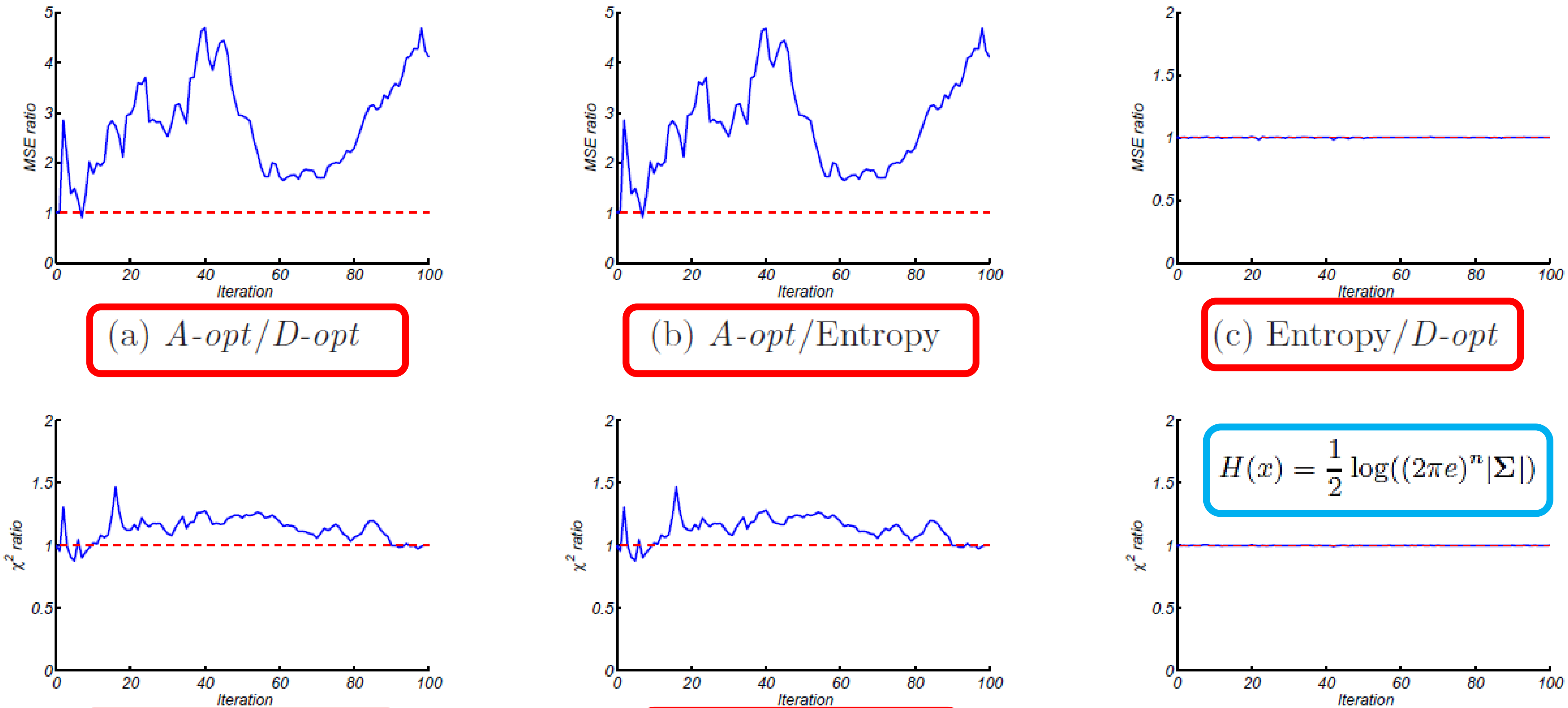
- Qualitative analysis.

A-opt **D-opt**



- Resulting trajectories for 10000 steps active SLAM simulation. (a). Initial trajectory. (b) *A-opt.* (c). *D-opt.*

2E – Quantitative Analysis 30x30 m



(a) *A-opt*/*D-opt* (b) *A-opt*/Entropy (c) Entropy/*D-opt*

(d) *A-opt*/*D-opt* (e) *A-opt*/Entropy (f) Entropy/*D-opt*

$$H(x) = \frac{1}{2}\log((2\pi e)^n|\Sigma|)$$

- Evolution of MSE ((a)-(c)) y chi2 ((d)-(f)) ratio. <u>**Average of 10 MC simulations**</u>. 30x30 m.

Take home message

- D-opt is the optimum criterion to measure uncertainty according to the TOED (*i.e.* better than A-opt (Trace)).

- It is possible to obtain useful information regarding the uncertainty of a SLAM process with D-opt.

- D-opt shows better performance than A-opt in our simulated experiments of active SLAM.

- To compute D-opt in the context of a SLAM process => use the formulation presented here.

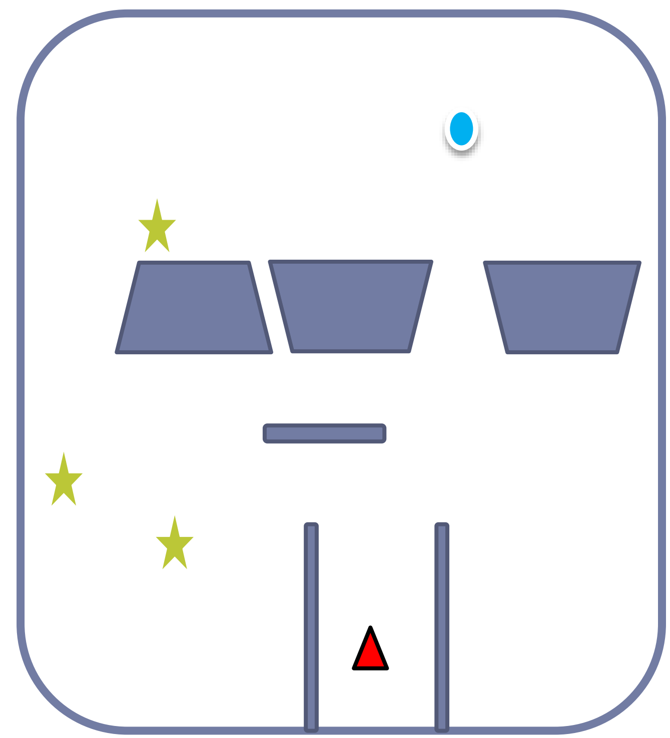
Fast Minimum Uncertainty Search on a Graph Map Representation

Universidad Zaragoza

Henry Carrillo

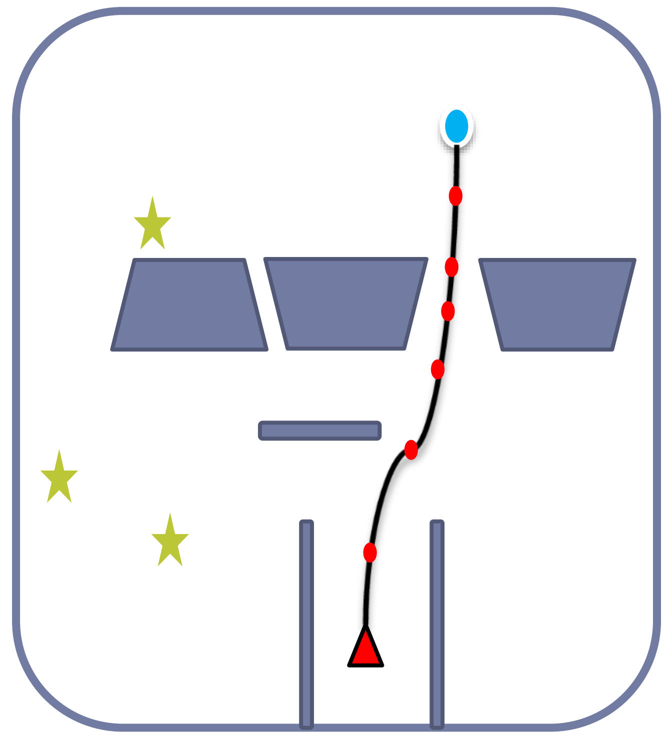
What is the minimum uncertainty path in a roadmap? (I)

- **Task** : Go safely from A [▲] to B [●]
  - Obstacles
  - Robot 6 D.O.F
  - Localization with **noisy sensors**
  - Beacons [★]



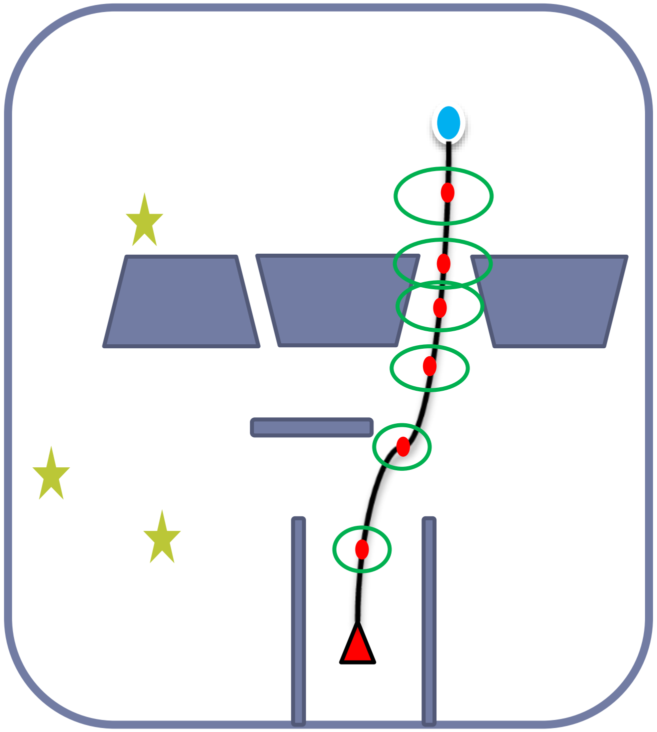
What is the minimum uncertainty path in a roadmap? (I)

- **Task** : Go safely from A [▲] to B [●]
  - Obstacles
  - Robot 6 D.O.F
  - Localization with **noisy sensors**
  - Beacons [★]
- **A solution:** Sampling-based path planning
  - PRM
    - Configuration space ($\mathcal{C}$)
    - Known environment
    - Shortest path in the roadmap



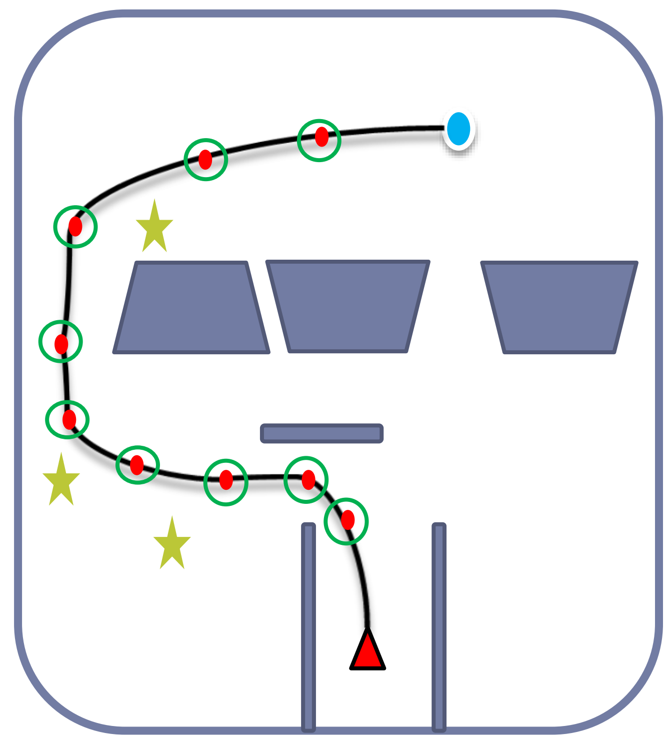
What is the minimum uncertainty path in a roadmap? (I)

- **Task** : Go safely from A [▲] to B [●]
  - Obstacles
  - Robot 6 D.O.F
  - Localization with **noisy sensors**
  - Beacons [★]
- **A solution:** Sampling-based path planning
  - PRM
    - Configuration space ($\mathcal{C}$)
    - Known environment
    - Shortest path in the roadmap
  - **BUT IT IS NOT SAFE**
    - Uncertainty in the localization

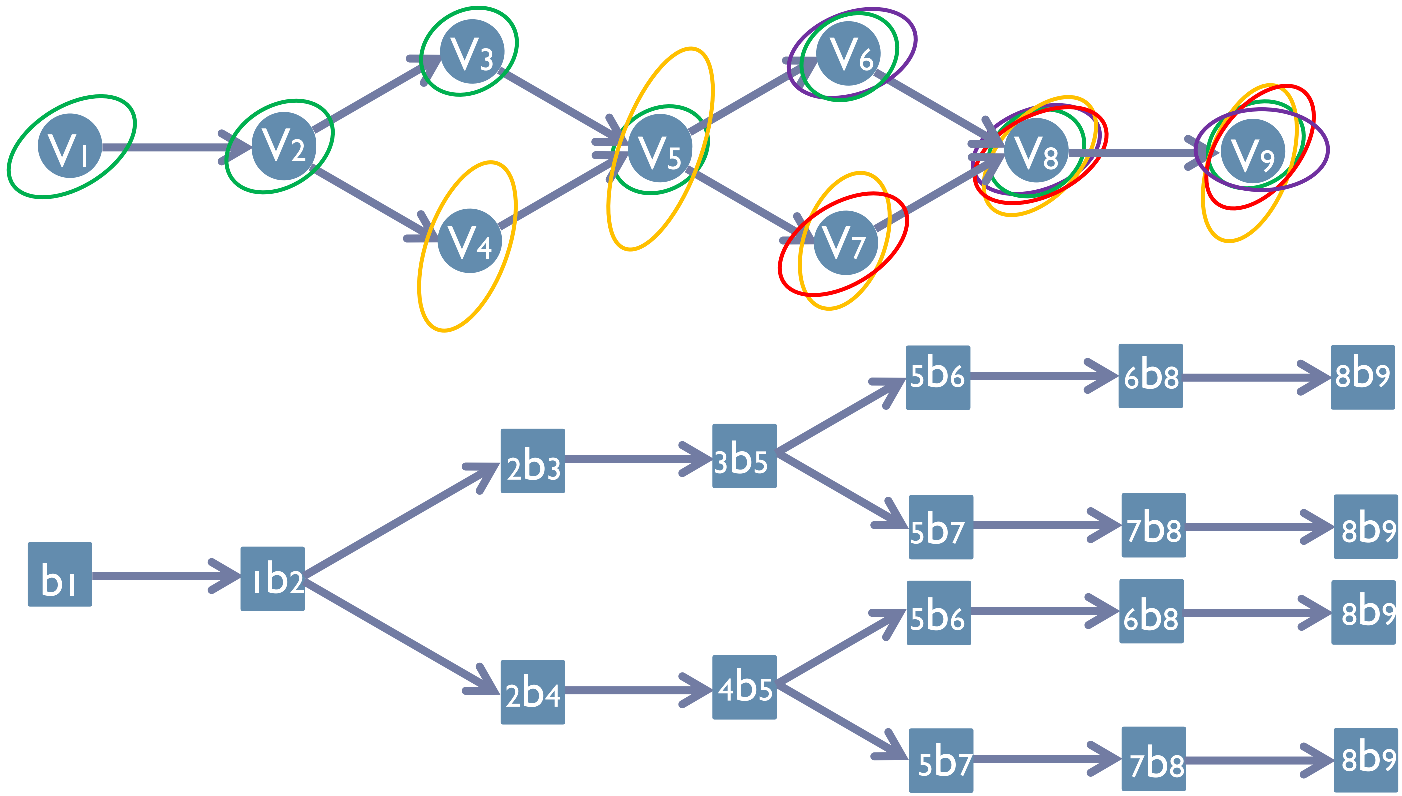


What is the minimum uncertainty path in a roadmap? (II)

- **A safer solution:** Belief space+Sampling-based path planning
  - BRM (Belief RoadMap) [Prentice and Roy 2008]
  - Belief space ($\mathcal{B}$)
    - Known environment
    - Minimum uncertainty path
  - **BUT**
    - Known environment
    - Slow ("Curse of history")

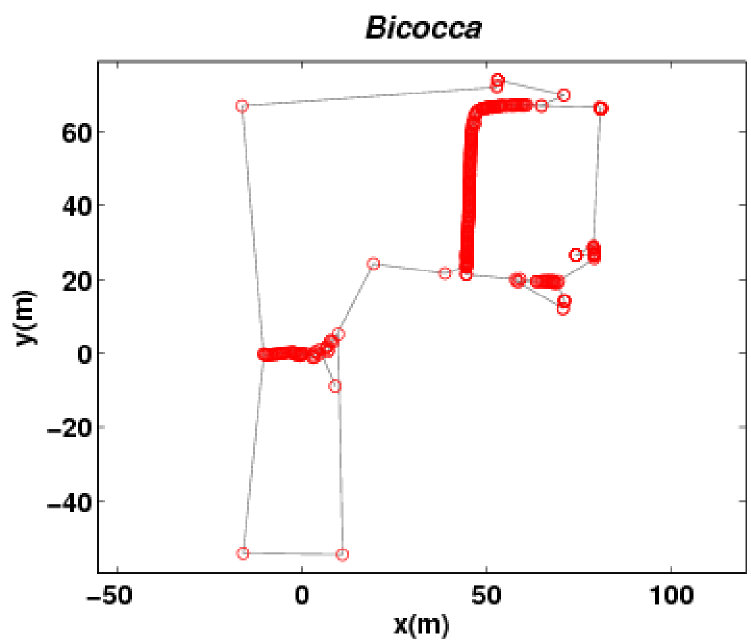


“Curse of history” in the belief space



Fast Minimum Uncertainty Search on a Graph Map Representation

- We propose **FaMUS** :
  - Concurrently build the map and search
    - Using graph based SLAM (e.g., iSAM, RSLAM, g2o)
    - Related work: "Path planning in belief space with Pose SLAM" [Valencia et al. 2011]
  - Fast planning by reducing the search space.



FaMUS: Fast Minimum Uncertainty Search

Algorithm 1 FaMUS algorithm

Require:

- A pose graph map of the environment **G**
- A initial pose n_s and a goal pose n_g.

Ensure:

- The path with the minimum accumulated uncertainty cost from the pose n_s to the pose n_g.

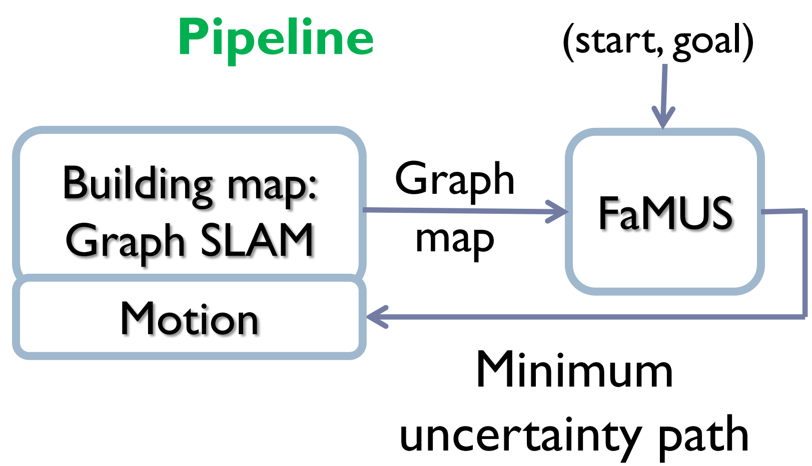
$\forall n_i \in \mathbf{G}$: calculate *D-opt*

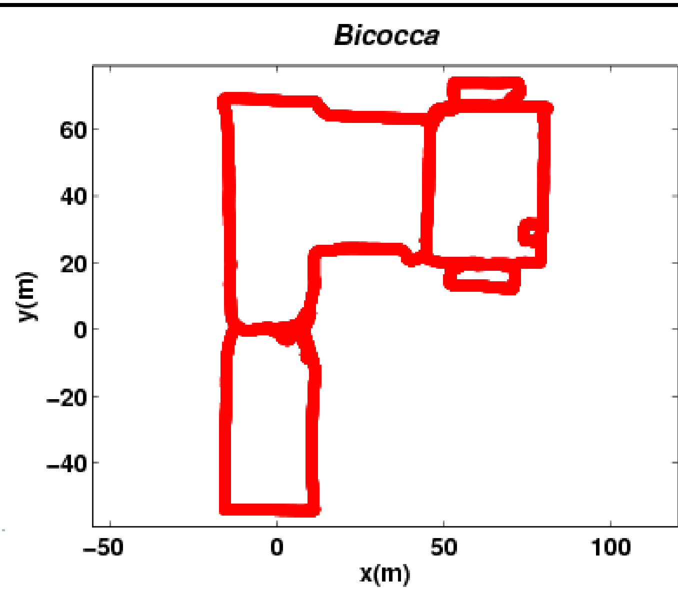
$\forall n_i \in \mathbf{G}$: find reachable neighbours and add edges

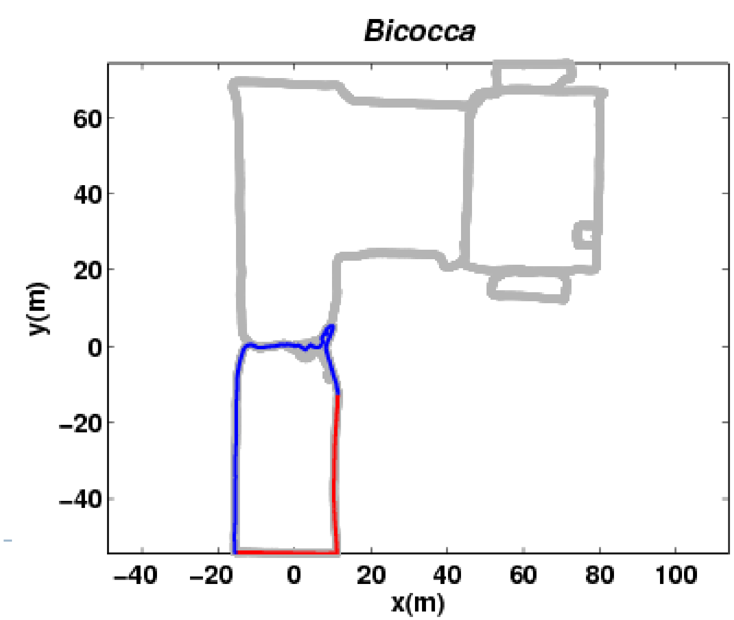
$\mathbf{G_d} \leftarrow$ ReduceGraph $(\mathbf{G})$

$minPath_d \leftarrow$ DijsktraSearch$(n_s, n_g, \mathbf{G_d})$

return ReconstructPath$(minPath_d, \mathbf{G_d})$







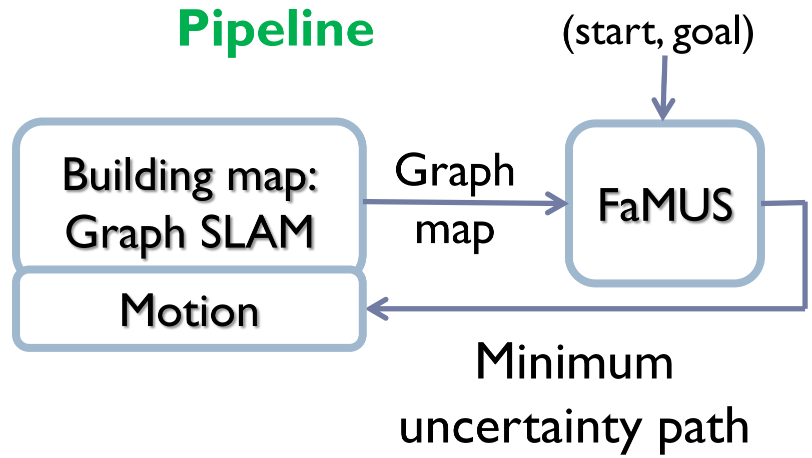
FaMUS: Fast Minimum Uncertainty Search

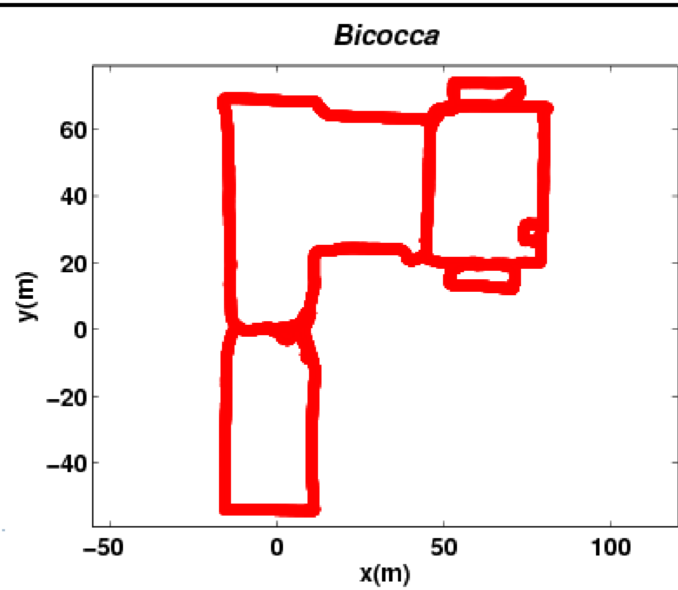
Algorithm 1 FaMUS algorithm

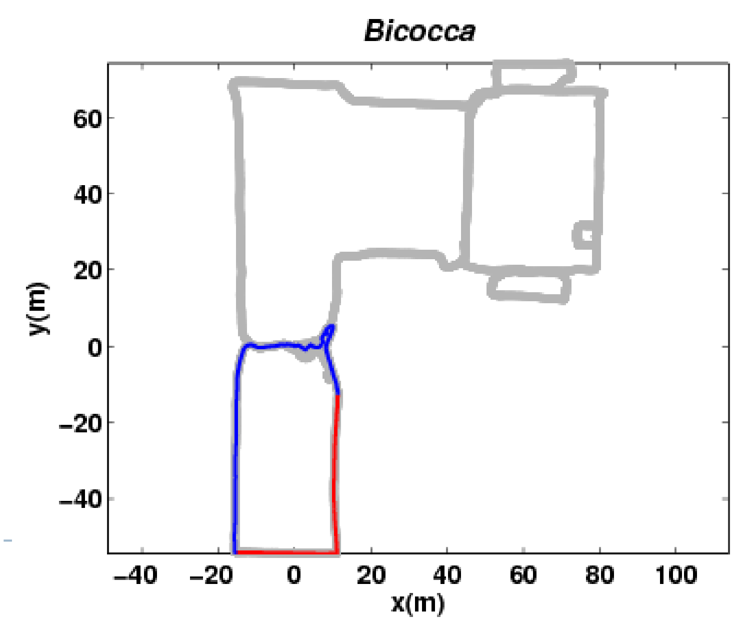
Require:
- A pose graph map of the environment $\mathbf{G}$
- A initial pose n_s and a goal pose n_g.

Ensure:
- The path with the minimum accumulated uncertainty cost from the pose n_s to the pose n_g.

```
∀n_i ∈ G : calculate D-opt
∀n_i ∈ G : find reachable neighbours and add edges
G_d ← ReduceGraph (G)
minPath_d ← DijsktraSearch(n_s,n_g,G_d)
return ReconstructPath(minPath_d,G_d)
```







FaMUS: Fast Minimum Uncertainty Search

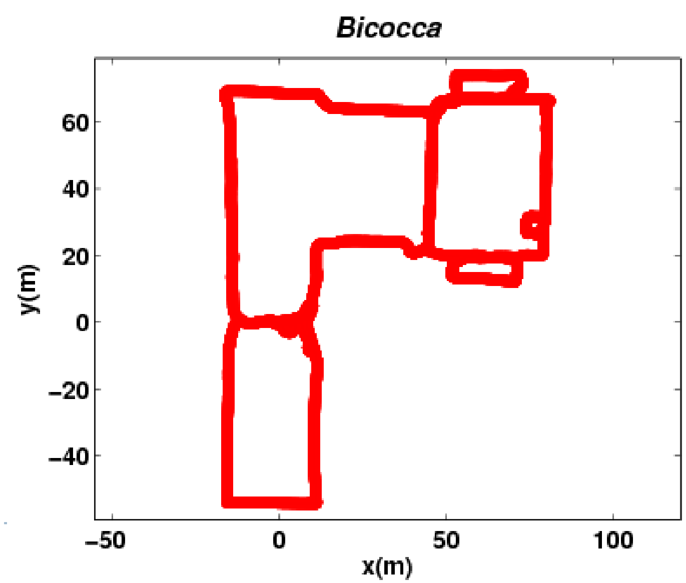
Algorithm 1 FaMUS algorithm

Require:
- A pose graph map of the environment $\mathbf{G}$
- A initial pose n_s and a goal pose n_g.

Ensure:
- The path with the minimum accumulated uncertainty cost from the pose n_s to the pose n_g.

$\forall n_i \in \mathbf{G}$: calculate *D-opt*
$\forall n_i \in \mathbf{G}$: find reachable neighbours and add edges
$\mathbf{G_d} \leftarrow$ ReduceGraph $(\mathbf{G})$
$minPath_d \leftarrow$ DijsktraSearch(n_s,n_g,$\mathbf{G_d}$)
return ReconstructPath($minPath_d$,$\mathbf{G_d}$)

- We measure the uncertainty at each node using D-opt.
  - A-opt [Prentice2008]
  - Entropy [Valencia2011]
- "On the Comparison of Uncertainty Criteria for Active SLAM" ICRA'12



SUMMARY OF UNCERTAINTY CRITERIA

Criterion	Classical Formulation	Modern Formulation
A-opt	$\text{trace}(\Sigma) = \sum_{k=1}^{l} \lambda_k$	$l^{-1}\text{trace}(\Sigma)$
D-opt	$\det(\Sigma) = \prod_{k=1}^{l} \lambda_k$	$\exp\left(l^{-1} \sum_{k=1}^{l} \log(\lambda_k)\right)$
E-opt	$max(\lambda_k)$	$max(\lambda_k)$

Universidad Zaragoza

FaMUS: Fast Minimum Uncertainty Search

Algorithm 1 FaMUS algorithm

Require:

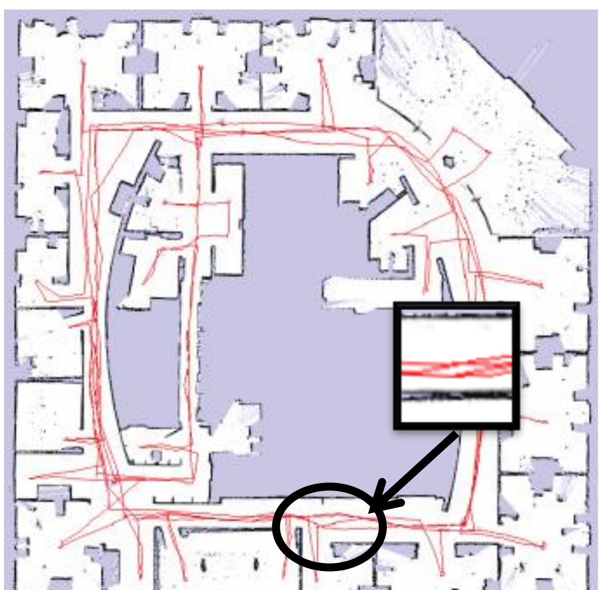
- A pose graph map of the environment **G**
- A initial pose n_s and a goal pose n_g.

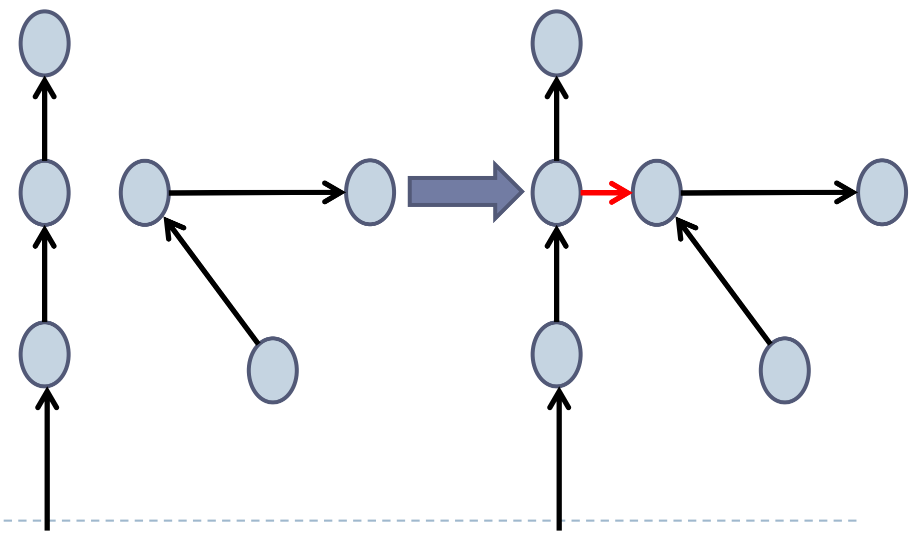
Ensure:

- The path with the minimum accumulated uncertainty cost from the pose n_s to the pose n_g.

```
∀n_i ∈ G : calculate D-opt
∀n_i ∈ G : find reachable neighbours and add edges
G_d ← ReduceGraph (G)
minPath_d ← DijsktraSearch(n_s,n_g,G_d)
return ReconstructPath(minPath_d,G_d)
```

- We increase the traversability by connecting near vertices.
- The roadmap is sparse mainly because of missed loop-closures.





FaMUS: Fast Minimum Uncertainty Search

Algorithm 1 FaMUS algorithm

Require:

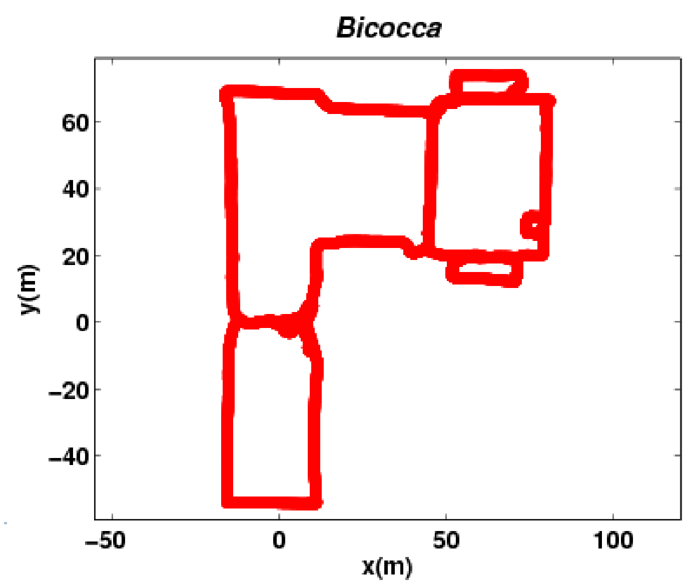
- A pose graph map of the environment **G**
- A initial pose n_s and a goal pose n_g.

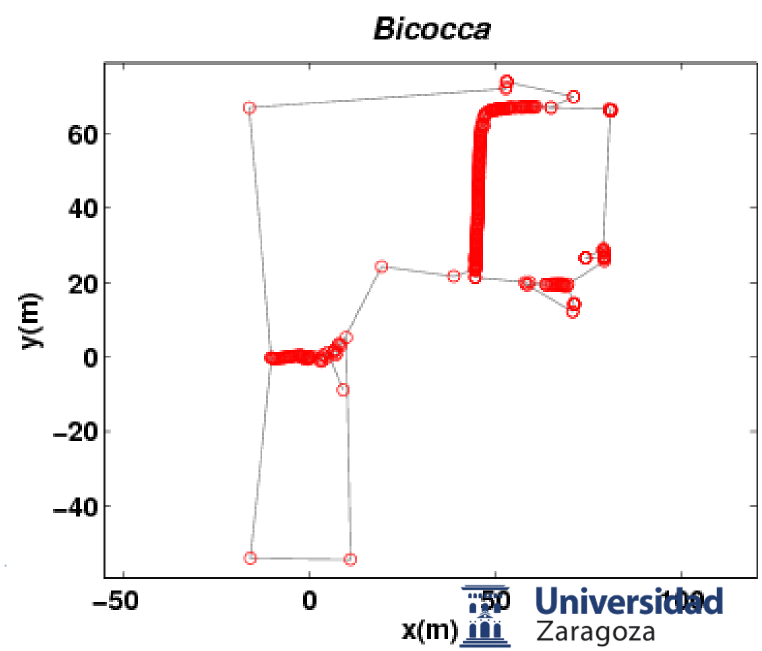
Ensure:

- The path with the minimum accumulated uncertainty cost from the pose n_s to the pose n_g.

$\forall n_i \in \mathbf{G}$: calculate *D-opt*
$\forall n_i \in \mathbf{G}$: find reachable neighbours and add edges
$\mathbf{G_d} \leftarrow$ ReduceGraph $(\mathbf{G})$
$minPath_d \leftarrow$ DijsktraSearch(n_s,n_g,$\mathbf{G_d}$)
return ReconstructPath($minPath_d$,$\mathbf{G_d}$)

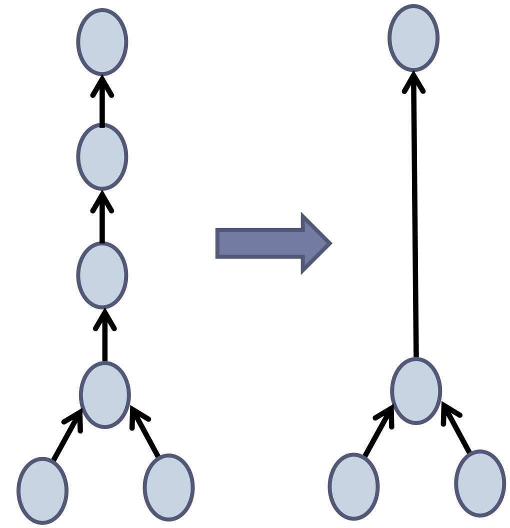
- We reduce the size of the roadmap by approximating it to a **decision graph**.





Decision graph

- Reduce the roadmap to a graph of "decision points"
- Properties
  - Form from vertices:
    - Initial and goal pose
    - Loop-closure
    - With connectivity more than 3
- ("~~Hard-to-buy~~") Assumptions
  - Static environment => no changes in sensors
  - Uncertainty is accumulated using worst case scenario
  - Under the assumptions the decision graph is provable equivalent to the full graph for path planning under uncertainty
  - Need to re-plan fast == SLAM in the background



FaMUS: Fast Minimum Uncertainty Search

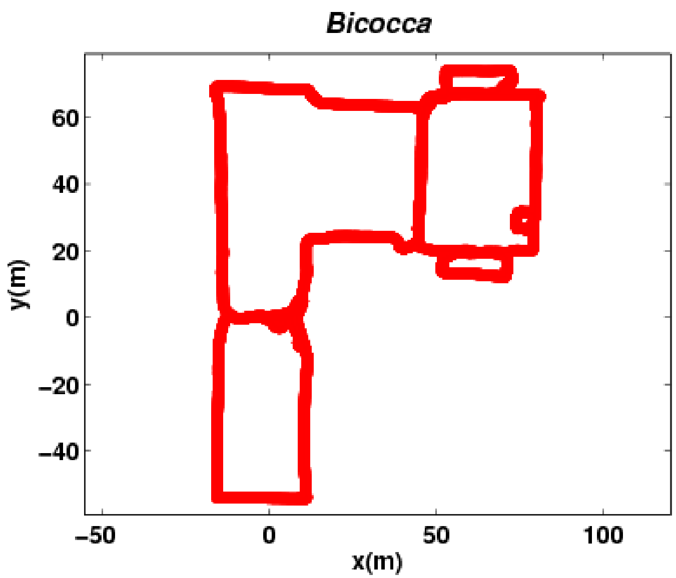
Algorithm 1 FaMUS algorithm

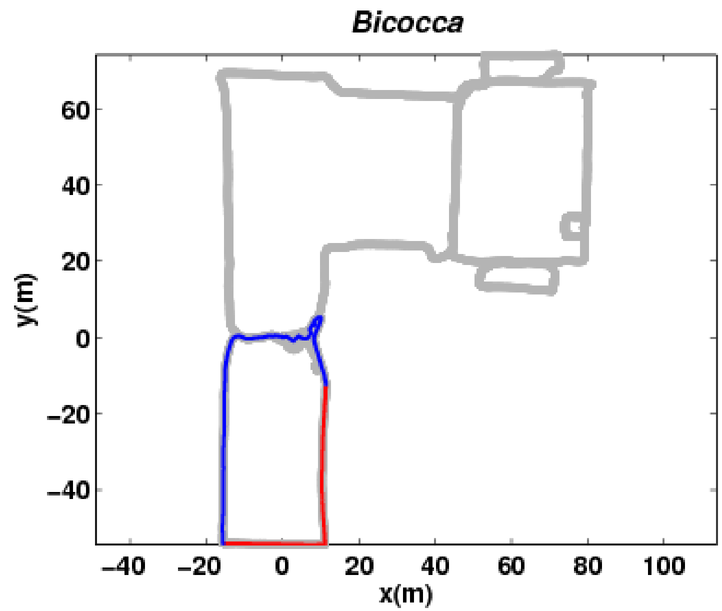
Require:
- A pose graph map of the environment $\mathbf{G}$
- A initial pose n_s and a goal pose n_g.

Ensure:
- The path with the minimum accumulated uncertainty cost from the pose n_s to the pose n_g.

1: $\forall n_i \in \mathbf{G}$: calculate *D-opt*
2: $\forall n_i \in \mathbf{G}$: find reachable neighbours and add edges
3: $\mathbf{G_d} \leftarrow$ ReduceGraph $(\mathbf{G})$
4: $minPath_d \leftarrow$ DijsktraSearch(n_s,n_g,$\mathbf{G_d}$)
5: **return** ReconstructPath($minPath_d$,$\mathbf{G_d}$)

- Search over the decision graph.
- Reconstruct the path over the roadmap.





Experiments

- Objective of the experiments:
  - Comparison of the **minimum uncertainty path** and the **shortest path**.
  - Computational properties of the **minimum uncertainty path**
- Scenarios:
  - g2o with Gauss-Newton solver.

- Simulated environment : **Manhattan dataset**
- Real outdoor environment : **Biccoca dataset**
- Real indoor environment : **Intel dataset**
- Real outdoor environment : **New college dataset**

Experiment I – Comparison (I)

- Experiment: Are the <u>minimum uncertainty path</u> and the <u>shortest path</u> necessarily equal?
  - Select two points A and B, and compare the final accumulated uncertainty.
  - 1000 times x 4 datasets. (Biccoca, Intel , New College and Manhattan).

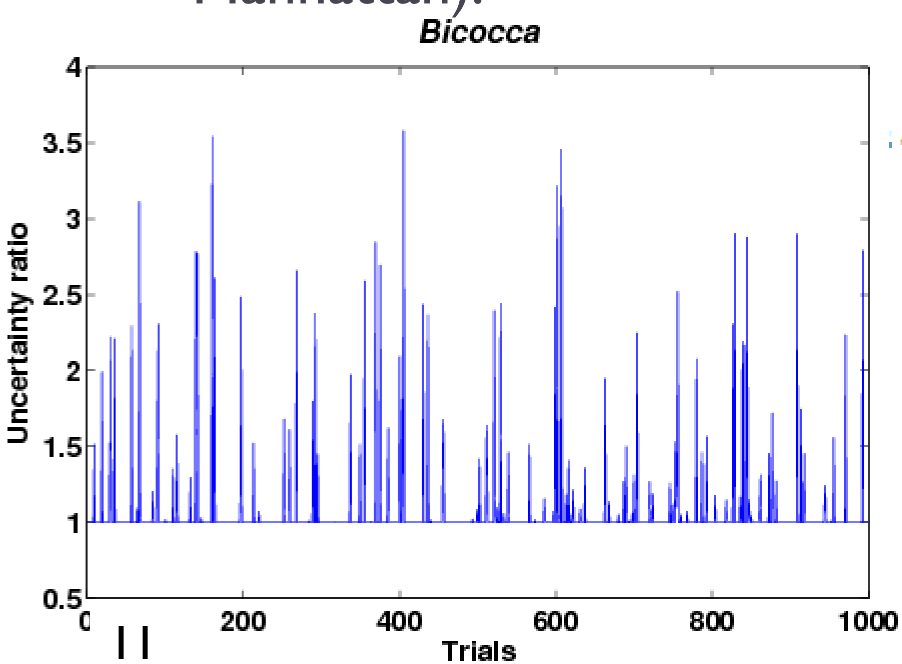


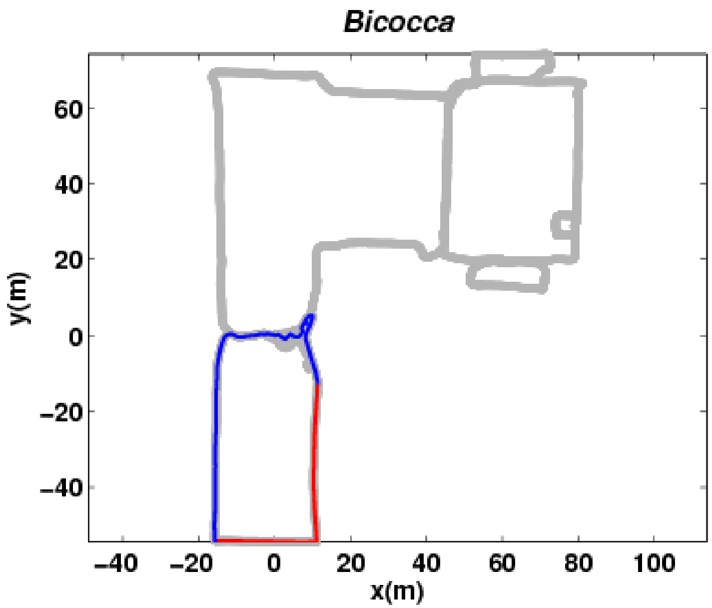
TABLE III
GENERATED BY THE FAMUS ALGORITHM VS THE SHORTEST

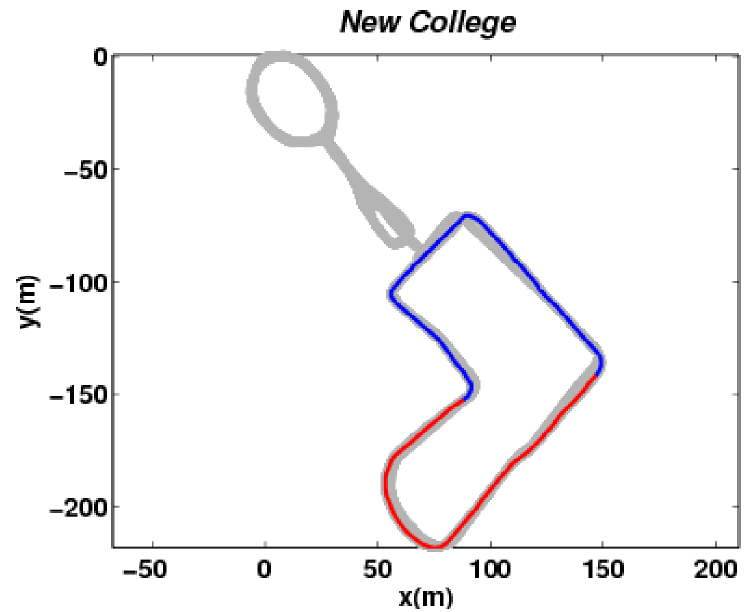
Dataset	= paths	!= paths	% overlap
Bicocca	261	51	87.35%
Intel	170	74	62.59%
Manhattan	146	37	70.22%
New College	215	21	87.79%

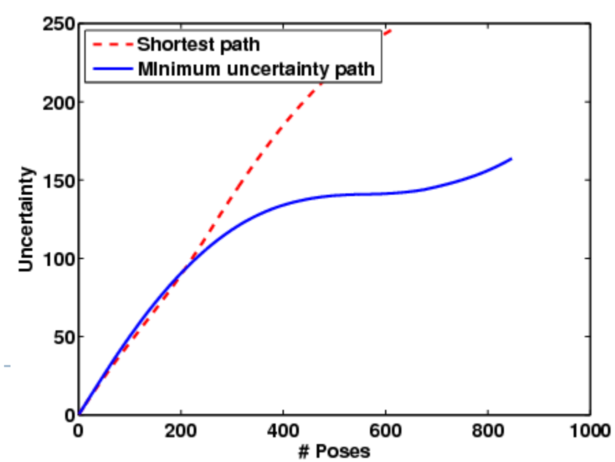
Universidad Zaragoza

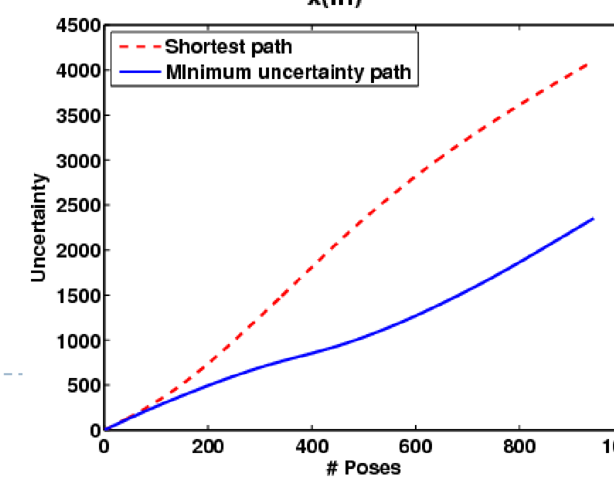
Experiment I – Comparison (II)

- Examples of paths : **ACTIVE SLAM BEHAVIOUR**









Experiment II – Computational (I)

- Reduction in vertices and edges:

TABLE II
PERCENTAGE REDUCTION OF VERTICES AND EDGES BY THE FAMUS ALGORITHM

Dataset (name)	Vertices (Full/Redu.)	Edges (Full/Redu.)	Vertices Reduction	Edges Reduction
Bicocca	8358/980	8513/6936	88.27%	18.52%
Intel	943/623	1837/1527	33.93%	16.87%
Manhattan	3500/2469	5598/4863	29.45%	13.12%
New College	12816/1055	13171/2624	91.76%	80.07%

- Asymptotic time complexity:
  - O [|edges| + |vertices| log(|vertices|)]
  - New College: Vertices 12816 to 1055 (91.76%)
    Edges 13171 to 2624 (80.07%)

Universidad Zaragoza 1542

Experiment II – Computational (II)

- Timing performance
  - Average of 1000 trials en each dataset
  - C++, Intel Core 2 Duo@ 2.8Ghz - 8GB

TABLE IV
TIMING PERFORMANCE OF THE FAMUS ALGORITHM

Dataset	# Vertices	# Edges	Time (ms)
Bicocca	8358	8513	1397.2
Intel	943	1837	215.180
Manhattan	3500	5598	839.97
New College	12816	13171	2143.2

- Improvement of <u>50%</u> in timing with respect to the state-of-the-art. [Valencia et al. 2011]

Take home message

- We proposed FaMUS for obtaining the minimum uncertainty path given a reduced representation of the environment and according to D-opt.

- We validated the algorithm in four dataset and report and improvement of the computation time with respect to the state-of-the-art.

- Further experiments with real robots are needed to generalize the safety of the generate paths under the assumptions.

- Code available in http://www.hcarrillo.com/ and repository at https://github.com/hcarrillo/FaMUS

Thanks!!!

hcarri@unizar.es

http://webdiis.unizar.es/~hcarri

http://www.hcarrillo.com