VISUAL NAVIGATION

Visual SLAM: Simultaneous Localization and Mapping with cameras

Part II

Visual SLAM

Lecture outline:

- Data association (continued)
- Loop closures
- Map management strategies in Visual SLAM
- Other approaches to SLAM
- A look on further challenges in SLAM theory

Data association

- Statistical validation in the observation space: batch gating
- Multiple associations are considered simultaneously
- Widely used methods:

  - Sequential Compatibility Nearest Neighbor (SCNN) algorithm

  - Joint Compatibility Branch and Bound (JCBB), based on a depth-first, continuously pruned tree-search.

  - Combined Constraint Data Association (CCDA), based on graph search (for those interested: *T. Bailey, “Mobile Robot Localisation and Mapping in Extensive Outdoor Environments”, PhD Thesis, University of Sydney, Australian Centre for Field Robotics, 2002*)

Data association

- Statistical validation in the observation space: batch gating with the SCNN algorithm

- First step: adjust state vector based on previous *k-1* associations

Given a first subset e_{k-1} *of paired observations-landmarks*

$$e_{k-1} : \{\mathbf{z}_{1:k-1} \leftrightarrow \mathbf{m}_{1:k-1}\}$$

compute an update conditioned to the current pairings

$$\mathbf{K}_{k-1} = \mathbf{\Sigma}_{k-1}\mathbf{H}_{k-1}^{\mathrm{T}}\left(\mathbf{H}_{k-1}\mathbf{\Sigma}_{k-1}\mathbf{H}_{k-1}^{\mathrm{T}} + \mathbf{Q}_{k-1}\right)^{-1}$$

$$\begin{aligned}\hat{\boldsymbol{\mu}}_{k-1} &= \bar{\boldsymbol{\mu}} + \mathbf{K}_{k-1}(\mathbf{z}_{k-1} - \mathbf{h}_{k-1}(\bar{\boldsymbol{\mu}})) \\ \hat{\mathbf{\Sigma}}_{k-1} &= (\mathbf{I} - \mathbf{K}_{k-1}\mathbf{H}_{k-1})\mathbf{\Sigma}_{k-1}\end{aligned}$$

Data association

- Statistical validation in the observation space: batch gating with the SCNN algorithm

- Second step: evaluate sequential compatibility.

Evaluate $D^2_{ij,k-1}$ *, now based on a partially "updated" pose based on previous k-1 associations*

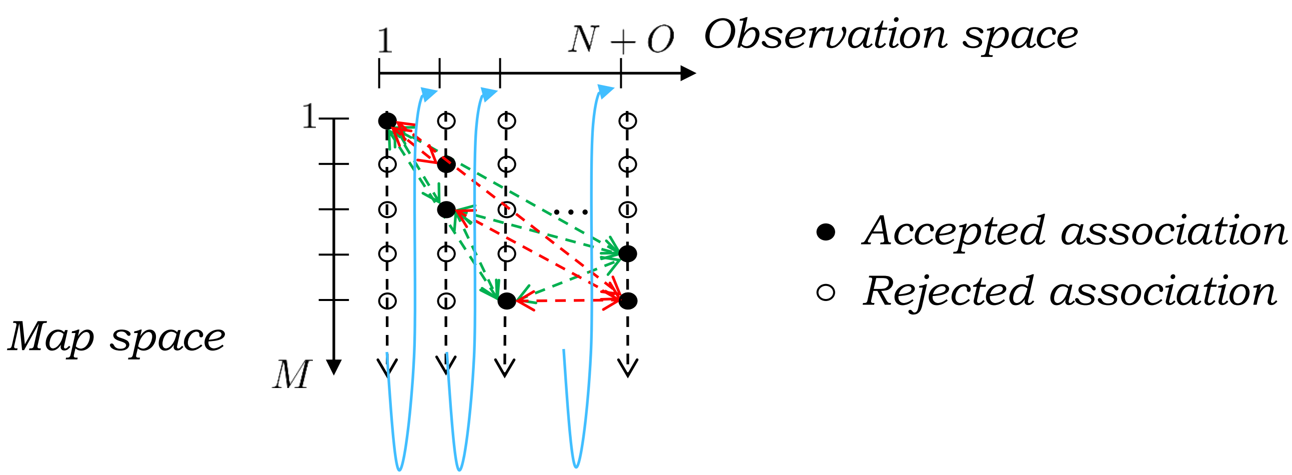
$$D^2_{ij,k-1} = \|\mathbf{z}_i - \mathbf{h}_i(\hat{\mathbf{y}}, \hat{\mathbf{m}}_j)\|^2_{\mathbf{S}_{i,k-1}} \leq k_\alpha$$

with $\mathbf{S}_{i,k-1} = (\mathbf{H}_k \hat{\boldsymbol{\Sigma}}_{k-1} \mathbf{H}_k^{\mathrm{T}} + \mathbf{Q}_k)_i$

An accepted association ij is at this point sequentially compatible with all previous k-1 associations.

Data association

- Statistical validation in the observation space: batch gating with the SCNN algorithm
- Iterate on all observations
- Search logic of SCNN:



Data association

- Statistical validation in the observation space: individual gating

- Advantages:
  - *Sequential conditioning guarantees a-posteriori compatibility*

- Disadvantages:
  - *Sequential conditioning does not guarantees a-priori compatibility*
  - *Update of the whole stochastic map is required at each step k*
  - *The probability of wrong/missed associations is higher in the beginning of SCNN. Such errors are potentially disruptive, as wrong associations lead to wrong decisions in following steps*

Data association

- Statistical validation in the observation space: batch gating with the JCBB algorithm

- JCBB, a modification of the SCNN, enables back-tracking on branches of the three. Two evaluations of each node 'quality' are computed: Joint Compatibility (JC) and Node Quality

- *At level k, we use the previous k-1 associations*

$$e_{k-1} : \{\mathbf{z}_{1:k-1} \leftrightarrow \mathbf{m}_{1:k-1}\}$$

and evaluate the ij association computing the Joint Compatibility, including all previous k-1 pairings :

$$D^2_{ij,1:k-1} = \|\mathbf{z}_{1:k-1,i} - \mathbf{h}_{1:k}(\bar{\mathbf{y}}, \mathbf{m}_{1:k-1,j})\|^2_{\mathbf{S}_k} \leq k_\alpha$$

$$\mathbf{S}_k = \mathbf{H}_{1:k}\bar{\mathbf{\Sigma}}\mathbf{H}^{\mathrm{T}}_{1:k} + \mathbf{Q}_{1:k}$$

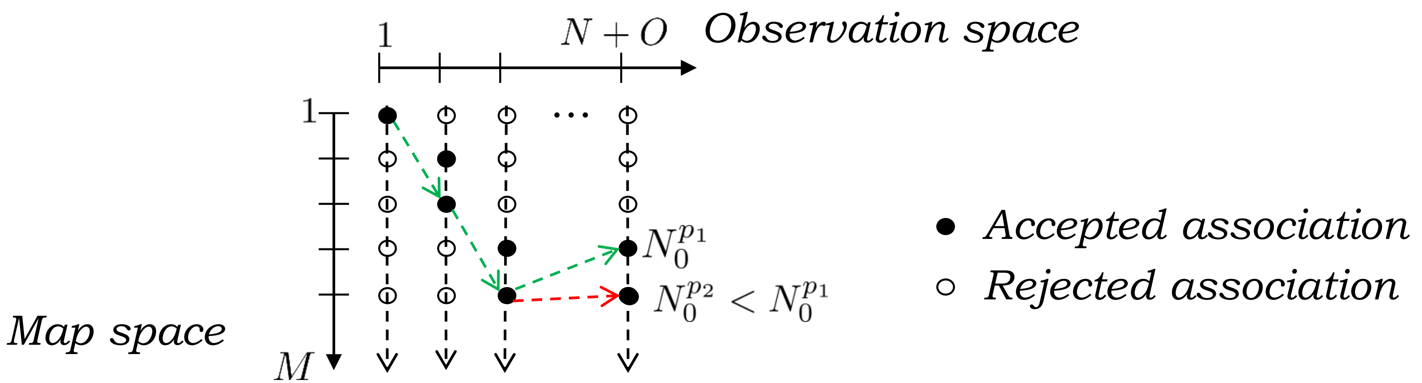
Data association

- Statistical validation in the observation space: batch gating with the JCBB algorithm
- JCBB, a modification of the SCNN, enables back-tracking on branches of the three. Two evaluations of each node 'quality' are computed: Joint Compatibility (JC) and Node Quality
- *Each accepted pairing at level k (node of the search tree) is associated to the number of jointly-compatible associations produced (Node Quality):* N_0^{ij}

Data association

- Statistical validation in the observation space: batch gating with the JCBB algorithm

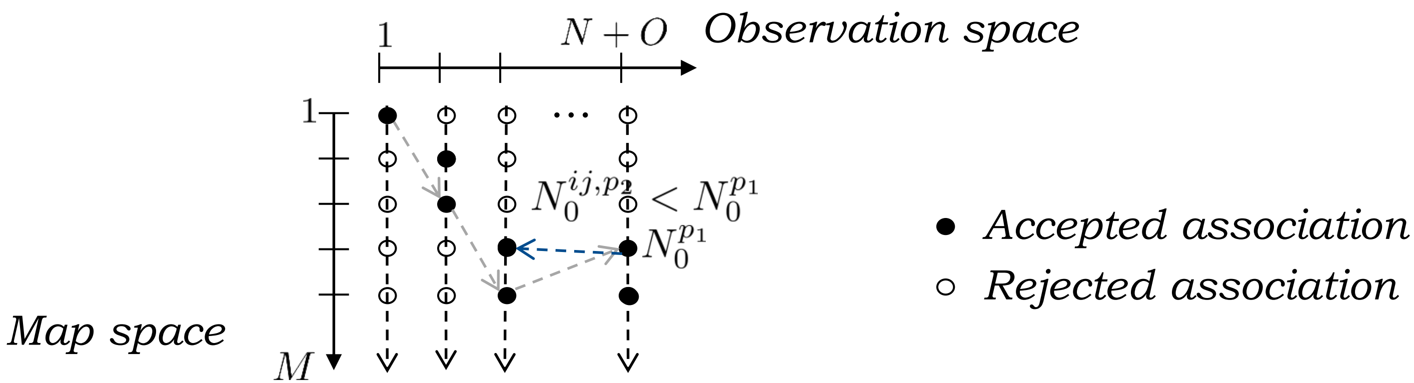
- The search proceeds depth-first, aiming to reach a complete evaluation of a path from the first to the last observations (branching). This path is characterized by $N_0^{p_1}$



Data association

- Statistical validation in the observation space: batch gating with the JCBB algorithm

- Then, the algorithm backtracks, and tries to evaluate a different node. From this node it only goes deep (Bound) if the number of jointly-compatible associations is already equal or larger than the best stored so far ($N_0^{p_1}$)



Data association

- Statistical validation in the observation space: batch gating with the JCBB algorithm

- The evaluation of JC guides the search when selecting the path, and it is the reason for the robustness of the JCBB algorithm.

- The node quality provides the bound heuristic that enables avoiding exploring less likely paths (different possible associations).

- The JCBB algorithm does not guarantee to find the optimal solution to the problem. However, it performs the search in a fast and efficient manner.

Data association

- Statistical validation in the observation space: batch gating with the JCBB algorithm

- *Note that the evaluation of the JC*

$$D^2_{ij,1:k-1} = \|\mathbf{z}_{1:k-1,i} - \mathbf{h}_{1:k}(\bar{\mathbf{y}}, \mathbf{m}_{1:k-1,j})\|^2_{\mathbf{S}_k} \leq k_\alpha$$

$$\mathbf{S}_k = \mathbf{H}_{1:k}\bar{\mathbf{\Sigma}}_{1:k}\mathbf{H}^{\mathrm{T}}_{1:k} + \mathbf{Q}_{1:k}$$

is potentially time-consuming, due to the inversion of a large matrix, $\mathbf{S}_k$, at each node of the search.
This is avoided by computing the inverse $\mathbf{S}_k^{-1}$ iteratively from the previous inverse $\mathbf{S}_k^{-1} = f(\mathbf{S}_{k-1}^{-1}, \mathbf{H}_k, \bar{\mathbf{\Sigma}}_{1:k}, \mathbf{Q}_k)$.
Demonstration that this computation only requires evaluating the inverse of a constant-sized matrix is left for exercise.

Data association

- Appearance signatures

- Cameras provide much richer information than only landmark positions: also the 'signature' of the observed landmark can be stored. E.g., SIFT, FAST or BRISK feature vectors

- *Example of a map entry (SIFT):* $\mathbf{m}_i^* = (\mathbf{m}_X, \mathbf{m}_Y, \mathbf{m}_Z, \mathbf{b}_i^{\mathrm{T}})^{\mathrm{T}}$

  with $\mathbf{b}$ *the feature descriptor*

- *Landmarks can be associated by any metric that measures 'distance' between two landmark appearance signatures*

  Example: $D_{ij}^2 = \|\mathbf{m}_i^* - \mathbf{m}_j^*\|_{\mathbf{\Sigma}_i^*}^2$

  $\mathbf{m}_i^*$ *Observed*
  $\mathbf{m}_j^*$ *From database (map)*
  $\mathbf{\Sigma}_i^*$ *Covariance associated to the observed landmark*

Data association

- Appearance signatures

- Cameras provide much richer information than only landmark positions: also the 'signature' of the observed landmark can be stored. E.g., SIFT, FAST or BRISK feature vectors

- *Example of a map entry (SIFT):* $\mathbf{m}_i^* = (\mathbf{m}_X, \mathbf{m}_Y, \mathbf{m}_Z, \mathbf{b}_i^{\mathrm{T}})^{\mathrm{T}}$

  with $\mathbf{b}$ *the feature descriptor*

- *Any of the previous search-based methods can now be applied (with better expected success) for taking the decision of either accepting or rejecting landmark associations.*

Data association

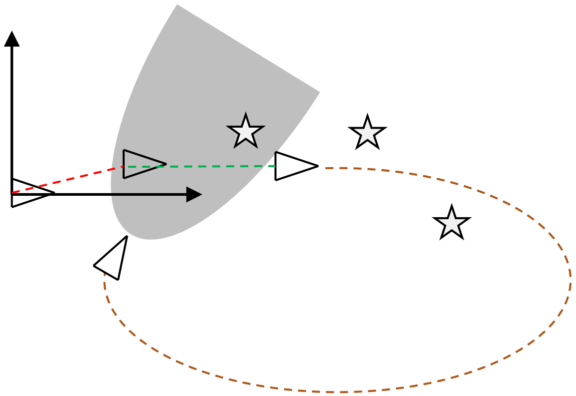
- Multi-hypothesis data association

- Multi-hypothesis data association resolves association ambiguities by generating a separate track estimate for each association hypothesis, creating over time an ever-branching tree of tracks. The number of tracks is typically limited by the available computational resources and low likelihood tracks are pruned from the hypothesis tree.

- Still to be proven feasible in real-time

Loop closures

- Heuristic definition: re-observation of a landmark poorly-correlated with current pose



- Re-observing a poorly-correlated landmark is much more informative than the continuous tracking of a same landmark

- The information about the pose history (trajectory), which is marginalized out during filtering, is retained implicitly though non-null values of correlation between landmarks. Thus, following a loop closure, there is no need to reconstruct the trajectory to correct for landmarks positions: these are 'automatically' re-adjusted due to the correlation in the covariance matrices.

EKF: optimality and computational complexity

- The update step in the EKF equals to one iteration of a nonlinear least squares: this is evidently sub-optimal

$$\begin{aligned} \boldsymbol{\mu}_t &= \bar{\boldsymbol{\mu}}_t + \mathbf{K}_t(\mathbf{z}_t - \mathbf{h}(\bar{\boldsymbol{\mu}}_t)) \\ \boldsymbol{\Sigma}_t &= (\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)\bar{\boldsymbol{\Sigma}}_t \end{aligned}$$

- Iterative EKF (IEKF) algorithms are being studied in order to improve estimation quality (cost: larger computational loads)

- Alternative methods iterate over the whole path and map variables (e.g., bundle adjustment, see later), thus providing an optimal method in the sense of minimizing the overall estimation residual.

EKF: optimality and computational complexity

- Computational complexity of EKF-SLAM is largely dominated by the number of landmarks: from $\propto (M^3)$ for a general approach to $\propto (M^2)$ for methods exploiting the sparseness of the Jacobian matrices.

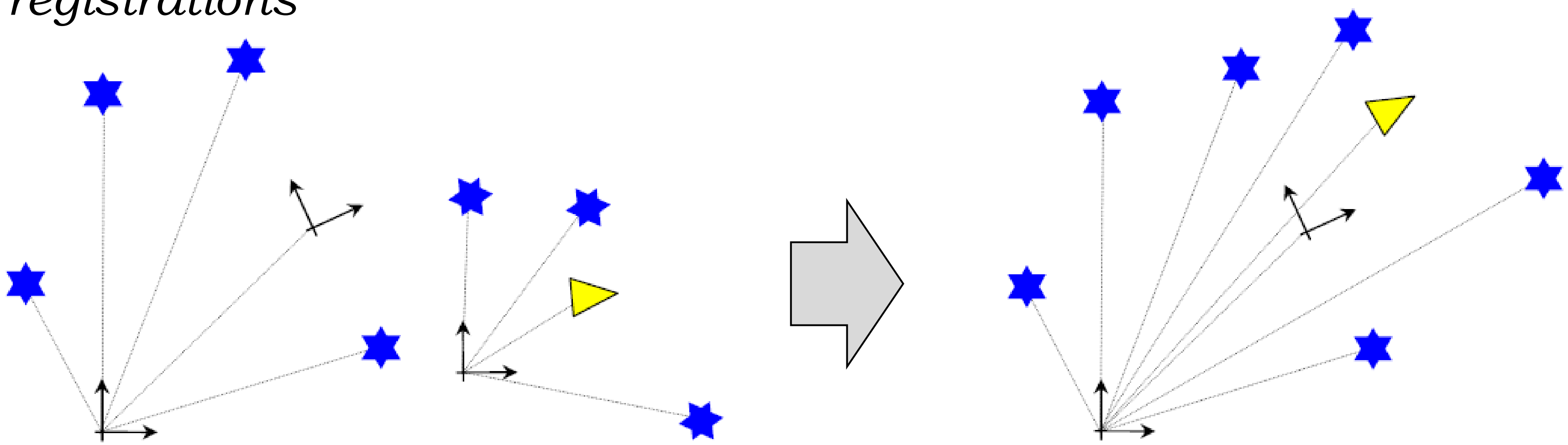
- *Example: the covariance matrix in the prediction step can be partioned as*

$$\bar{\mathbf{\Sigma}}_t = \begin{bmatrix} \mathbf{G}'_{t-1}\mathbf{\Sigma}_{\mathbf{y}_{t-1}}\mathbf{G}'_{t-1}{}^{\mathrm{T}} + \mathbf{R}'_t & \mathbf{G}'_{t-1}\mathbf{\Sigma}_{\mathbf{y}_{t-1},\mathbf{m}_{t-1}} \\ \mathbf{\Sigma}_{\mathbf{m}_{t-1},\mathbf{y}_{t-1}}\mathbf{G}'_{t-1}{}^{\mathrm{T}} & \mathbf{\Sigma}_{\mathbf{m}_{t-1}} \end{bmatrix}$$

thus reducing the complexity from $\propto (M^2)$ *to* $\propto (M)$

EKF: optimality and computational complexity

- *Example : the update step can be executed only on a portion of the map (submaps), with only timely-sparse periodical global registrations*



(a)Global map (b)Local map (c)Global registration

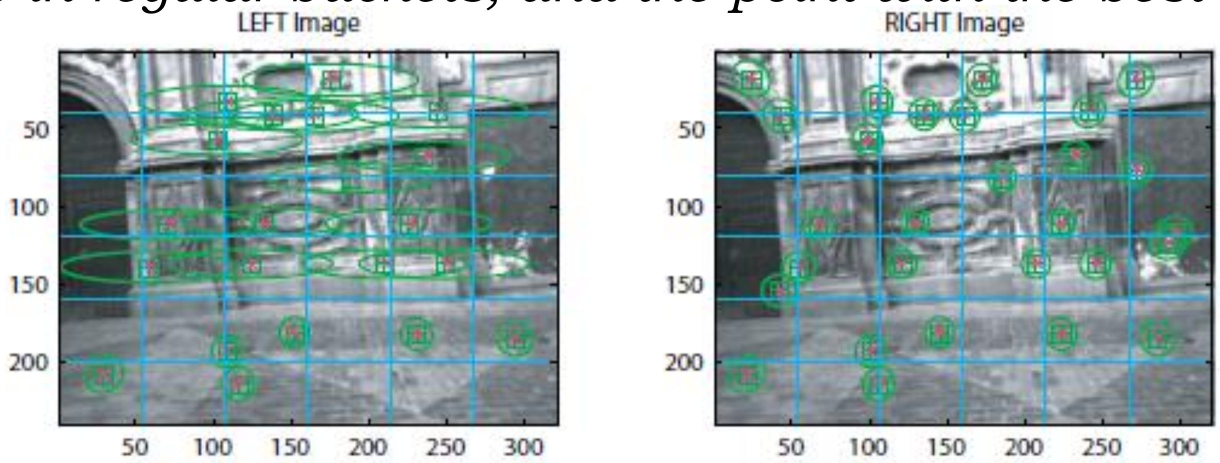
- In general, EKF-based SLAM becomes unmanageable for large maps when all landmarks are stored and updated in the same state vector

Map management strategies in Visual SLAM

- Main objective is scalability: updating and maintaining map should not increases exponentially with the size of the area of interest.

- First and fundamental basic step: limit number of map points by only storing 'strongest' visual features, as sparse as possible.

E.g.: images split up in regular buckets, and the point with the best detector response per cell is selected



From: L.M. Paz, P. Pinies, J.D. Tardos, J. Neira,"6DOF SLAM with Stereo-in-hand", IEEE Transactions on Robotics, 24(5):946-957, 2008

Map management strategies in Visual SLAM

- Two families of approaches: methods that improve on existing filtering techniques (such as EKF-SLAM) by exploiting the 'sparseness' of the observation Jacobian (e.g., *Postponement*, *Compressed EKF*, *Local Map Sequencing*, *Divide and Conquer*) and methods that approach the SLAM problem in a radically different way (e.g., *graph-SLAM, bundle adjustment*)

- *Postponement* and *Compressed EKF* only update a subset of the state vector associated to a submap of points (recently observed and predicted to be observed next). Due to the sparseness of the Jacobian matrices, the prediction and update steps can be re-shaped to only include the subset of variables in the state vector. The full update can be postponed indefinitely (but must be done periodically to update the full map)

Map management strategies in Visual SLAM

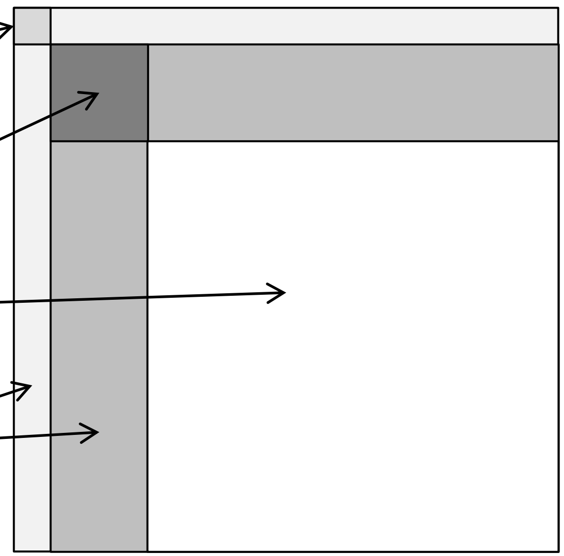
- *Example of submap selection*

Vehicle kinematics/dynamics

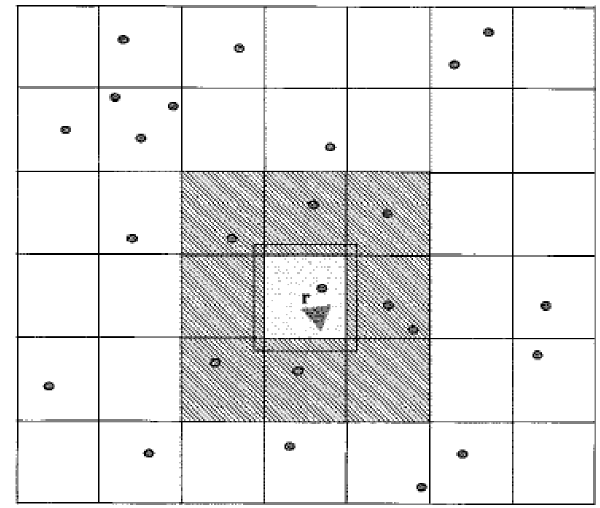
Preserved landmarks

'Discarded' landmarks

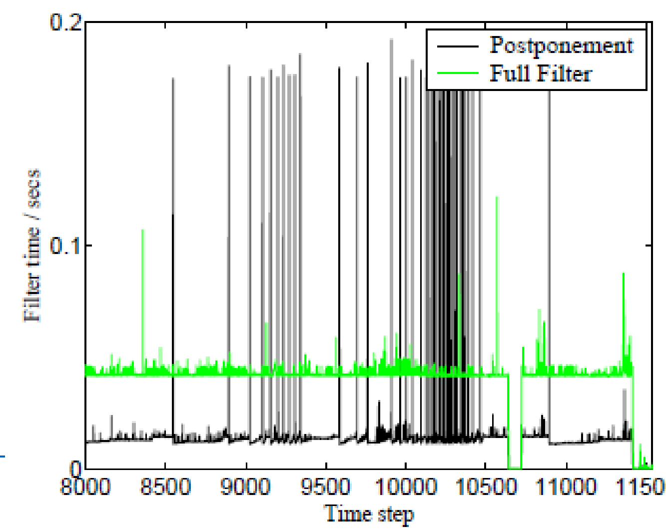
Updated correlations (not discarded!)



Full state covariance matrix



- *Example of computational load for postponement: the time required is smaller than the full EKF, except when an overall update is triggered*



From (Knight Et al., 2001) and (Guivant and Nebot, 2001), see references

Map management strategies in Visual SLAM

- *Local map sequencing*: a map is initialized and managed from a local reference frame for *k* steps. The elements of the local map are statistically independent and uncorrelated with all other submaps.

- During local map building, there is no need to compute the correlations between features in the current local map and features in any other local map. The cost of local map becomes independent from the full size of the map.

- The last known pose in the previous state vector relative to submap *i-1* is used as a first known pose in the current state vector in the local submap *i*.

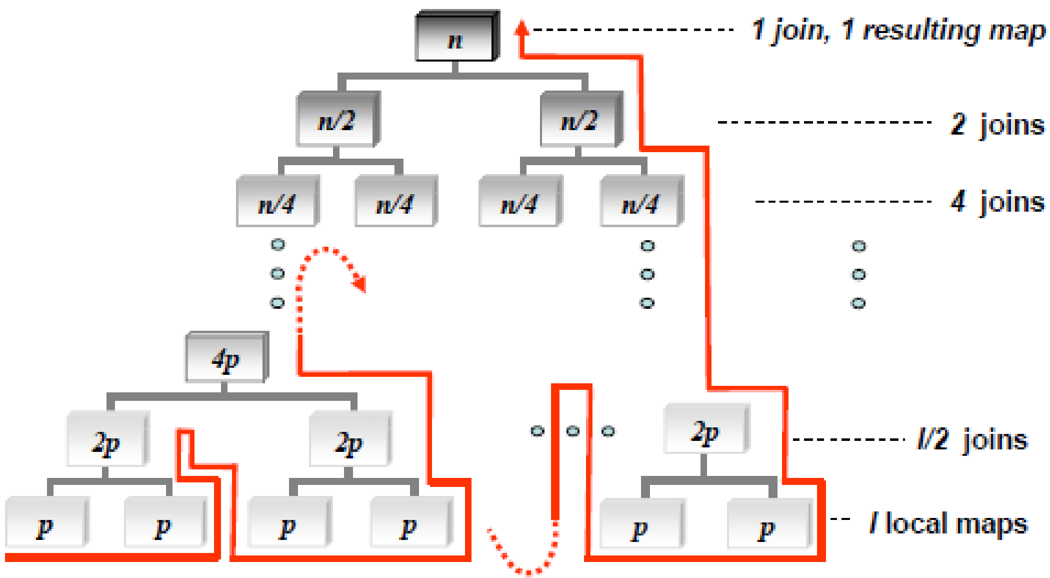
- Submaps are *joined* using common map elements in overlapping consecutive submaps. Correlations are then 'automatically' created

Map management strategies in Visual SLAM

- To eliminate ‘dual’ features after map joining, a *fusion* step is performed, by applying data association techniques in order to identify same landmarks that were not recognized as such after the *joining* step.

Map management strategies in Visual SLAM

- *Divide and Conquer* : improvement on local map sequencing based on the following *join and fusion* tree:



- The complexity associated to the *join and fusion* operations is now proportional* to n^2, *instead of* n^3 as in LMS.

From (*Paz et al., 2008), see references*
(*) n here is the number of landmarks in the map

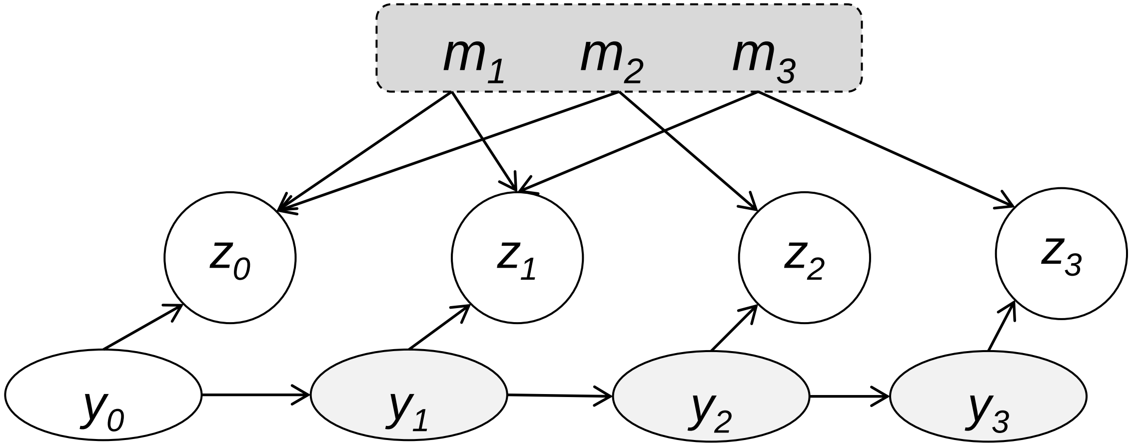
Other approaches to SLAM

- *Graph-SLAM*: the map is NOT part of the state vector, and the full trajectory of the agent is estimated. The correlation between two different poses arises from observations of the same landmark taken from the two poses.

- *Example:*

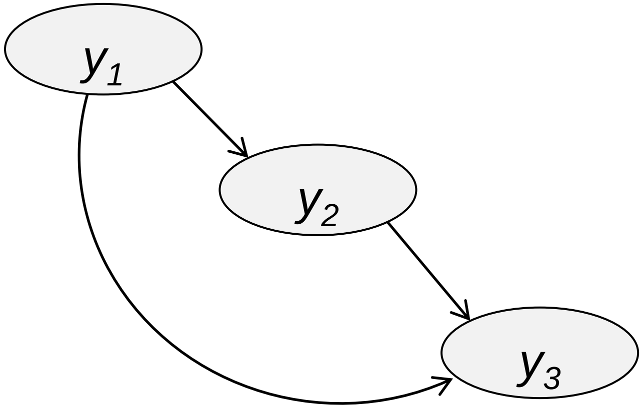
Full SLAM

(Markov representation)



Graph-SLAM formulation

(Graph representation)



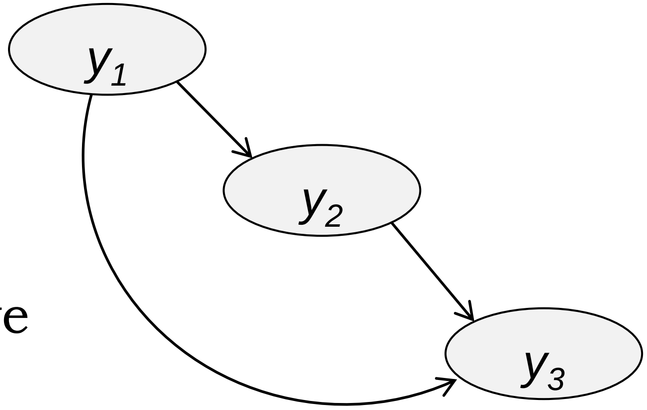
Other approaches to SLAM

- *Graph-SLAM*: the map is NOT part of the state vector, and the full trajectory of the agent is estimated. The correlation between two different poses arises from observations of the same landmark taken from the two poses.

- *Example:*

Graph-SLAM formulation

(Graph representation)



- Nodes: agent's path

- Edges (connections): spatial constraint relating two poses. A constraint consists in a probability distribution over the relative transformations between the two poses.

Other approaches to SLAM

- *Graph-SLAM*: the map is NOT part of the state vector, and the full trajectory of the agent is estimated. The correlation between two different poses arises from observations of the same landmark taken from the two poses.

- A constraint consists in a probability distribution over the relative transformations between the two poses. These transformations are either odometry measurements or motion model propagation between sequential robot positions or are determined by aligning the observations acquired at the two robot locations (1st step: *graph construction)*.

- Once the graph is constructed one seeks to find the configuration of the robot poses that best satisfies the constraints (2nd step: *graph optimization)*.

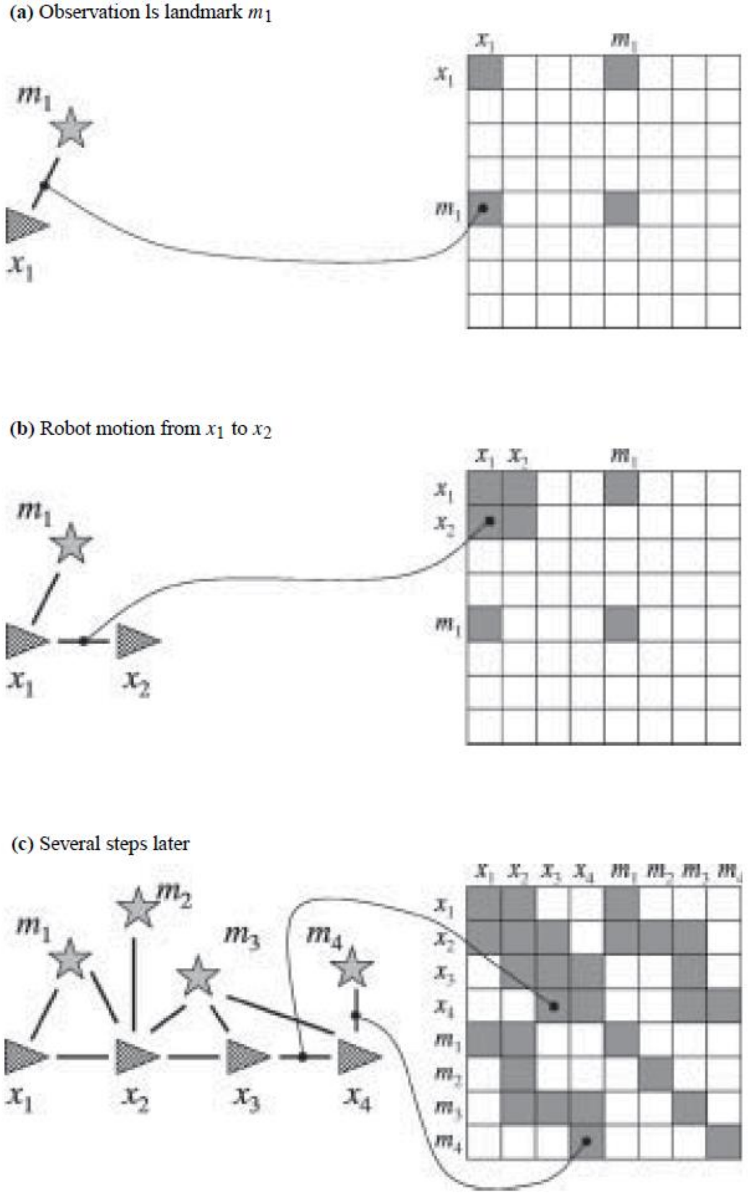
Other approaches to SLAM

- *Graph-SLAM: working principle*

- *Maximization of the a-posteriori probability distribution*

$$p(\mathbf{x}_{1:t}, \mathbf{m} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$$

- *Normally, we would acquire observations and consider directly the "constraints" between landmarks and poses*



From (Thrun and Montemerlo, 2006), see references

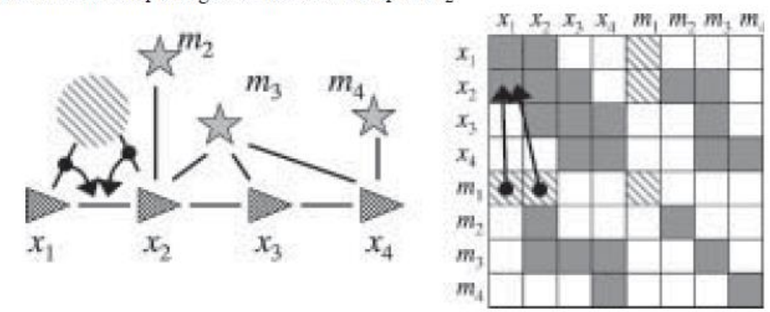
Other approaches to SLAM

- *Graph-SLAM: working principle*

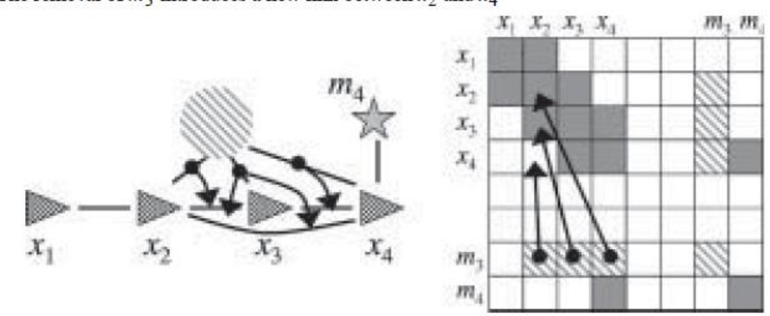
- *Instead, Graph-SLAM applies an important factorization trick: it is possible to remove the landmarks from the (full) state vector by appropriately modifying the correlations between the poses from which that specific landmark was observed*

- *The full pose history is then (non-linearly) estimated , with efficient sparse, nonlinear, least-squares methods.*

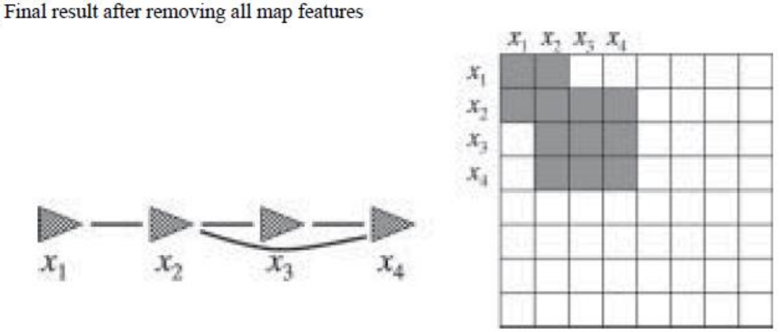
(a) The removal of m_1 changes the link between x_1 and x_2



(b) The removal of m_3 introduces a new link between x_2 and x_4



(c) Final result after removing all map features

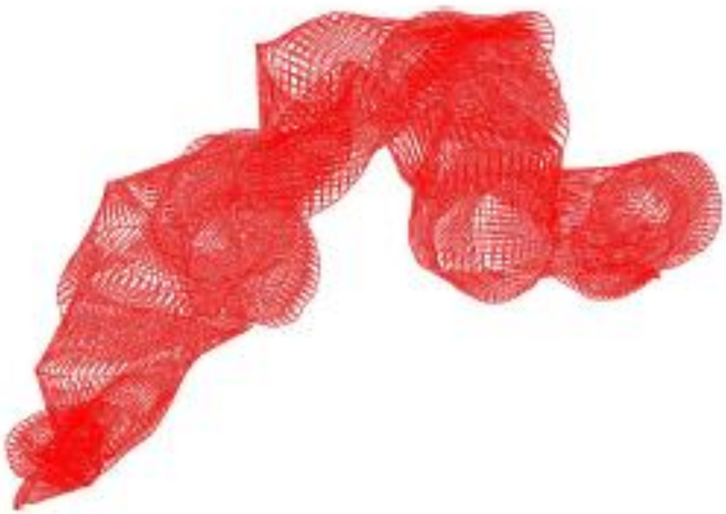


From (Thrun and Montemerlo, 2006), see references

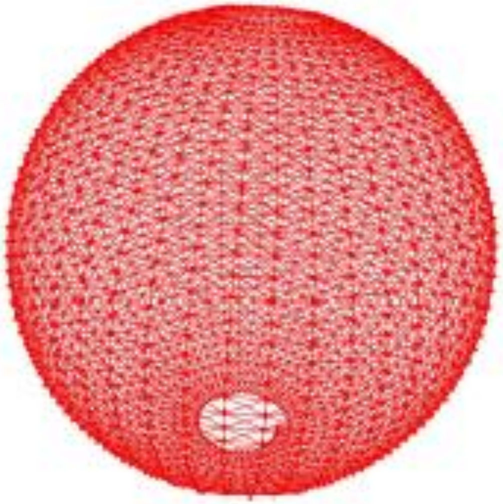
Other approaches to SLAM

➢ *Example of Graph-SLAM: (simulated) robot moving on a sphere*

Graph construction



Graph optimization



Other approaches to SLAM

- *Bundle Adjustment*: the full SLAM problem is resolved with a batch (nonlinear) least-squares estimation. This could be seen as an extension of the Graph-SLAM method approach that explicitly includes landmark positions in the unknown vector.

$$\begin{aligned} \mathbf{x}_t &= \mathbf{g}(\mathbf{x}_{t-1}) + \boldsymbol{\epsilon}_t \quad , \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}\left(\mathbf{0}, \mathbf{R}_t\right) \\ \mathbf{z}_t &= \mathbf{h}(\mathbf{x}_t) + \boldsymbol{\delta}_t \quad , \quad \boldsymbol{\delta}_t \sim \mathcal{N}\left(\mathbf{0}, \mathbf{Q}_t\right) \end{aligned}$$

$$\Downarrow$$

- Combine all motion/odometry information and landmark measurements in a unique (vector) function:

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} \|\mathbf{x}_{t_1} - \mathbf{g}(\mathbf{x}_{t_0})\|^2_{\mathbf{R}_{t_1}} + \|\mathbf{z}_{t_1} - \mathbf{h}(\mathbf{x}_{t_1})\|^2_{\mathbf{Q}_{t_1}} \\ \vdots \\ \|\mathbf{x}_{t_K} - \mathbf{g}(\mathbf{x}_{t_{K-1}})\|^2_{\mathbf{R}_{t_K}} + \|\mathbf{z}_{t_K} - \mathbf{h}(\mathbf{x}_{t_K})\|^2_{\mathbf{Q}_{t_K}} \end{pmatrix}$$

Other approaches to SLAM

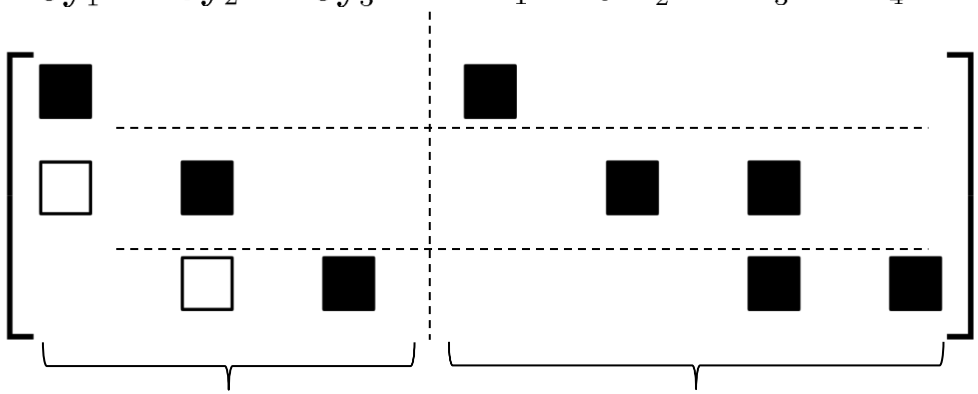
- *Bundle Adjustment*

- Function $\mathbf{f}(\mathbf{x})$ may contain a very large number of nonlinear terms to be evaluated, so it requires the computation of large Jacobian matrices. However, these are sparsely populated!

- *Example:*

$$\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{y}_1} \quad \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{y}_2} \quad \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{y}_3} \quad \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{m}_1} \quad \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{m}_2} \quad \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{m}_3} \quad \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{m}_4}$$

$$\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} =$$



Block diagonal or banded *Sparsely populated*

Other approaches to SLAM

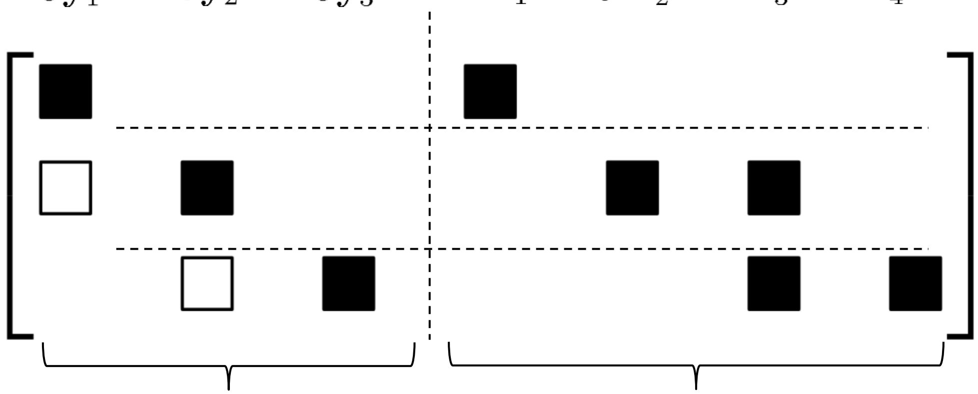
- *Bundle Adjustment*

- The bundle adjustment can then be executed efficiently by means of numerical techniques that exploit the sparseness of the Jacobian matrix

- *Example:*

$$\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{y}_1} \quad \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{y}_2} \quad \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{y}_3} \quad \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{m}_1} \quad \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{m}_2} \quad \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{m}_3} \quad \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{m}_4}$$

$$\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} =$$



Block diagonal or banded — *Sparsely populated*

Other approaches to SLAM

- *Bundle Adjustment in visual navigation*

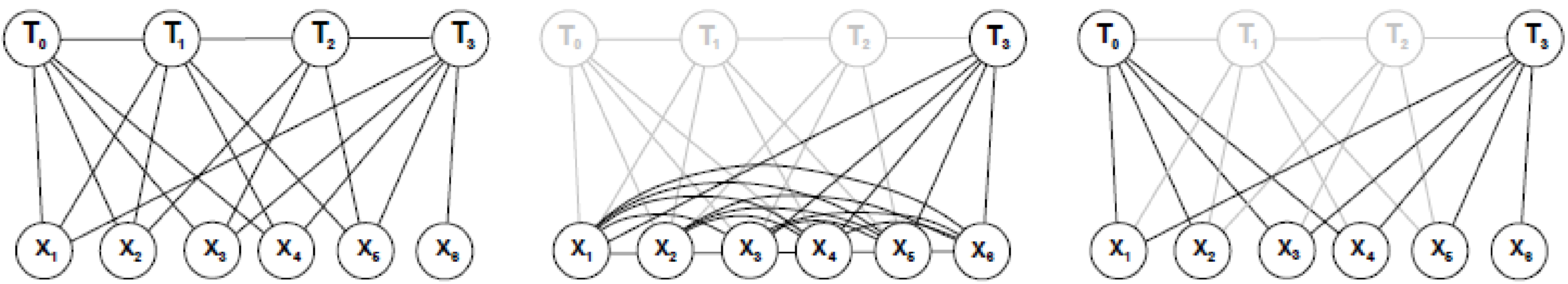
- BA is optimal in the sense that we minimize the overall residuals (reprojection erros). However, despite the existence of very fast numerical approaches to the solution of the nonlinear minimization problem, full BA is infeasible for large maps and paths.

- *Keyframe bundle adjustment* enables potentially real-time BA by discarding (heuristically) multiple frames (portion of the travelled path). This is a trade off between the necessity of including a larger number of landmarks and compensating for the discarded information.

Other approaches to SLAM

- *Bundle Adjustment in visual navigation*

- It is shown in (*Strasdat et Al., 2012,* see references) that Keyframe BA is potentially capable of outperforming EKF in terms of accuracy per unit of computing time



Full SLAM/BA *EKF-SLAM* *(Keyframe) BA*

Further challenges in SLAM theory

- *Stability versus plasticity issues*. The static-world hypothesis (constant map elements) may not hold, and dynamic changes in the environment should be included by means of a certain degree of plasticity in the map (e.g., by implementing algorithms capable of dividing between dynamic elements, that can be updated with time, and static elements).

- *Dimensionality issues*. State-of-the-art SLAM techniques are, at best, complex proportionally to *M* (elements in the map). This prevents building vast maps of the world. Three principles to attack this problem can be isolated:

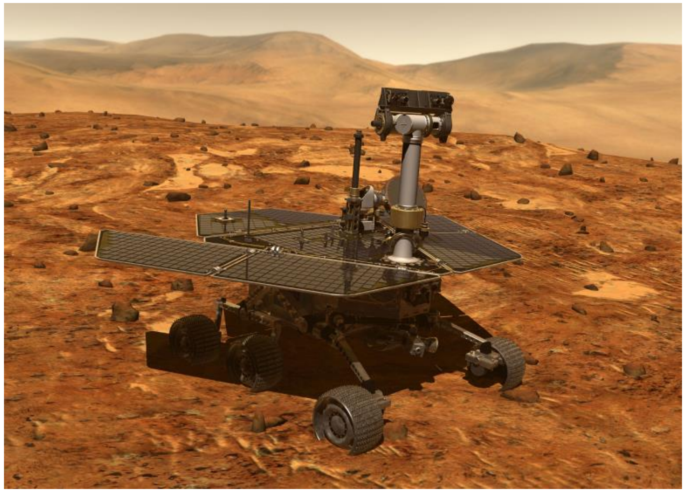
  I) Use of *hierarchical maps*

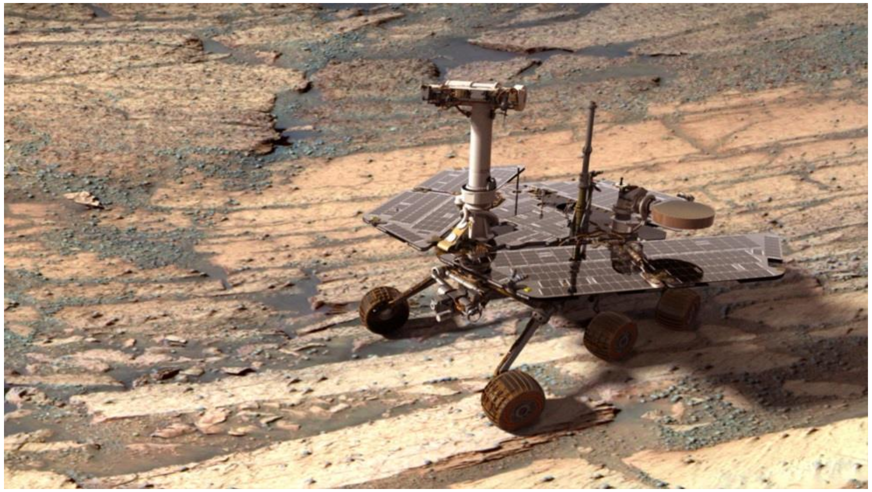
  II) Map maintenance by "*forgetting*" maps elements

  III) Development of *unlimited-scalability* algorithms (*The holy grail!*)

Bonus: Visual Odometry on the Mars Exploration Rovers

- NASA's Mars Exploration Rovers (MER) Spirit and Opportunity successfully landed on Mars in January 2004





- During the first two years of operations, Visual Odometry evolved from an “extra credit" capability into a critical vehicle safety system.

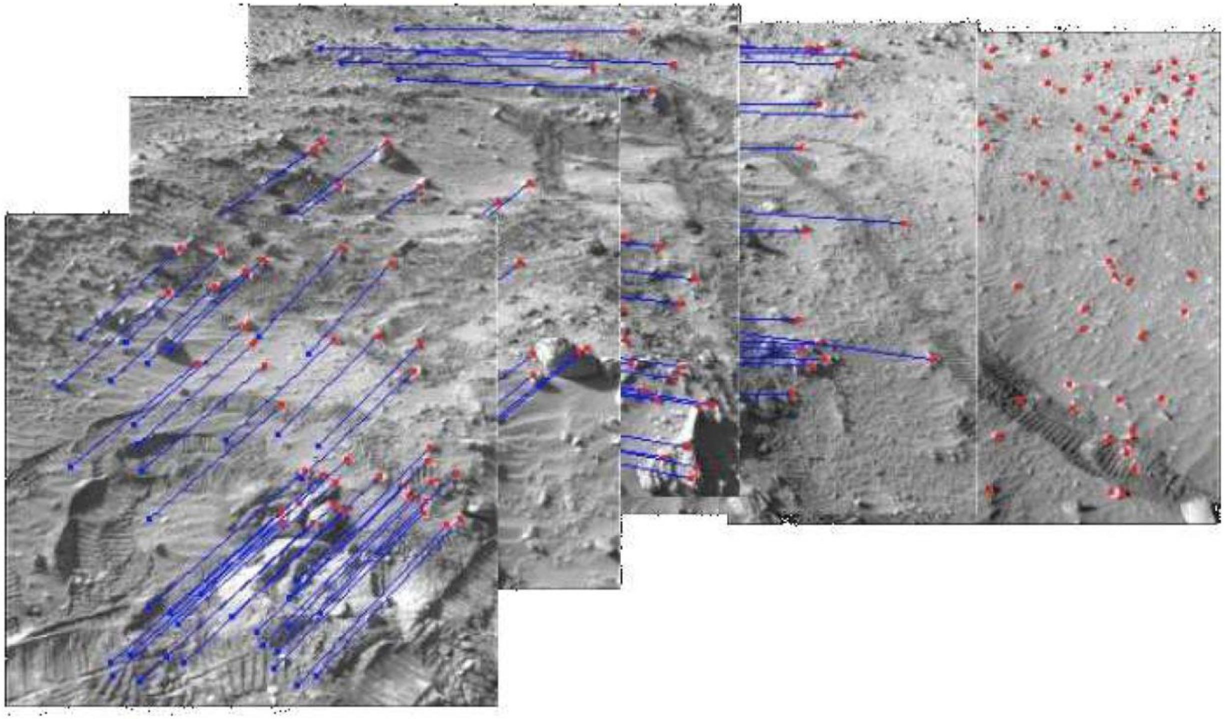
Bonus: Visual Odometry on the Mars Exploration Rovers

- Navigation on extra-terrestrial bodies: no GNSSs, (partially) unknown magnetic fields
- Primary navigation equipment on both MERs: Litton LN-200 Inertial Measurement Unit (IMU, 3-axis accelerometers and 3-axis angular rate sensors) plus wheel odometry
- What about drift and/or slippages (not infrequent on Mars!) ?
- Scientists quickly realized images could be used for a more accurate (visual) odometry solution
- VO more accurate when features were present (thresholds were implemented), but slowed down the rover's exploration speed by an order of magnitude (a mere 20Mhz CPU available onboard)*

(*) *Rovers' CPUs took up to three minutes for a single tracking step using the nominal mission sw*

Bonus: Visual Odometry on the Mars Exploration Rovers

- VO based on continuous tracking of features from a stereo camera



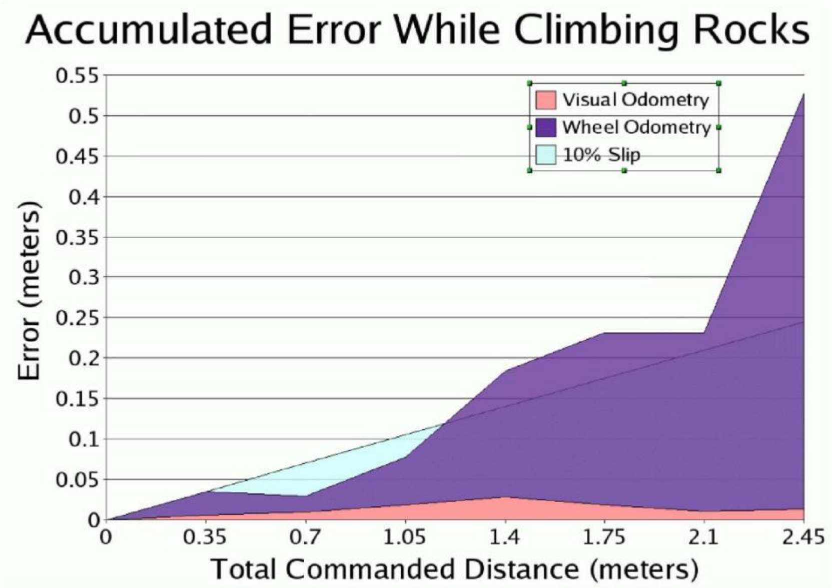
- Feature detectors: Harris' and Forstner's. Images were "bucketed" to limit the number of points and guarantee sparseness

Bonus: Visual Odometry on the Mars Exploration Rovers

- Matching through horizontal epipolar lines (stereo cameras were well calibrated for the task)
- 3D landmark reconstruction performed at each time instance t
- Features tracked across frames, and used to estimate the rover's relative movement between frames (see slides 36-38 of the Visual Odometry lecture)
- RANSAC applied to reduce stereo matching errors (increase estimate robustness)

Bonus: Visual Odometry on the Mars Exploration Rovers

- (Accumulated) errors greatly reduce: here an example of Visual Odometry Error measured during a 2.45 meter drive of a MER Surface System Testbed Lite rover. The rover was driven over several large non-obstacle rocks, each less than 20 cm tall, in 35 cm steps. Severe slips were artificially induced.

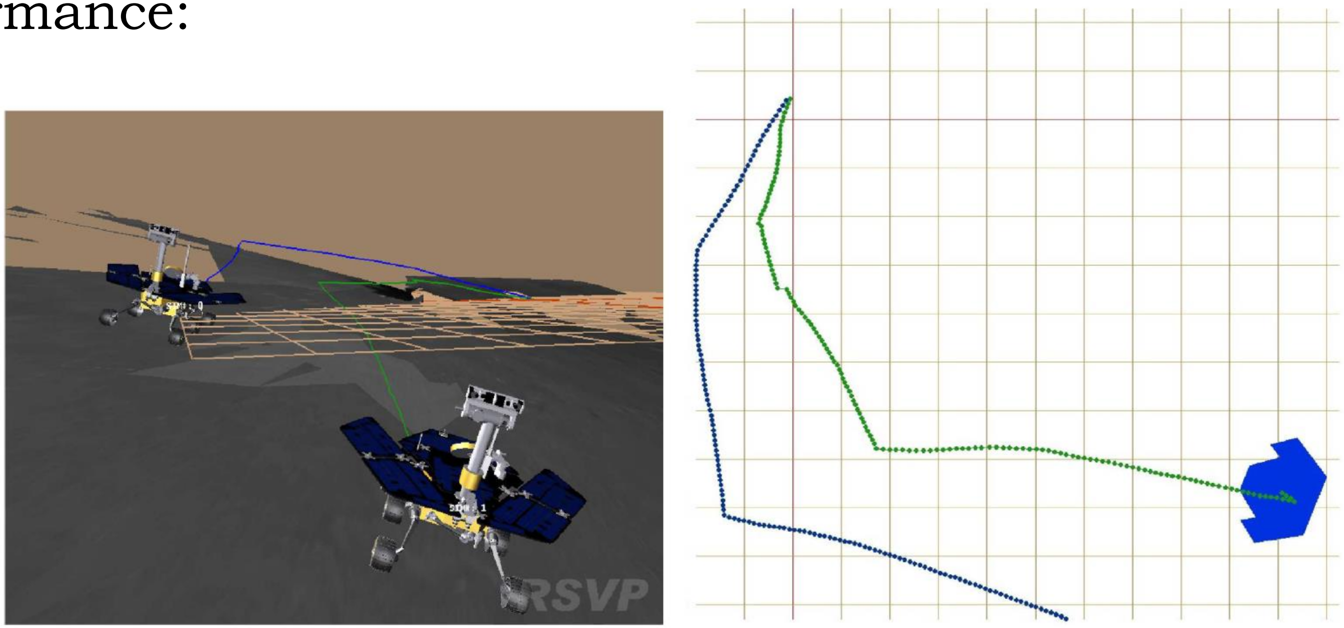


Bonus: Visual Odometry on the Mars Exploration Rovers

- Operational:
  - *45-deg FOV stereo camera, mounted at 1.5 m height*
  - *Image-processing and VO was performed at pre-defined steps (too little computational power available for continuous evaluation)*
  - *Typically, the rover was commanded to drive no more than 75 cm per step, with curves limited to 18 degrees per step*
  - *Due to lack of efficiency, VO was only triggered relatively short drives (typically less than 15 meters) on either steep slopes (typically more than 10 degrees), or in situations where a wheel was being dragged (digging a trench, or conserving drive motor lifetime on Spirit's right front wheel), or driving through piles of sand*

Bonus: Visual Odometry on the Mars Exploration Rovers

- Performance:



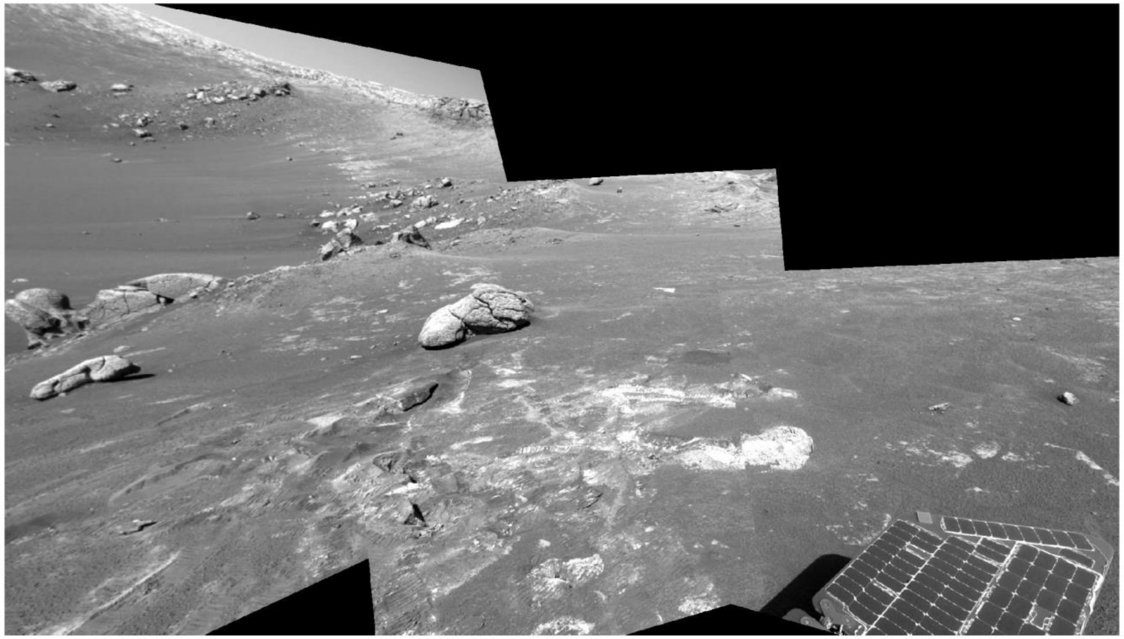
- Views of Opportunity's 19 meter drive from Sol 188 through Sol 191. The green line shows the correct, VO-updated location. The blue line shows how its path would have been estimated from the IMU and wheel encoders alone. Each cell: 1 sqm.

Bonus: Visual Odometry on the Mars Exploration Rovers

- Performance:

← Wopmay, an obstacle in Endurance Crater 60 cm tall, 90 cm wide, and 150cm long

Looking back at → Opportunity's tracks near Wopmay on sol 268 after weeks of trying to drive away. The slope of the sandy terrain varied from 17 to 23 degrees.



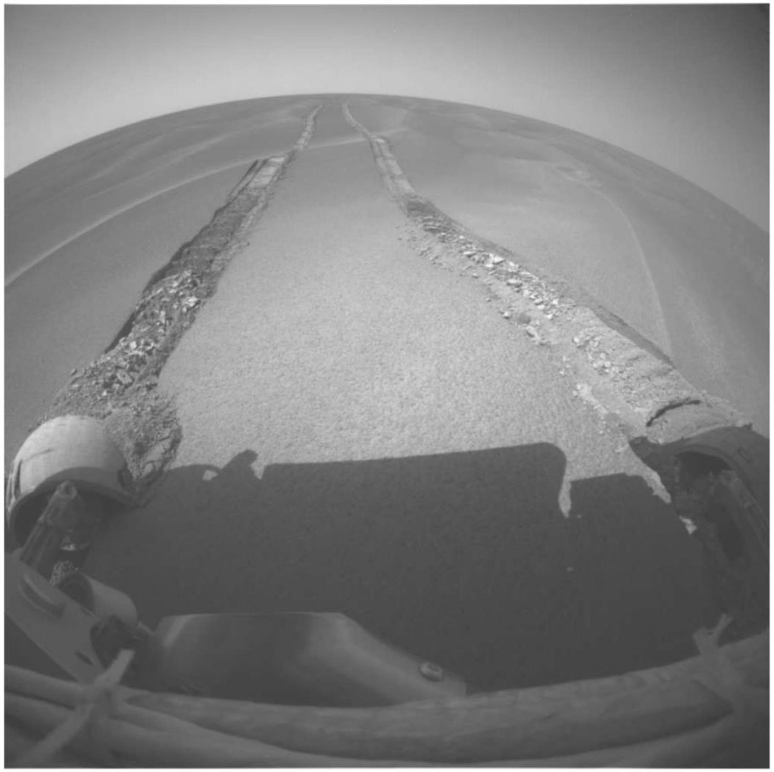
Bonus: Visual Odometry on the Mars Exploration Rovers

- The most extensive use of Visual Odometry was made by Opportunity inside 130 meter diameter Endurance Crater from Sol 133 to Sol 312. Except for a 12 meter approach and return at the lowest point (with lowest rover tilt) on Sols 201 and 203 and a 17 meter drive on Sol 249, Visual Odometry was used virtually continuously throughout. Had it not been available onboard, many more sols would have been needed to approach targets, and fewer targets would have been achieved.

Bonus: Visual Odometry on the Mars Exploration Rovers

- Get out of quicksand!

On sol 446, Opportunity nearly buried its wheels in Purgatory ripple, a nondescript pile of sand, by executing 50 meters of blind driving while failing to climb over it. How to avoid this? After much deliberation, VO was used: after each commanded 2 meters of driving, VO would be used to estimate the vehicle's motion. If VO confirmed that little or no motion had taken place, more commands would be sent'.



Look back at Rovers track!

However, not many features on piles of sand! Solution?

(*) *It took one month to escape Purgatory ripple*

References

General:

- *J.A. Fernandez-Madrigal, J.L. Blanco Claraco, "Simultaneous localization and mapping for mobile robots" IGI Global; 1st Ed., 2012*

- *T. Bailey, H. Durrant-Whyte, "Simultaneous localization and mapping: part I", IEEE Robotics & Automation Magazine, 13(2):99 – 110, 2006*

- *T. Bailey, H. Durrant-Whyte, "Simultaneous localization and mapping: part II", IEEE Robotics & Automation Magazine, 13(3):108 – 117, 2006*

Data association:

- *J. Neira, J.D.Tardos,"Data association in stochastic Mapping Using the Joint Compatibility Test", IEEE Transaction on Robotics and Automation, Vol. 17, n. 6, 2001*

References

Map management:

- *J. Knight, A. Davison , I. Reid, "Towards Constant Time SLAM using Postponement", IEEE/RSJ International Conference on Intelligent Robots and Systems, Maui, Hawaii, 2001*

- *J. E. Guivant , E.M. Nebot, "Optimization of the Simultaneous Localization and Map-Building Algorithm for Real-Time Implementation", IEEE Transactions on Robotics and Automation, 17(3):242–257, 2001*

- *J. Tardos, J. Neira, P. Newman, J. Leonard, "Robust mapping and localization in indoor environments using sonar data", International Journal of Robotics Research, 21(4):311–330, 2002.*

- *L.M. Paz, J.D. Tardos, J. Neira, "Divide and Conquer: EKF SLAM in O(n)", IEEE Transactions on Robotics, 24(5):1107,1120, 2008*

References

Other approaches to SLAM:

- *G. Grisetti, R. Kümmerle, C. Stachniss, W. Burgard, "A Tutorial on Graph-Based SLAM", IEEE Intelligent Transportation Systems Magazine, 2(4):31 – 43, 2011*

- *S. Thrun, M. Montemerlo, "The Graph-SLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures" International Journal of Robotics Research, 25(5):403–430, 2006.*

- *H. Strasdat, J.M.M. Montiel, A.J. Davison, "Visual SLAM: Why filter?", Image and Vision Computing, 30:65-77, 2012*

Visual odometry on Mars:

- *M. Maimone, Y. Cheng, L. Matthies, "Two Years of Visual Odometry on the Mars Exploration Rovers", Journal of Field Robotics, Special Issue on Space Robotics, Part I, 24(3):169–186, 2007*