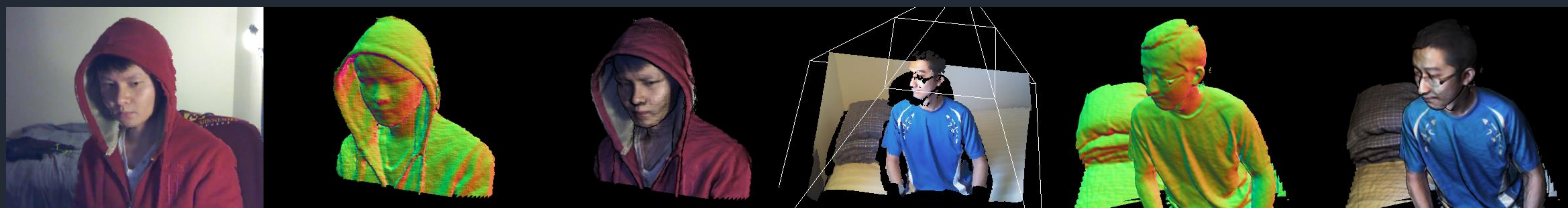
# Kinect Scanner

## 3D Color Model Reconstruction System


**Peng Liu**

University of Minnesota

## Introduction

3D color models of the real world, such as indoor scenes, the human body and objects around our daily life, are useful for applications in computer games, animation and virtual reality. Accurate model recreation systems, however, either require very expensive scanners or require tedious effort using modeling software. Cheap depth sensors like the Microsoft Kinect, which have the potential to facilitate the model creation task, have attracted tremendous attention in academia. However, most research efforts have been aimed at doing pose tracking or robotic navigation, and few have focused on using these depth sensors to do color model reconstruction. In this paper we present such a depth sensor based system, which uses a Kinect to do 3D color model reconstruction. The system uses an incremental updating scheme based on the KinectFusion algorithm[1]. With this fast and affordable color modeling system (only a Kinect and decent graphic card are needed), creating real world objects can be done more easily than before, which promises a more efficient development process in model creation related applications.

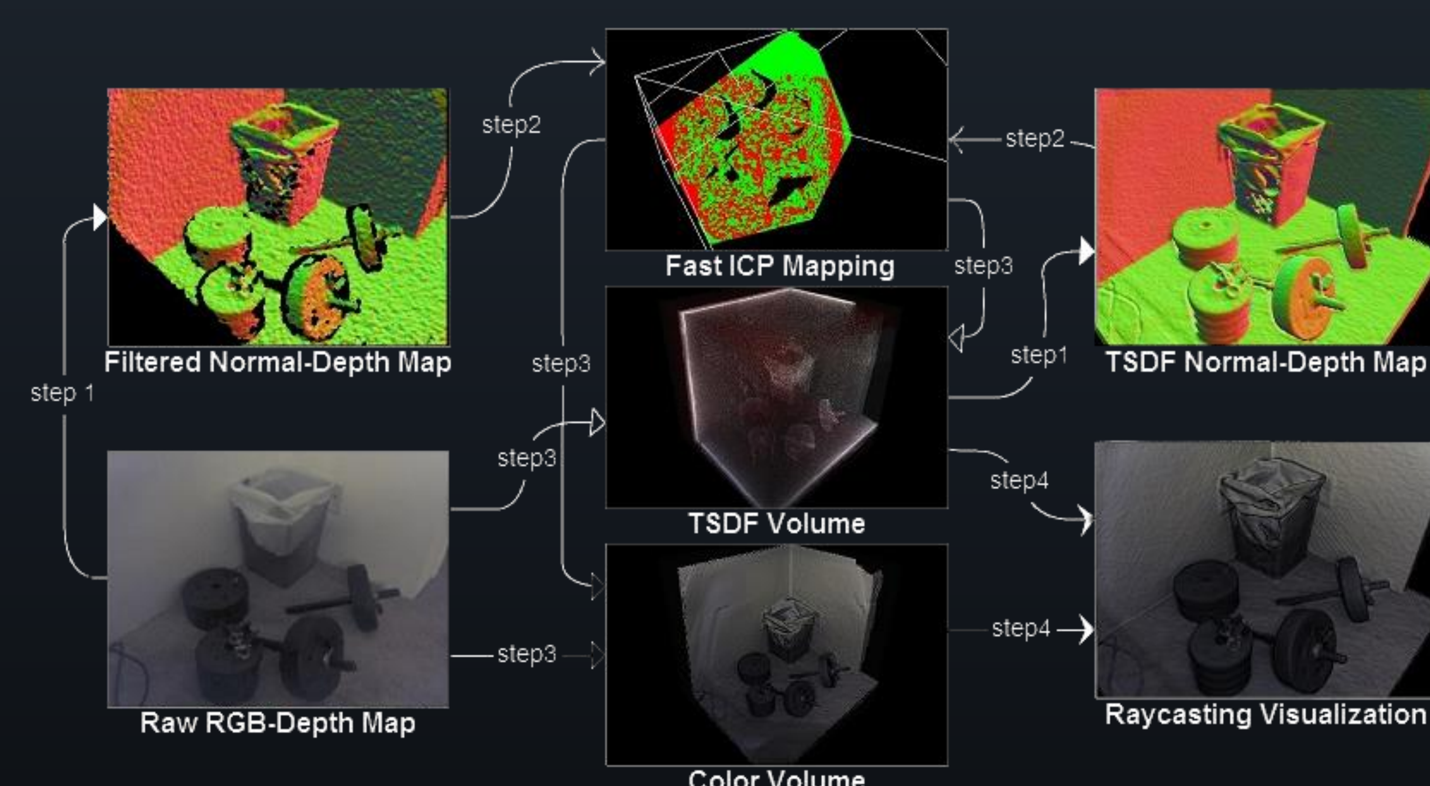
## System Work-Flow

### Step 1:

After each new raw RGB-Depth map has been received from the Kinect sensor, the data is filtered to get rid of noise, then a normal-depth map is generated. At the same time, another normal-depth map is extracted from the TSDF volume for mapping.

### Step 2:

Given two RGB-Depth maps from Step1, the fast ICP algorithm (See Key Process I below) computes the rigid transform matrix to align these normal-depth maps. Also by continuously multiplying this matrix with previous one, the system can keep track of the Kinect pose.




### Step 3:

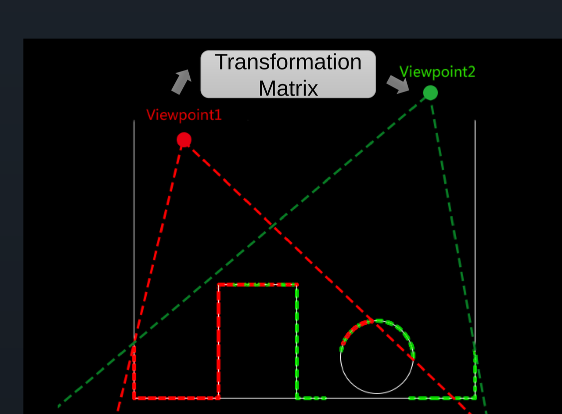
Since one RGB-Depth map is extracted from the TSDF, which represents the global model, then with the new RGB-Depth map along with the matrix from Step2, the system can correctly update both the TSDF volume and the Color volume by incorporating the new measured surface.

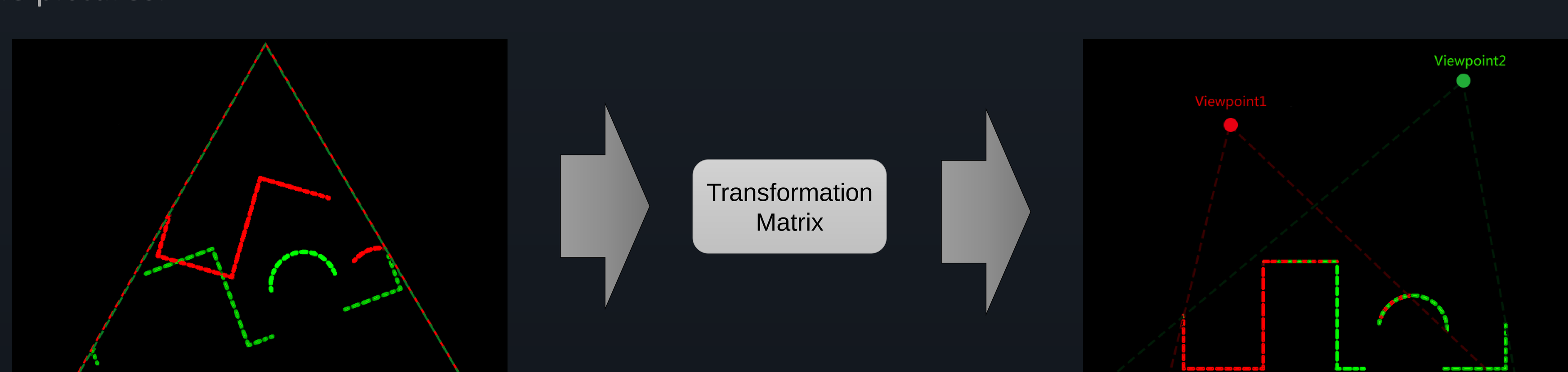
### Step 4:

After model updating, a ray-casting algorithm will sample both the TSDF and the Color volume to generate a Phong shaded image of the target model from the Kinect's point of view

## Key Process I: Surface Alignment

The Kinect is not tracked, so from the Kinect's point of view, each depth frame is independent from each other, even though they represent the surfaces of the same target. So the system needs to know the relationship between each surface; in other words, the system needs to know the rigid body transformation matrix that can bring the red surface and the green surface into alignment as shown in the pictures.

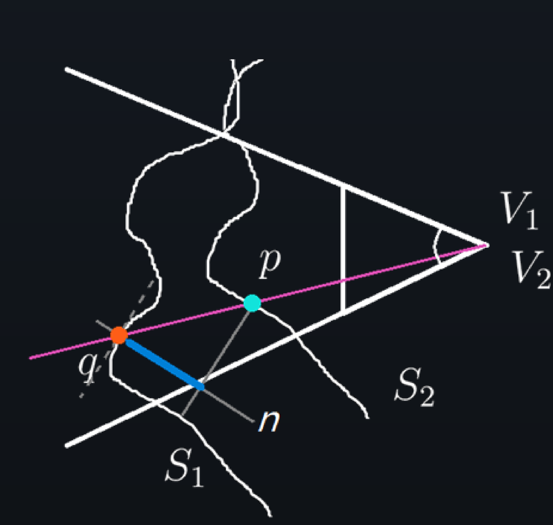



To align two surfaces, we use Fast ICP[2]. Basically, for each point pair $p, q$ from each surface, we use an iterative method to find an optimal rotation and translation matrix ($R$ and $t$) to minimize the alignment error $E$ as shown below:

$$E = \sum_i [(Rp_i + t - q_i) \cdot n_i]^2$$

It is the sum of the alignment error for all point pairs, and the error for each point pair is the length of the blue line shown on the right ($S_1, S_2$ are measured surfaces from view point $V_1, V_2$ and $n$ is the surface normal at point $q$)

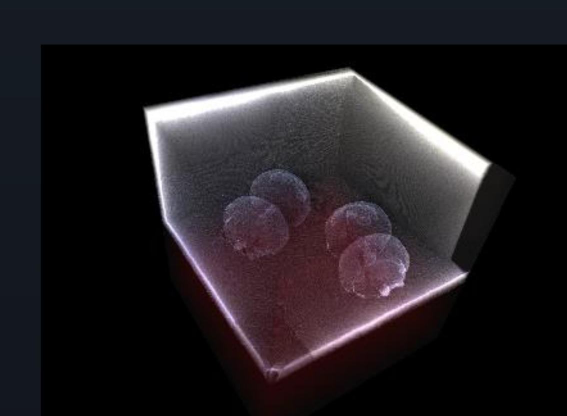


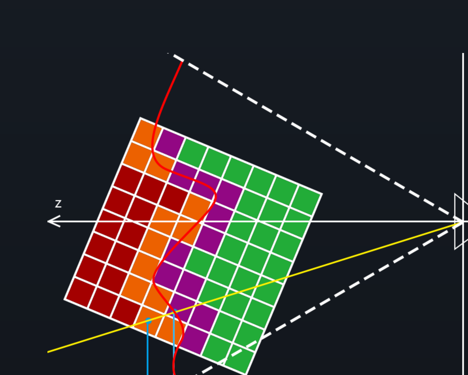

## Key Process II: Model Updating

### Model Representation: TSDF

The widely used vertex based mesh model cannot be used in our system, for the updating phase will generate nearly 300000 vertices each frame and no real time system can keep merging these vertices for more than few minuets. Instead we use the grid based TSDF model representation technique.

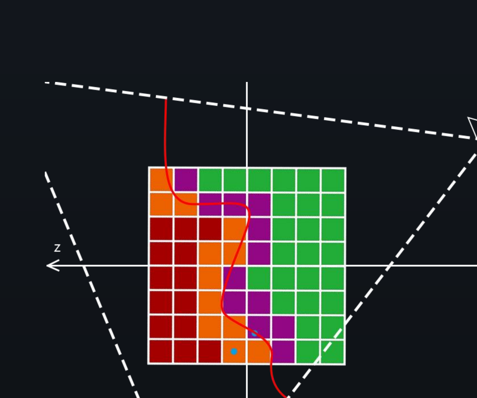
The picture on the right shows the 2D version of TSDF. Each cell in the TSDF grid stores the truncated distance to the nearest surface, thus the geometric information can be represented by the zero-crossing surface in the TSDF grid.





### Updating

Once our system has the transformation matrix, the updating phase is straight forward. With the proper settings of the TSDF volume (voxel size and volume resolution), we can get each voxel's coordinate in world space. And with the transformation matrix we get from the surface alignment, we can transform the voxel coordinate into view space. Then we project the voxel's 3D coordinate into image plane to find the texture index to sample the depth image for the z value, and then get the truncated distance for each voxel to update the whole TSDF volume along with the color volume, as shown on the right.



## Reference


[1] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," 2011 10th IEEE International Symposium on Mixed and Augmented Reality, pp. 127{136, Oct. 2011.

[2] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," Image and Vision Computing, 10, 3 (April 1992) pp. 145-155.



## Acknowledgements

Linda & Ted Johnson Digital Design Consortium Endowment and Lab Setup Funds