A Hybrid direct feature matching based on-line loop-closure detection algorithm for topological mapping

Swagat Kumar*, Nishant Kejriwal* and Tomohiro Shibata†

TATA

TATA CONSULTANCY SERVICES

*Innovation Lab, Tata Consultancy Services, New Delhi, India

Kyutech

Kyushu Institute of Technology

†Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology, Japan

ICRA 2014 Workshop on Active Visual Learning and Hierarchical Visual Representation for General-Purpose Robot Vision

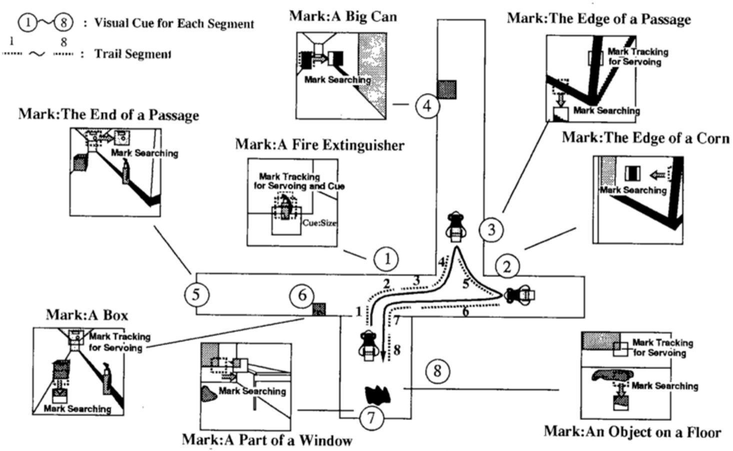
May 31, 2014

Overview

- Motivation
- Existing Solutions for Appearance-based Schemes
- The Proposed Method: HDFM
  - Loop Closure Detection
  - Node Update
  - Map Generation through Bayesian Learning
- Experimental Results
  - Performance Analysis
- Future Work
- Conclusion

Motivation

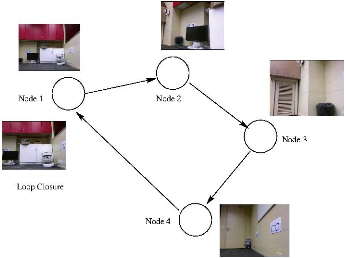
- Topological mapping is suitable for human-robot interaction



Shibata, Matsumoto, et al., IEEE Trans Mechatronics, 1996

Motivation

- Topological mapping aims at building a graphical model of the environment comprising of *key locations* and their connectivity.
  - Images are used to identify places.
  - Geometrical information is not used.



- Challenges
  - Loop-closure detection: Robot has to recognize previously visited places.
  - Perceptual Aliasing: Physically distinct locations may appear similar to robot sensors.
  - Growing computational complexity with increasing map size.

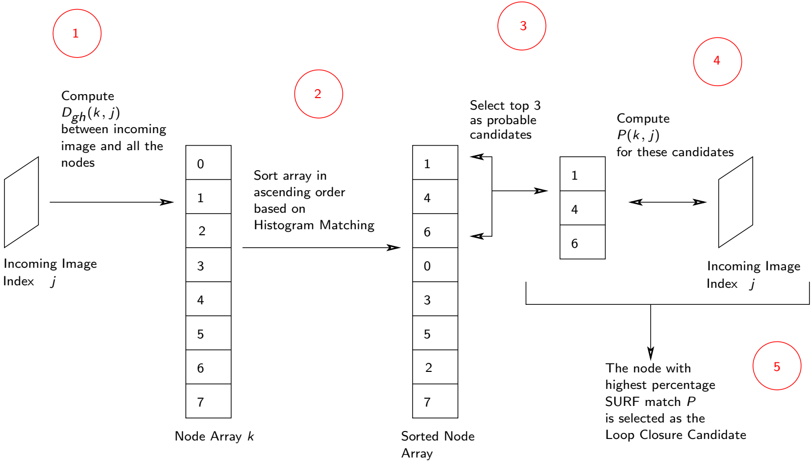
Existing Solutions: Appearance based schemes

- Pixel-wise matching, local/global histogram matching and Bag-of-words.
  - Histogram matching is simpler and faster - Perceptual aliasing due to quantization error.
- Bag-of-Words method is most commonly used method for topological mapping: Filiat (2007), Angeli (2008), Cummins & Newmann (2008), Lopez & Tardos (2012), Nicosevici & Garcia (2012).
- Direct feature matching is superior to other methods for finding similarity between images: Zhang (2010)
  - Bag of Raw features (BoRF): Zhang (2007)
  - Use of KD-tree for loop closure detection: Liu (2012)
  - Use of LSH to deal with computational complexity: Shahbazi (2013)
  - Growing computational complexity with map size.

The proposed Approach

- Combining simplicity of histogram matching and accuracy of direct feature matching: Hybrid Direct Feature Matching (HDFM)
- New locations identified as images are stored in a binary search tree.
- For every incoming image, three loop closure candidates are selected using histogram matching (Hellinger distance): $\sim \log n$, n is the map size.
- The best loop closing candidate is selected using direct feature matching: fixed time independent of map size.
- The node in the graph is updated on loop closure detection.
- The effect of wrong decisions are reduced over time through a naive Bayesian filter.
  - The map of an environment is comprises of most frequently travelled locations.

Proposed method for loop closure detection

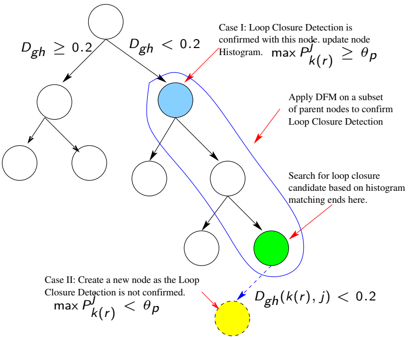


Loop Closure Detection

Two Step method:

- Select a set of possible loop closure candidates (nodes) using global histogram matching.
- Best loop closure candidate is selected using direct feature matching.
- Percentage match for an incoming image I_j with a given node k is defined as

$$P_k^j = \frac{|\mathscr{V}_k \cap V(I_j)|}{|V(I_j)|} \times 100 \tag{1}$$



Hellinger Distance: $D_{gh} = \sqrt{1 - \sum_x \sqrt{p(x)q(x)}}$

where $p(x)$ and $q(x)$ are the source and target histograms

Node Update

- A node k is represented by a histogram H_k and a pool $\mathscr{V}_k$ of SURF descriptors.
- Whenever a loop closure is detected, the node update include two steps
  - Adding descriptors of incoming image into the pool

$$\mathscr{V}_k = \mathscr{V}_k \cup V(I_j) \tag{2}$$

    A fixed size descriptor pool could be maintain to limit the growing memory requirement.
  - Histogram of the node is updated by re-distributing the descriptors of the new image among the histogram bins.

Experimental Results

Detecting Loop Closures:

Case I:	Query Image $j = 436$	Node (k)	Images belonging to the node
		1	
		5	
		12	
Case II:	**Query Image $j = 942$**	**Node (k)**	**Images belonging to the node**
		18	
		24	
		27	

Figure : Loop Closure Detection under two cases.

Loop Closure Detection

Table : Confirming Loop Closure Detection using Direct feature matching

Case I: $j = 436$				Case II: $j = 941$		
r	Node (k)	D_{gh}	$P^j_{k(r)}$	Node (k)	D_{gh}	$P^j_{k(r)}$
1	1	0.1615	06%	18	0.1349	01%
2	5	0.1586	30%	24	0.2028	04%
3	12	0.1848	05%	27	0.1862	16%

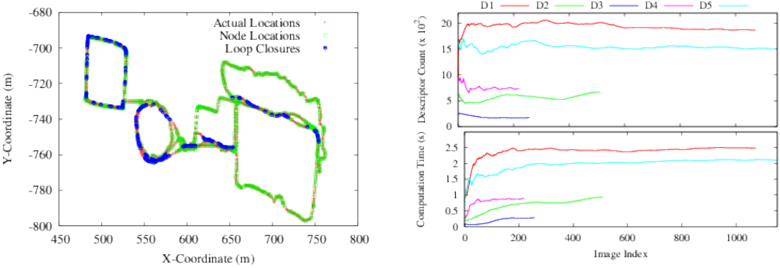
- Two-step process involving histogram matching and direct-feature matching is crucial in correctly identifying loop closures.
- Case I: For query image $j = 436$, loop closing node is $k = 5$ for which D_{gh} is minimum and $P^j_{k(r)}$ is maximum.
- Case II: For query image $j = 942$, loop closing node is 27 for which D_{gh} is NOT minimum but $P^j_{k(r)}$ is maximum.

Performance on different datasets

S. No	Data Set	No. of Images, Resolution, Avg. no. of descriptors/image	No. of nodes	No. of loop closures	Recall (%)	Avg. comp. time per image (sec)	Total time (hours)
D1	New College dataset (Cummins)	2146, 640×480, 1800	1573	220	24	2.4	1.48
D2	Hallway dataset #6 (Zhang)	512, 640×480, 500	128	0		0.7	0.13
D3	Hallway dataset #2 (Zhang)	259, 320×240, 200	32	76	70	0.23	0.017
D4	Our own Indoor Lab data	222, 640×480, 800	40	150	87	0.8	0.054
D5	City Centre dataset (Cummins)	2474, 640×480, 1500	1866	244	22	2.1	1.44

- If the dictionary is created off-line the computation time per image is comparable to existing methods.
- Recall rate is comparatively lower.

Computation time for different algorithms



- Computation time has a sub-linear growth w.r.t map size.
- Computation time per image is directly proportional to the number of SURF descriptors available in the image.
- If the dictionary is created off-line, the computation time per image is comparable to existing methods.
- Recall is comparatively lower.

Computational Complexity of the Algorithm

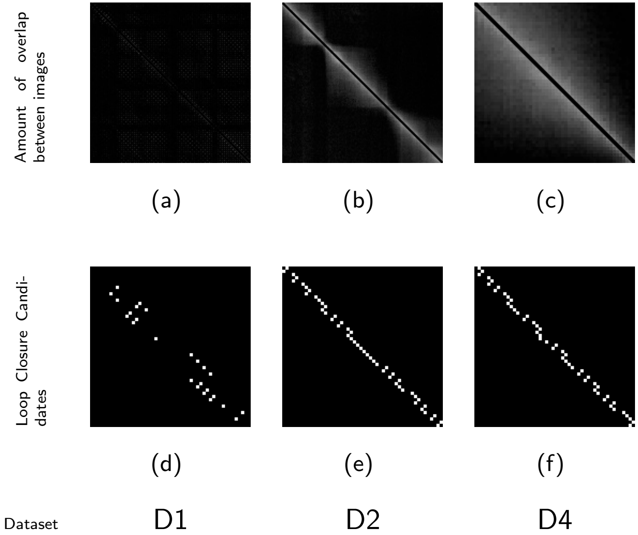
Total computation time for each image: $T = T_e + T_1 + T_2$

- T_e: Time needed for extracting SURF descriptors from each image.

$$
\begin{aligned}
T_e &\propto N_d \quad \left(\text{\scriptsize Number of descriptors in an image}\right) \\
T_e &= C \quad \left(\text{\scriptsize constant w.r.t map size}\right)
\end{aligned}
$$

- T_1: Time needed for finding loop closure candidates through histogram matching. $T_1 \sim O(\log M)$, M being the map size
- T_2: Time needed for confirming loop closure using direct feature matching. $T_2 = C$ and does not increase with map size.
- Overall, the computation time increases logarithmically with map size.

Dealing with Similarity between consecutive images



Conclusion

- A new method for loop closure detection is proposed by combining histogram matching with direct feature matching.
- It does not require any off-line dictionary.
- The computation time per image is comparable to other direct feature matching methods.
- However, our recall rates are lower.
- Future work: focus on increasing recall rates and reducing computation time.

Thank You

Contact Emails:
swagat.kumar@tcs.com
tom@brain.kyutech.ac.jp