

# 1. Analysis: SLAM++

## 1.1. Problem Statement

The authors of this paper propose an approach to the SLAM problem using a combination of the KinectFusion algorithm along with an efficient graph based 3D object recognition system. According to them, this approach offers several advantages of existing SLAM systems in operation that operate at the level of low level primitives (i.e. points, lines, etc).

## 1.2. Innovative Contribution

The SLAM problem has been approached from the perspective of 3D object recognition before. However these methods generally reveal huge amounts of wasted computational effort via repeated low level geometry processing of the 3D objects. To counter this, the authors propose the building of pose graph maps based on an 'object-oriented' approach that directly encodes the positions of recognized 3D structures. With each new measurement, the graph is continually optimised with new measurements from the sensors and allows for efficient tracking of the camera system based on recognized landmarks. In addition to this, the algorithms make the assumption that the world has "intrinsic symmetry in the form of repetitive objects" thereby allowing for the the objects in a scene to be identified and segmented as salient repeated elements. The algorithm leverages this repetitiveness along with the efficient use of GPU architectures to provide a real time processing system.

## 1.3. Proposed Method

**Creating an Object Database** - The authors first create a database of repeatedly occurring objects via known KinectFusion algorithms. These are objects that are subsequently recognized and used in their SLAM process.

**SLAM Map Representation** - The authors represent the world via a graph where each node stores the 6DOF pose of discovered objects relative to a fixed world frame as well as an annotation of the type of the object from the earlier created database.

**Real-Time Object Recognition** - This portion of the method recognizes objects in the world based on standard mesh recognition algorithms. The implementation is parallelized on GPUs to allow the real-time detection of multiple instances of multiple objects. These correspondences are obtained via the use of Point-Pair Features (PPFs) which are four dimensional descriptors.

**Camera Tracking and Object Pose Estimation** - The iterative closest point (ICP) algorithm is used to track the pose of the camera model based on the earlier computed object based locations. A Huber penalty function is used to in this optimization process. Criteria is developed to ensure successful convergence of the trackign error.

**Graph Optimization** - The poses of the static object is now viewed as a graph optimization problem which minimizes the sum over all the measurement constraints based on the known features of each object. T

**Relocalization** - The system accounts for a loss in camera tracking by re-initilazing localization based on matching at least 3 of the objects seen in the previously tracked long-term graph.

## 1.4. Experimental Evaluation

The authors reference a video submitted along with this paper to CVPR as a better description of the advantages of their method.

**Loop Closure** - Small loop closures are detected and compensated for by the ICP algorithm. Larger loop closures are compensated for via the use of the relocalization method.

**Large Scale Mapping** - Scaled mapping of a large room (15mX10mX3m) was obtained along with the mapping of 34 different objects around the room. The algorithm uses no priors regarding the original placement of these objects.

**Moved Object Detection** - The algorithm also displays the ability to track these objects while they, themselves are in motion.

**System Statistics** - The algorithm displays the amount of storage used as compared to the more traditional KinectFusion algorithm. The given mapped rooms is stored in about 20MB of space as compared to the 1.4GB used by KinectFusion. The resultant compression ratio is 1/70 which is a dramatic improvement.

## 1.5. Subsequent Conclusions

The paper makes several bold claims with regard to its own contributions to the literature. The graph based optimization method does indeed seem novel and the system has significantly large data compression ratio when compared to the KinectFusion algorithm. UnfortunatelyThe experimentally evaluated conclusions are quite sparse when compared to the dense and well written introduction and methodology. The biggest issue pertains from the fact that no standard metric is used to compare the system's advantages and efficieny to other 3D object recognition based SLAM approaches. Thus, though several claims are made about the paper's SLAM advantages over other methods, there are no easy ways to determine the validity of these claims.

## **2. Analysis:**

**Good Features to  
Track for Visual SLAM**

### **2.1. Problem Statement**

### **2.2. Innovative Contribution**

**System Observability Measure**

**Rank-k Temporal Update of Observability Score**

### **2.3. Proposed Method**

**Computation of System Observability Measure**

**Computation of Rank-k Temporal Update of Observability Score**

**Submodular Learning for Feature Grouping**

### **2.4. Experimental Method**

**Moved Object Detection - Ego motion estimation**

### **2.5. Subsequent Conclusions**

**It's pure shit**