Sparser Relative Bundle Adjustment: constant-time maintenance and local optimization of arbitrarily large maps

José-Luis Blanco-Claraco
Javier González-Jiménez
Juan-Antonio Fernández-Madrigal

May 7, 2013
Karlsruhe – ICRA 2013
Long-term goal: truly autonomous robots

Most roboticians agree a robot must autonomously learn **how the world looks like** and **where it is**: **SLAM**
Ideal SLAM pipeline… for a life time?

1/4: Introduction
Ideal SLAM pipeline… for a life time?

1/4: Introduction
Ideal SLAM pipeline… for a life time?

SLAM front-end

Data association

O(1)

Sensory data

SLAM back-end

World model

Where am I?

The map
That’s the question

Rational decision towards $O(1)$?

**To use** global coordinates

**Not to use** global coordinates
SLAM in relative coordinates
SLAM in relative coordinates

Unknowns:
- Keyframe-to-keyframe poses
- Landmark relative positions

Known data:
- Observations
SLAM in relative coordinates

Key for O(1): *Locally* consistent maps

Introduced by Gabe Sibley and colleagues:

Loop closing in *relative* SLAM

vs.

2/4: Loop closures in RBA
Edge creation and the locally-consistent area

- Fewer unknowns
- Consistent maps are larger

Tradeoff?

2/4: Loop closures in RBA
A totally new problem: edge creation

The problem of edge-creation:
Given a set of observations, how many and which edges should be created?
A totally new problem: edge creation

**The problem of edge-creation:**
Given a set of observations, how many and which edges should be created?

**Optimal solution?** We still don’t know
The power of the edge creation policy

Different policies:

• The “intuitive” linear graph policy: (→ RBA [G.Sibley et al.])

![Diagram showing a linear sequence of nodes labeled 0, 1, 2, 3, 4, ..., 14, 15, 16.]

2/4: Loop closures in RBA
The power of the edge creation policy

Different policies:

- The “intuitive” linear graph policy: (→ RBA [G. Sibley et al.])

- All edges to the same keyframe: (→ becomes *global* SLAM)
The power of the edge creation policy

Different policies:

• The “intuitive” linear graph policy: (→ RBA [G.Sibley et al.])

• All edges to the same keyframe: (→ becomes global SLAM)

• Something in between?

2/4: Loop closures in RBA
Our proposed policy

Inspired by hierarchical submapping methods:
Our proposed policy

Inspired by hierarchical **submapping** methods:

Probably, the first framework that **seamless integrate global and relative** coordinates.
The need for spanning trees in RBA

Observation model of landmark include **the path** from: observer KF → base KF

▶ 3/4: Incrementally building spanning trees
The need for spanning trees in RBA

Observation model of landmark include the path from: observer KF $\rightarrow$ base KF

→ Keep STs up to a maximum topological depth $D_{\text{max}}$

3/4: Incrementally building spanning trees
An algorithm for updating STs: basic idea

3/4: Incrementally building spanning trees
An algorithm for updating STs: basic idea

For all $r$ and $s$:
Is the new path shorter?
Is a new path in range of $D_{\text{max}}$?

Update the STs accordingly:
- $r \rightarrow s$: Go towards “$n$”
- $s \rightarrow r$: Go towards “$i_k$”
An algorithm for updating STs: complexity

Computational complexity:

\[ O\left( N_R^2 \log N_R \right) \]
An algorithm for updating STs: complexity

Computational complexity:

\[ O\left(N_R^2 \log N_R \right) \]

In practice: \( O(1) \)

3/4: Incrementally building spanning trees
Experiments: (1) Monocular SLAM
Experiments: (1) Monocular SLAM
Experiments: (1) Overall processing time

**Total**: 55,000 keyframes, 4,000,000 observations, 400,000 landmarks

[Graph showing time/iteration vs. key-frame (x10^3)]
Experiments: (2) 2D graph-SLAM demo
Experiments: (2) 2D graph-SLAM demo

Current Kf=1
After opt RMSE=0.00000 | Total step time=0.357ms

GROUND TRUTH

4/4: Results
Conclusions
Conclusions

• Contributions:

  ➢ Proposal of **edge-creation policies** as worthy of research.

  ➢ **Blended global-relative** coordinates, similar to submapping.

  ➢ **O(1)** algorithm for **online updating** of spanning trees.
Open source release

Public C++ implementation.
Policy-based design → flexibility
Open source release

Public C++ implementation.
Policy-based design $\rightarrow$ flexibility

```cpp
typedef RbaEngine<
    kf2kf_poses::SE3, // Parameterization KF-to-KF poses
    landmarks::Euclidean3D, // Parameterization of landmark positions
    observations::RangeBearing_3D // Type of observations
>
my_rba_t;
```

SLAM and BA-like problems

Relative Graph SLAM

4/4: Results
Open source release

Available in Ubuntu 13.04 official repository (other distros → can use PPA)

```bash
$ sudo apt-get install libmrpt-dev mrpt-apps
$ srba-slam --help
```

More online:

http://www.mrpt.org/srba
Sparser Relative Bundle Adjustment: constant-time maintenance and local optimization of arbitrarily large maps

Thanks for your attention!

More info online: http://www.mrpt.org/srba