

Real-Time Spherical Mosaicing using whole image alignment



team5

Andreas Bigontina

Oleksandr Golovatenko

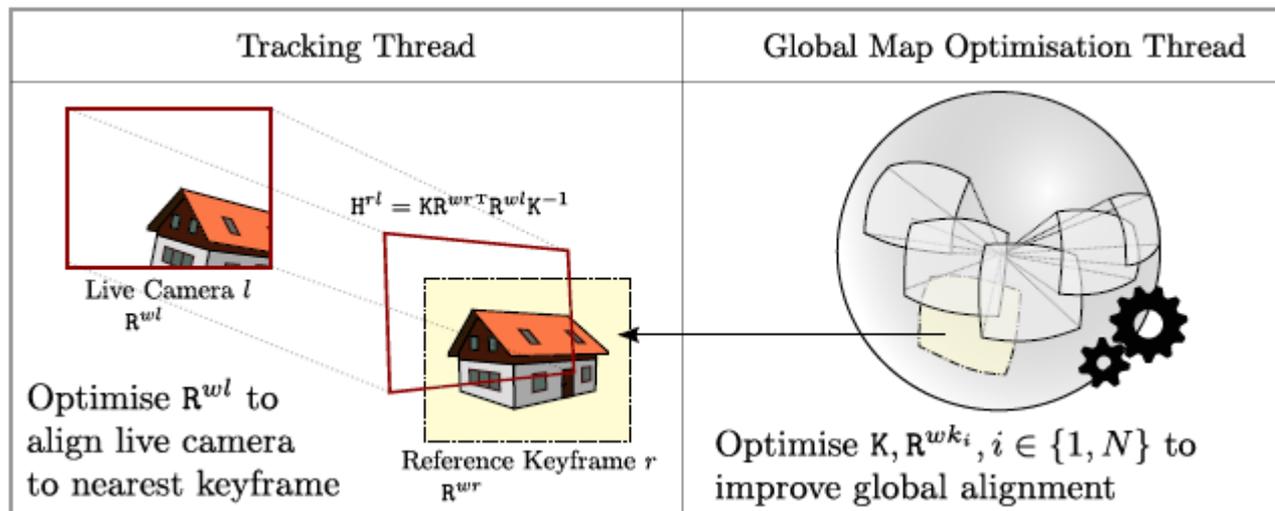
Vladyslav Usenko

Contents

1. Task Clarification
2. Solution Implementation
3. Encountered Problems
 - Global optimization: regularization
4. ROS Interface
5. Documentation

Task Clarification

Implement real-time video mosaicing algorithm based on parallel image alignment via ESM and global optimization of a map of keyframes over the whole viewsphere.



Solution Implementation

- Local Motion Estimation
 - **Status**: done + bugs fixed
 - Comment: whole image alignment + ESM
- Global Map Optimization
 - **Status**: implemented. Code and math review required.
 - Comment: ESM
- GPU implementation using OpenCL
 - **Status**: already using pyopencl
 - Comment: memory vs. performance

Solution Implementation

- Real-time video camera tracking
 - **Status**: cannot be achieved due to hardware limitations
- Parallel threads implementation
 - **Status**: already done in camera tracking
 - Comment: may need revision in ROS implementation
- ROS implementation
 - **Status**: done
- System performance/accuracy estimation
 - **Status**: planned

Encountered Problems

1. Global optimization: regularization
2. Boundary effects for gradient computation:
 - Computing image gradients, we used Mirror Reflection for point outside boundaries
3. Accumulated error in rotation matrices:
 - Rodrigues formula to enforce orthonormality
 - SVD also might be used

Global optimization: regularization

$$f(x) = \frac{1}{2} \sum_j \sum_i \sum_{p_j \in \Omega_j} \left[\mathcal{I}^i(\mathbf{H}^{ij}(x)p_j) - \mathcal{I}^j(p_j) \right]^2$$

$$f'(x) = f(x) + \frac{1}{2} \mu^2 (f_u e^{k_0} - f_{u \text{ init}})^2 + \frac{1}{2} \mu^2 (f_v e^{k_1} - f_{v \text{ init}})^2 +$$

$$\frac{1}{2} \mu^2 (u_0 e^{k_2} - u_{0 \text{ init}})^2 + \frac{1}{2} \mu^2 (v_0 e^{k_3} - v_{0 \text{ init}})^2$$

$$\left. \frac{\partial f'(x)}{\partial k_0} \right|_0 = \mu f_u (f_u - f_{u \text{ init}})$$

$$f'_{k_0}(0) = \mu (f_u - f_{u \text{ init}})$$

ROS Interface

- Subscribe to
 - “/camera/image_color_rect”
 - “/camera/camera_info”
- Publish
 - “/panorama/image”

Documentation

- You can find source code with its description, documentation and usage examples here:
https://github.com/tum-uav/rvc_team5/tree/master/spherical_mosaicing