

Real-Time 3D volumetric human body reconstruction from single view RGB-D capture device

Rafael Diniz, Mylène C.Q. Farias
rafael@riseup.net, mylene@ieee.org

Department of Computer Science
University of Brasília



Introduction

Table of Contents

- Introduction
- 3D reconstruction from single view RGB-D
- Results
- Conclusions

Introduction

Volumetric video

- Volumetric Video is a promising new technology being pushed forward by most of the Mixed Reality systems being developed;
- Typically employ Point Cloud or Mesh data structures;
- Acquisition of volumetric video needs an array of accurately calibrated sensors, which has no easy setup.

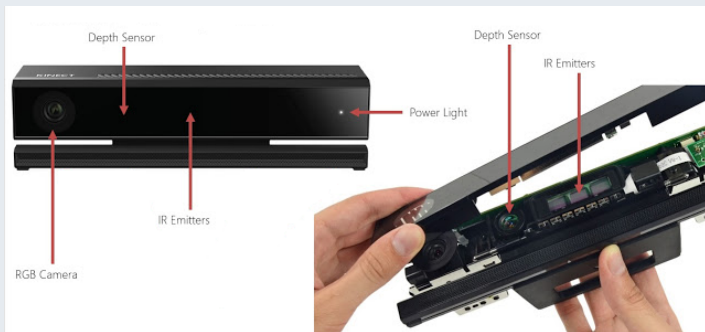
Motivation

- Contribute to increase volumetric video use cases without complex capture setups.

Introduction

Kinect 2 capture device

- Time-of-Flight depth ranging technology
- Depth resolution of 512x424, distances from 0.5 to 4.5m, FoV of 70.6° by 60° (HxV), millimeter accuracy
- RGB in HD, downconverted to match depth resolution



Introduction

Typical volumetric video capture setup

Example with 4 Kinect 2 devices.



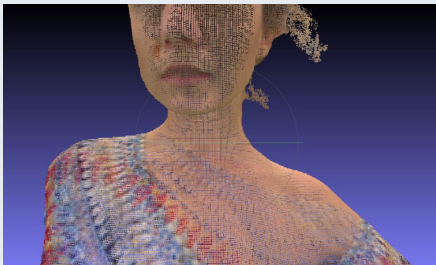
3D object reconstruction method examples

- Based on registration and fusion of RGB-D frames
- Priors based - objects shape database
- Data-driven based - using machine learning
- Approaches for specific cases, like human face or human body

3D reconstruction from single view RGB-D

Volumetric Video Formats

- Point-clouds: 3D coordinates + color (+other attributes) - Used in this work
- Meshes: Polygonal surfaces (Edges + Vertices).



3D reconstruction from single view RGB-D

Proposed system - brief description

- Pre-capture a 3D model of the object in advance;
- Just 1 RGB-D capture device
- A process to recreate each volumetric video frame from the 3D model and RGB-D stream

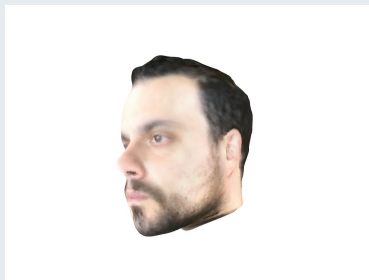
Constraints

- Assumes that the back of the head of the person is non-deformable;
- The speaker is looking ahead during most of the time;
- Self-occlusions do not occur often.

3D reconstruction from single view RGB-D

Model capture

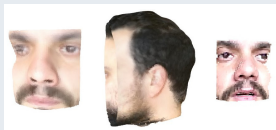
- Based on the Truncated Signed Distance Function and Kinect Fusion;
- Assembles the 3D object by moving the capture device around the object.



3D reconstruction from single view RGB-D

3D reconstruction

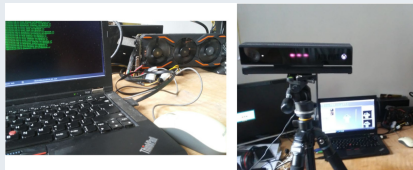
- Live RGB-D frames are pre-processed and converted to point-cloud;
- Model and input RGB-D point-clouds are segmented and registered;
- Registered model and input frames are segmented and merged.



3D reconstruction from single view RGB-D

Implementation and Experimental Setup

- Code written from scratch in C and C++;
- Using Open3D and libfreenect (1 and 2) library;
- Kinect 2 as main capture device;
- Intel Xeon E5-2620, with 80GB of RAM hardware;
- Lenovo T430 for mobile capture device.



3D reconstruction from single view RGB-D

Results

- Realtime CPU execution - under 33ms at 30fps;
- Better experience when compared to incomplete objects;
- Allows volumetric video use cases using single RGB-D capture device;
- Room left for GPU offloading optimization;
- Can be extended to different type of objects;



3D reconstruction from single view RGB-D

Conclusion

- Work shows Mixed Reality Volumetric Video use cases using single RGB-D camera running in an affordable CPU is possible.

Issues to be addressed

- Solve the color difference problems between model and input frame;
- Fusion / Merge of Point Cloud issues;



3D reconstruction from single view RGB-D

Input: volumetric pre-captured model, RGB stream, Depth stream

Output: recontructed volumetric point-cloud frames

$model_pc \leftarrow pre_captured_model$

$model_face_pc \leftarrow get_face(model_pc)$

$model_no_face \leftarrow model_pc - model_face_pc$

while $rgb_d_camera_state = capturing$ **do**

$color \leftarrow input_rgb$

$depth \leftarrow input_depth$

$color \leftarrow registration_get_color(depth, color)$

$depth \leftarrow registration_get_depth(depth, color)$

$pc \leftarrow create_point_cloud_from_rgb_d(depth, color)$

$face_pc \leftarrow get_face(pc)$

$transform_matrix \leftarrow$

$fast_pc_registration(model_face_pc, face_pc)$

$local_model \leftarrow transform(transform_matrix, model_no_face)$

$reconstructed_pc \leftarrow merge_pc(local_model, pc)$

end

3D reconstruction from single view RGB-D

Thanks.