3D Reconstruction with Tango

Ivan Dryanovski, Google Inc.
Contents

Problem statement and motivation
The Tango SDK
3D reconstruction - data structures & algorithms
Applications
Developer tools
Problem formulation

**Goal:** Create accurate, textured 3D models of indoor spaces
- real-time, on mobile device
- offline, in the cloud

**Motivation**
- immersive VR / AR - gaming, professional apps
- large scale maps, floorpans - navigation
Problem formulation

Challenges:

- Partial, noisy depth data
- Memory and computational constraints
- Platform diversity
Tango software stack

Motion Tracking

Area Learning

Depth perception
Tango 3D reconstruction

Real-time, online process

6DoF Pose 100Hz

Color images 10Hz

Depth images 10Hz

Volumetric fusion
- using two-tier TSDF voxel grid

Mesh extraction
- using Marching Cubes

Application tasks
- rendering
- occlusions
- path planning
- collision detection

Tango 3D reconstruction

Offline process

6DoF Pose

Volumetric fusion
- using two-tier TSDF voxel grid

Mesh extraction
- using Marching Cubes

Mesh simplification
- vertex decimation

Mesh texturing
- texture atlas generation
- projective image texturing

Application tasks
- rendering
- occlusions
- path planning
- collision detection
Signed Distance Function (SDF)

Represent space using **voxel grid**

Voxel contains signed distance function to nearest surface (**SDF**)

Only update near surface - estimate truncated function (**TSDF**)

Grid is updated by fusing depth readings (using running **weighted average** filter)

Signed Distance Function (SDF)

Surface of objects is represented by the **zero-isosurface** in voxel grid.

Traversing the isosurface is done using **Marching Cubes**.

Output is a **triangle mesh**.

---

Voxel memory layout

64-bit per voxel (includes TSDF value, color value, filter weights)
Dynamic Two-Tier TSDF voxel grid

Single TSDF grid for the entire scene would require **too much memory**

Hierarchical grid - larger **volumes** contain sub-grids

Volumes are sparsely allocated, and indexed by a **hash map**
Local frustum culling

When adding new observation, do intersect the **depth camera frustum** with the coarse volume grid

New volumes are allocated on-demand
Local frustum culling

Only intersected volumes receive TSDF updates

Only intersected volumes have their mesh segments re-extracted

Allows building of large models while updating and streaming only local segments

Original mesh with vertex colors
Large-scale reconstruction

25 individual trajectories (~3 hrs)
Co-located using area learning

Final 3D map (false color by height)
Large-scale reconstruction

Ivan sits here
Developer tools

Tango Client API

- C / Java / Unity
- 6DoF pose, images, depth
Developer tools

- **Tango Client API**
  - C / Java / Unity
  - 6DoF pose, images, depth

- **Tango 3D Reconstruction API**
  - C / Java / Unity
  - Provides textured 3D meshes
Developer tools

Constructor
- Android app for building and sharing 3D models

Tango Client API
- C / Java / Unity
- 6DoF pose, images, depth

Tango 3D Reconstruction API
- C / Java / Unity
- Provides textured 3D meshes
GPU implementation

If no shared memory between CPU & GPU, copying data is a bottleneck

Volumetric fusion
- using two-tier TSDF voxel grid

Mesh extraction
- using Marching Cubes

Application tasks
- rendering
- occlusions
- path planning
- collision detection

Depth images as graphics texture?
Collision calculations on graphics hardware?
Thank you!

Q&A

Ivan Dryanovski, Google Inc.