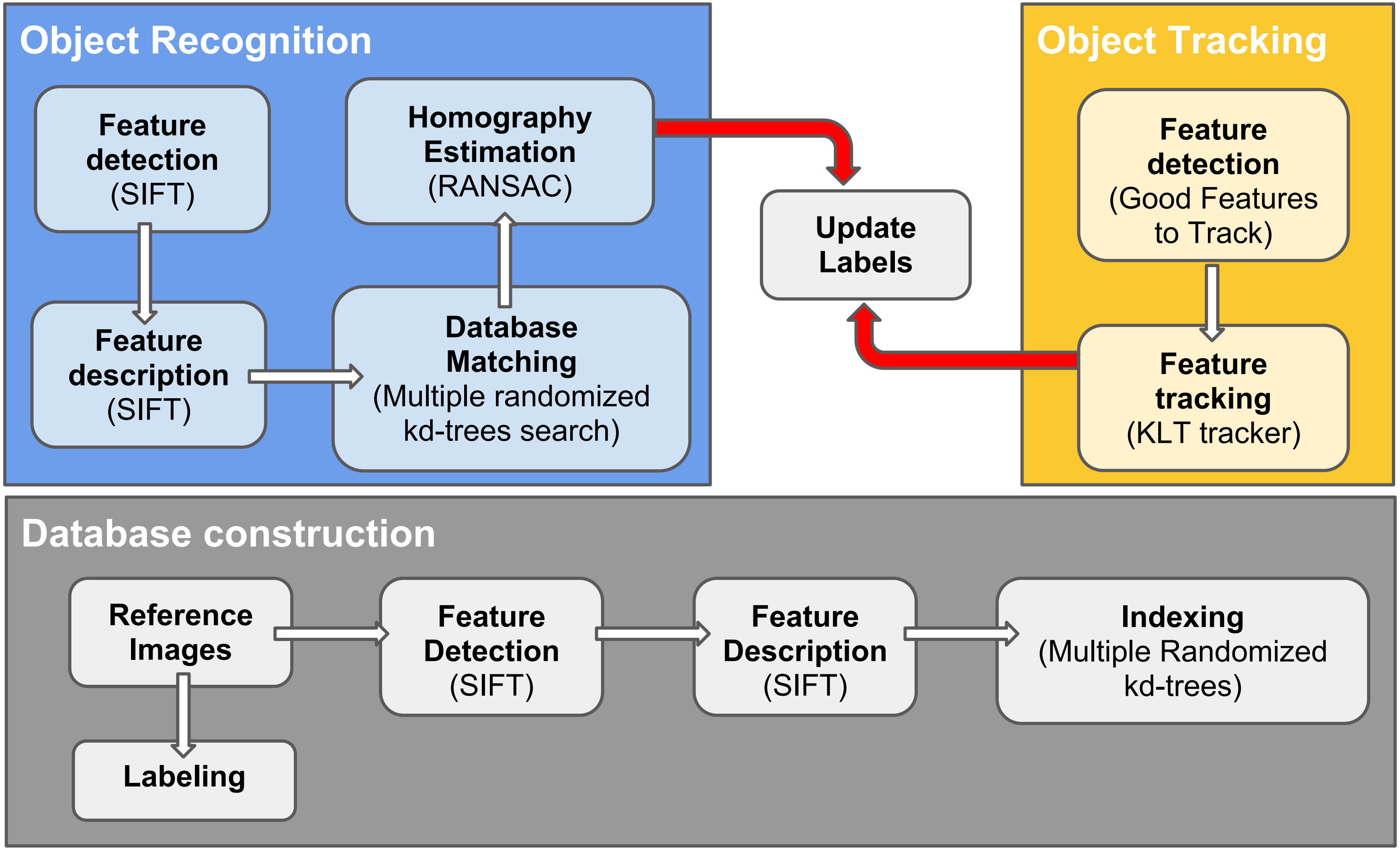
Joint 3D object recognition and tracking with OpenCV

M.Zucchelli R.Orizio M.Rizzini

Scheme



Object Recognition

Feature detection (SIFT)

Feature description (SIFT)

Database Matching (Multiple randomized kd-trees search)

Homography Estimation (RANSAC)

Update Labels

Object Tracking

Feature detection (Good Features to Track)

Feature tracking (KLT tracker)

Database construction

Reference Images

Labeling

Feature Detection (SIFT)

Feature Description (SIFT)

Indexing (Multiple Randomized kd-trees)

OpenCV Functions

- **Object Recognition/Database construction**
  - Feature Detection: `FeatureDetector::create( "SIFT") FeatureDetector::detect`
  - Feature Description: `DescriptorExtractor::create( "SIFT"), DescriptorExtractor::compute`
  - Database Indexing: `flann::Index_<T>::Index_, flann::KDTreeIndexParams;`
  - Database Matching: `flann::Index_<T>::knnSearch`
  - Homography Estimation: `findHomography(...., CV_RANSAC,...)`

- **Object Tracking**
  - Feature Detection: `FeatureDetector::create( "GFTT"), FeatureDetector::detect`
  - Feature Tracking (KLT Tracker): `calcOpticalFlowPyrLK`

Tips/Details

- **Recognition:** Do recognition every N frames, and tracking in-between (e.g. N = 20)
- **Tracking:** Once you found the object, track only the "relevant" GFTT features (i.e. on the object!)
- **Good labels:** Define (manually) the labels, well spread on the whole object (front, back, side)
- **Label update:** Don't track the pixel correspondent to the labels, they could be unstable features (on non-textured area). Update label positions using the 2D motion estimated on the GFTT features (use only the nearest to the label?)
- **Use an object which is rigid and has some texture/detail on it!**