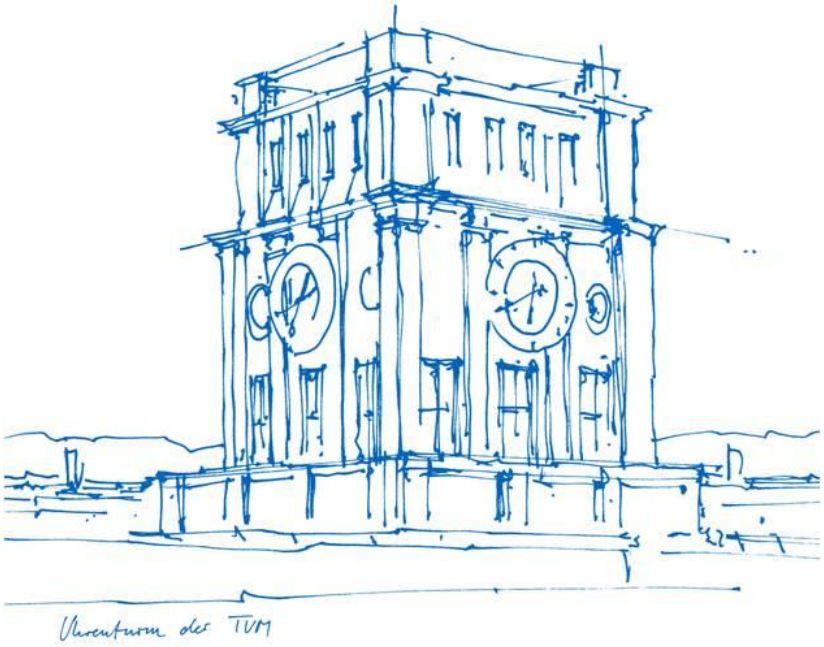
KinectFusion: Real-Time Dense Surface Mapping and Tracking

Seminar: The Evolution of Motion Estimation and Real-time 3D Reconstruction

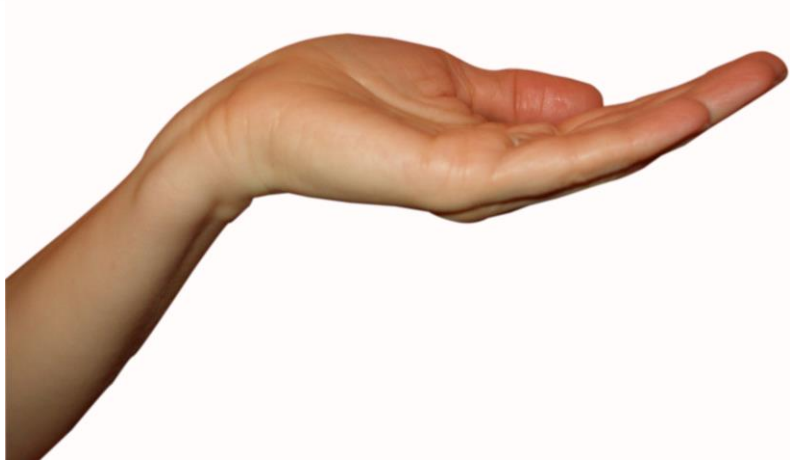
Qiao Qiao

Technical University of Munich

Munich, 15.April.2020

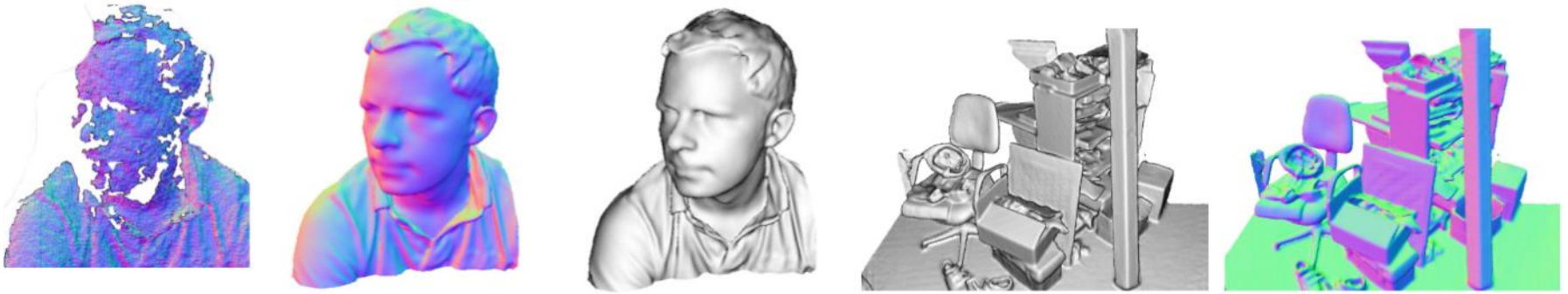


What do we have?



KinectFusion: Real-Time Dense Surface Mapping and Tracking

Newcombe, Izadi, Hilliges, Molyneaux, Kim, Davison, Kohli, Shotton, Hodges, Fitzgibbon 2011 [1]



Keywords: Real-Time, Dense Reconstruction, Frame-to-Model Tracking, GPU, SLAM, Depth Cameras, TSDF, AR

First system which is able to perform a real-time, dense 3D model reconstruction of room sized scenes without any landmarks using a hand-held Kinect depth sensor

Outline

- Introduction
- Method description
- Experiments
- Personal comments
- Summary

Kinect Sensor

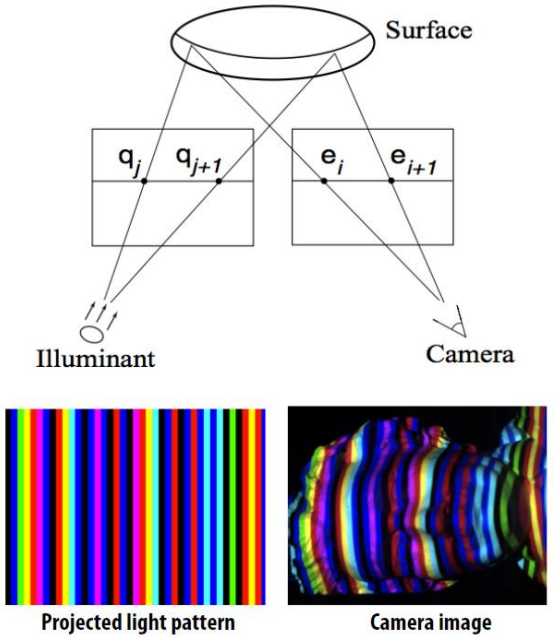
Commodity sensor(30 € - 300 €)

Colour map (don't used in this paper)
Depth map – structured light based

30Hz frame-rate
Min and max sensor range: 0.4m-8m

Problem:
holes ← structures do not reflect infra-red light
Blur ← moving fast

Advantage of only using depth map:
no lighting condition

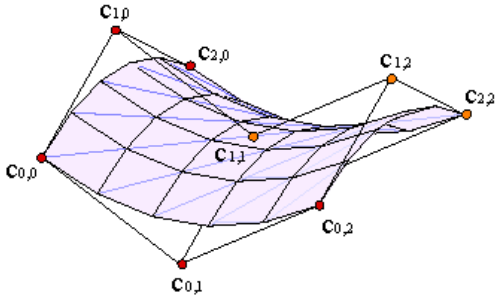


structure light [7]

Left: raw noisy depth map (input)
Right: Reconstructed model [1]

Surface Model

Explicit model: e.g. spline surface & Control points



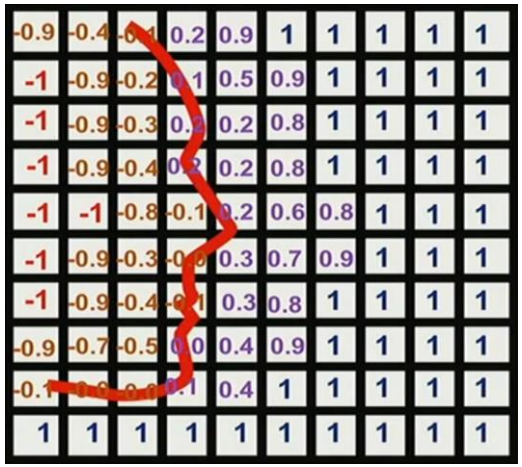
Explicit model [4]

Implicit model: e.g. Signed Distance Function (SDF)
Surface interfaces → 0 → easily extractable
Free space → positive
Occupied space → negative

In this paper: Truncated Signed Distance Function (TSDF)
→ avoid surfaces interfering

Averaging weighted TSDF into the global frame

Can directly apply a raycast to get the 2D view



Implicit model [4]

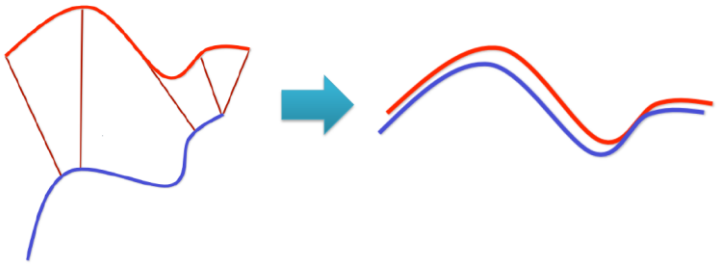
Sensor Pose Estimation

First real-time monocular SLAM → probabilistic filtering → high computational cost → small scale scene

In this paper: frame to model tracking VS previous papers: frame to frame tracking

Coarse to fine alignment to speed up
low-resolution depth map → finer levels
It also allows faster motion

Iterative closest point (ICP)
Use to estimate consecutive poses
Assume closest points are corresponding
Iterate to find alignment
Converge if poses are close enough



ICP alignment [5]

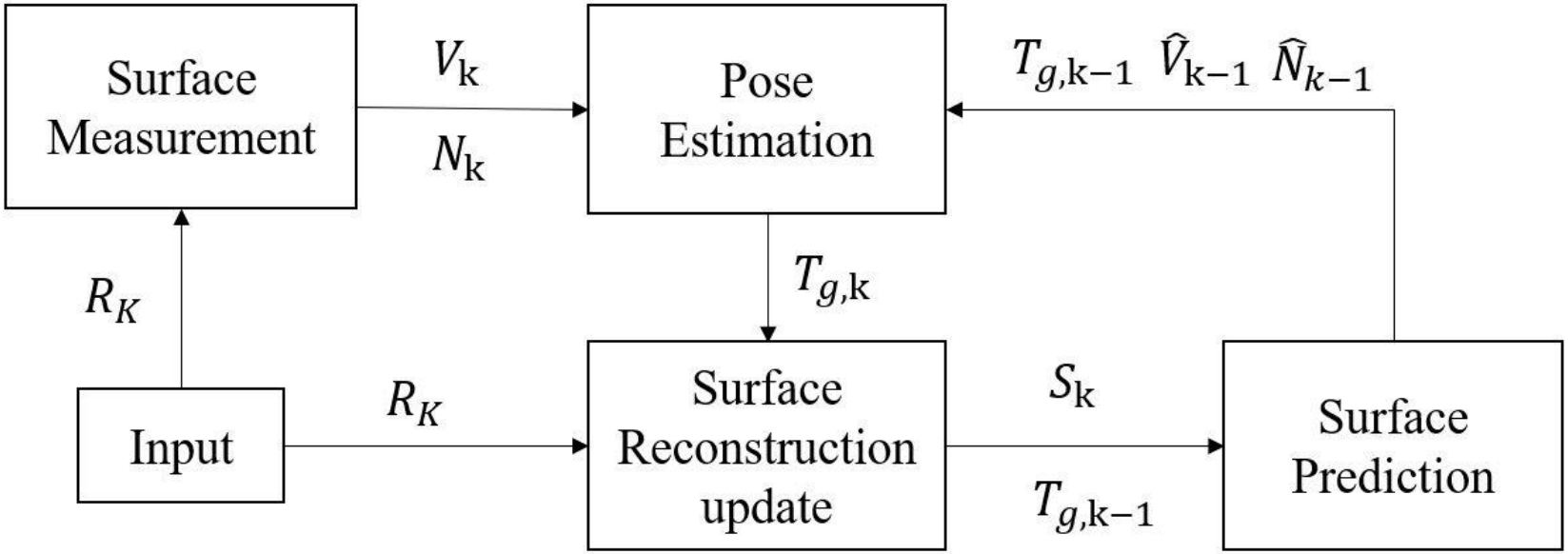
Augmented Reality (AR)

highly parallel GPU techniques & GPU → 3D reconstruction in real-time

Enable many AR and interaction applications

Pokemon go [6]

Methode description



4 Steps:
Surface measurement
Surface reconstruction update
Surface prediction
Pose estimation

R_k: Raw Depth Map
S_k: Global TSDF Model
V_k: Vertex Map
N_k: Normal Map
$T_{g,k}$: Camera Pose at time k in global frame

Surface measurement

Pre-processing

Input: Raw depth map & camera calibration information
Output: vertex map & normal map pyramid (3 levels)

Raw depth map → bilateral filter → depth map with reduced noise → Back-project →

vertex map → cross product between neighbouring vertices → normal map

Additionally: Vertex validity mask

Surface reconstruction update

Global scene fusion step

Depth map & estimated camera pose → fusion → one global TSDF model

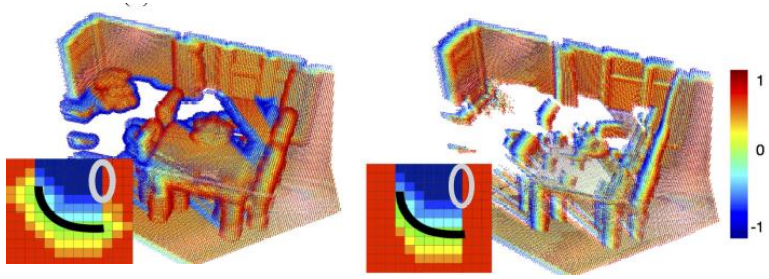
Global surface fusion: weighted averaging the TSDF of multiple 3D surface measurement

Current value & weight stored in global GPU memory

Assumption:
true value lies within ±μ of the measured value

Projective TSDF → easy to compute & parallelisable

Use raw depth map for fusion, not denoised version



TSDF projective TSDF [3]

Surface prediction

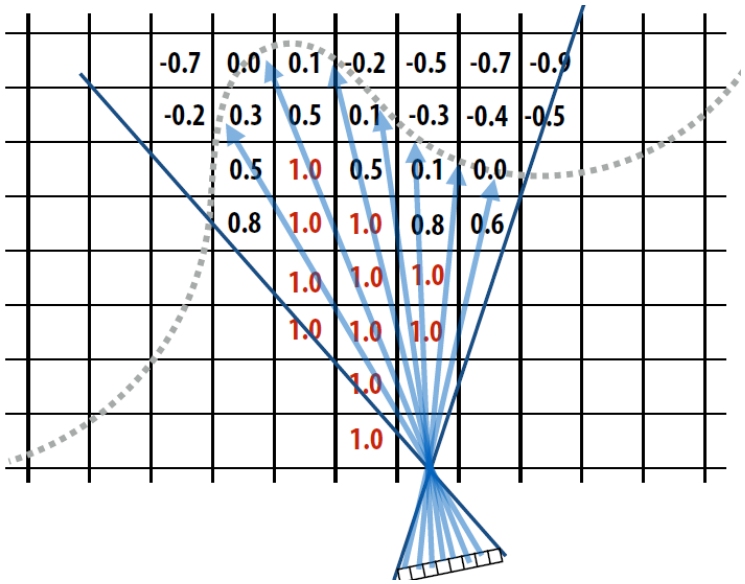
Raycasting the global TSDF into the estimated frame position (virtual camera)

Surface prediction stored as a vertex and normal map

Ray skipping → skip empty space → acceleration

Free space: step size μ

Near surface: more steps



Raycasting TSDF [8]

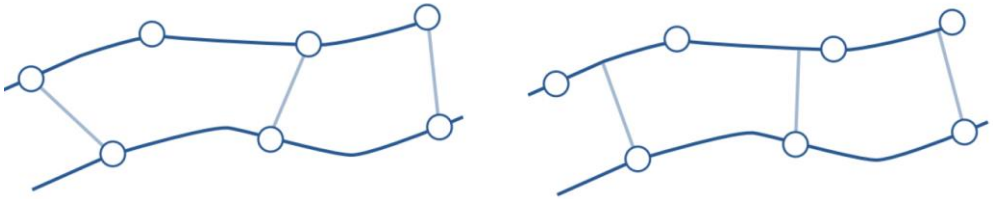
Sensor pose estimation

ICP alignment between current sensor measurement and predicted surface from last frame

Many other tracking algorithms select feature points to speed up, but this paper use all of the data for two reasons:

1. high tracking frame rate → Assume small motion between two sequential frames → Use projective data association to find vertex correspondences & point-plane metric, which can improve the convergence rate
2. GPU

Minimise global point-plane energy



Point-to-point error metric Point-to-plane error metric [5]

Minimising global point-plane energy

use projective data association algorithm → find vertex correspondences between current vertex map and predicted vertex map from last frame $\{\mathbf{V}_k(\mathbf{u}), \hat{\mathbf{V}}_{k-1}(\hat{\mathbf{u}}) | \Omega(\mathbf{u}) \neq \text{null}\}$

Valid pixel

global point-plane energy

predicted normal map from last frame

desired camera pose estimate

$$\mathbf{E}(\mathtt{T}_{g,k}) = \sum_{\substack{\mathbf{u}\in\mathscr{U} \\ \Omega_k(\mathbf{u})\neq\text{null}}} \left\| \left(\mathtt{T}_{g,k}\dot{\mathbf{V}}_k(\mathbf{u}) - \hat{\mathbf{V}}^g_{k-1}(\hat{\mathbf{u}}) \right)^\top \hat{\mathbf{N}}^g_{k-1}(\hat{\mathbf{u}}) \right\|_2$$

Small angle assumption

$$\begin{bmatrix} 1 & \alpha & -\gamma & t_x \\ -\alpha & 1 & \beta & t_y \\ \gamma & -\beta & 1 & t_z \end{bmatrix}$$

a ***linearized*** version of point-plane energy

Additional Implementations

Stability and validity check: To make sure the tracking no fail

$$\mathbf{x} = (\beta, \gamma, \alpha, t_x, t_y, t_z)^\top \in \mathbb{R}^6$$

1. A check on the *incremental transform parameters* → small angle assumption → projective data association & Linearization of the point-plane energy
2. A check on the null space of the normal system → 6 DOF is enough constrained

If either test fails: Re-localization mode

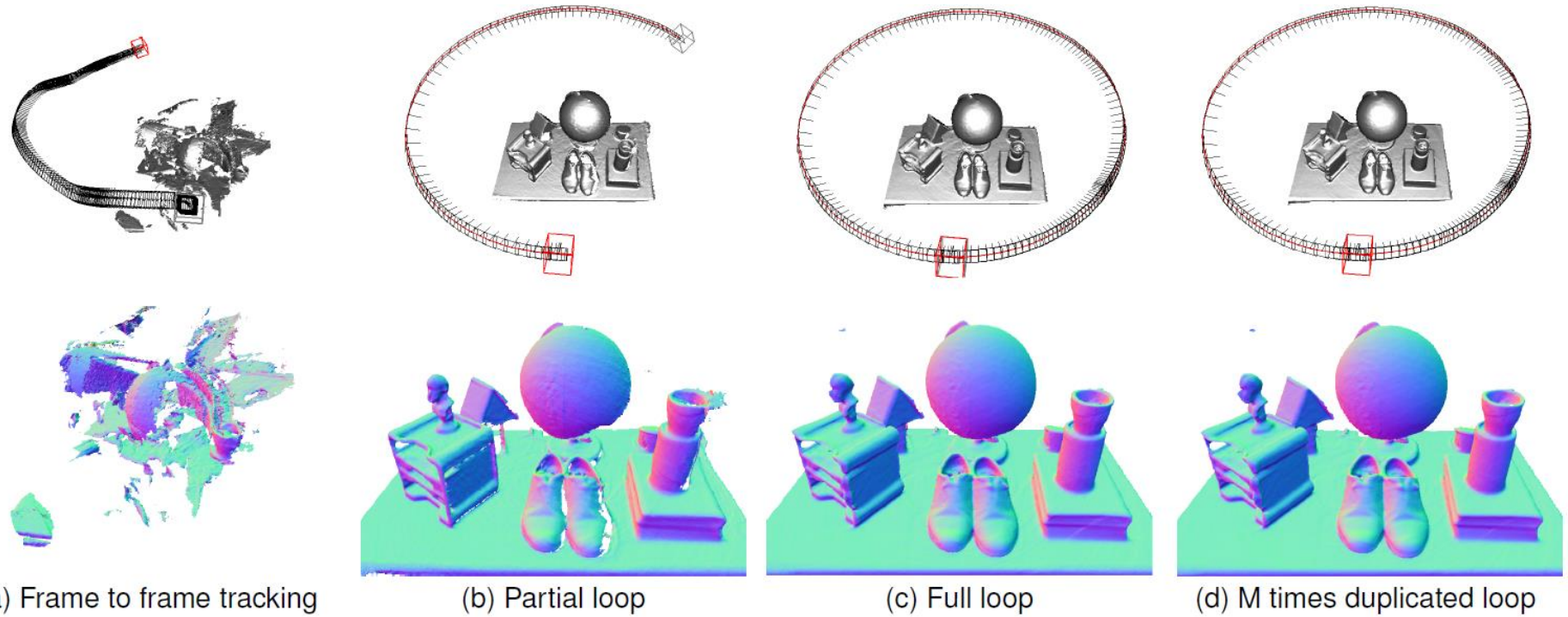
Re-localization:

If the sensor loses track → use the last known sensor pose to provide a surface prediction → Users need to align it with the upcoming depth map

Experiments

Frame-to-frame / Frame-to-model → show the ability of creating consistent models without explicit global joint-estimation

partial loop / full loop / repeat several times → show convergence properties



(a) Frame to frame tracking (b) Partial loop (c) Full loop (d) M times duplicated loop

Experiment results [1]

Personal Comments

Important: first paper generates 3D dense model using depth sensor and GPU in real-time

Creative: use all the depth data ⟷ high computational cost

GPU techniques ⟷ impossible to use in smartphone

Useful: dense surface, could be applied in AR application

Excellent performance at reducing drift

No colour geometry ⟷ higher lighting condition

Limited scene, only room size, indoor

Re-localization is not automatic!

Summary

Introduction:
Kinect hardware, surface model, pose tracking, AR

Kinectfusion method (4 steps):
Surface measurement, Surface reconstruction update, Surface prediction, Pose estimation

Experiment

Kinectfusion:
Use all of the data instead of featured based
Use only depth map instead of RGB
One single global dense model (TSDF)
Model always up-to-date
Use frame-to-model tracking instead of frame-to-frame tracking
Reconstruction in real-time because of GPU and fully parallel algorithms
Can scale according to processing and memory resources

Source

[1] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality, ISMAR '11, pages 127{136, Washington, DC, USA, 2011.
[2] G. Klein and D. W. Murray. Parallel tracking and mapping for small AR workspaces. In Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR), 2007.
[3] Shuran Song Fisher Yu Andy Zeng Angel X. Chang Manolis Savva Thomas Funkhouser Semantic Scene Completion from a Single Depth Image
[4] Daniel Cremers. TUM teaching materials lecture: Multiple View Geometry, Chapter 10
[5] Justus Thies, Angela Dai. TUM teaching materials lecture: 3D Scanning and Motion Capture
[6] https://www.pokemon.com/fr/strategie/devenez-un-utilisateur-expert-de-pokemon-go/
[7] Li Zhang, Brian Curless, and Steven M. Seitz. Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming
[8] Visual Comuting Systems CMU 15-769 lecture 16: Real-time Dense 3D Reconstruction

Question?