Efficient Multi-Frequency Phase Unwrapping using Kernel Density Estimation

Felix Järemo Lawin, Per-Erik Forssén, Hannes Ovrén
{felix.jaremo-lawin, per-erik.forssen, hannes.ovren}@liu.se

LINKÖPING UNIVERSITY

Introduction

We introduce an efficient method to unwrap multi-frequency phase estimates for time-of-flight ranging. The algorithm generates multiple depth hypotheses and uses a spatial kernel density estimate (KDE) to rank them. We apply the method on the Kinect v2.

- The Kinect v2 is designed for scenes with less than 8m range, but with our method the effective range can be extended.
- When extending the depth range to the maximal value of 18.75m, we get about 52% more valid measurements than existing drivers in *libfreenect2* and *Microsoft Kinect SDK*.
- Runs in ≈ 190Hz on a *Nvidia GeForce GTX 760* GPU. Code is available at: `http://www.cvl.isy.liu.se/research/datasets/kinect2-dataset/`

Method

In amplitude modulated time-of-flight ranging the depth is obtained by measuring the phase shift $\phi_m \in [0, 2\pi)$ of the modulated signal given the frequencies f_m.

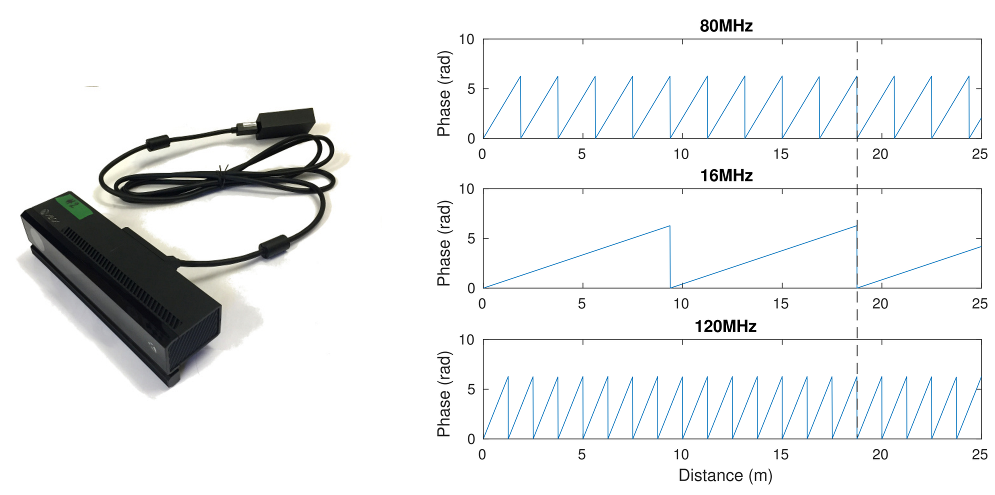

Figure 1: Left: The Kinect v2 sensor, right: wrapped phases for Kinect v2, in the range 0 to 25 meters. Top to bottom: ϕ_0, ϕ_1, ϕ_2. The dashed line at 18.75 meters indicates the common wrap-around point for all three phases. Just before this line we have $n_0 = 9$, $n_1 = 1$, and $n_2 = 14$.

- The corresponding unwrapped phase measurements are $\phi_m + 2\pi n_m$.
- Phase unwrapping means finding the unwrapping coefficients $\mathbf{n} = (n_0, \dots, n_{M-1})$.
- Each vector $\mathbf{n}$ corresponds to a hypothesis of the depth t^i.
- Our method considers several hypotheses for each pixel location and selects the one with the highest kernel density value in a spatial neighbourhood $\mathcal{N}(\mathbf{x})$:

$$p(t^i(\mathbf{x})) = \frac{\sum_{j\in\mathcal{I},k\in\mathcal{N}(\mathbf{x})} w_{jk} K(t^i(\mathbf{x}) - t^j(\mathbf{x}_k))}{\sum_{j\in\mathcal{I},k\in\mathcal{N}(\mathbf{x})} w_{jk}},$$

$$w_{ik} = g(\mathbf{x}-\mathbf{x}_k,\sigma)p(t^i|\mathbf{n}_i)p(t^i|\mathbf{a}_i)\,, K(x) = e^{\frac{-x^2}{2h^2}}.$$

The three factors in w_{ik} are:

- the *spatial weight* $g(\mathbf{x}-\mathbf{x}_k, \sigma)$.
- the *unwrapping likelihood* $p(t^i(\mathbf{x})|\mathbf{n}_i(\mathbf{x}))$.
- the *phase likelihood* $p(t^i(\mathbf{x})|\mathbf{a}_i(\mathbf{x}))$, where $\mathbf{a}_i = (a_0, \dots, a_{M-1})$, are the amplitudes.

The final hypothesis selection is then made as:

$$i^* = \arg\max_{i\in\mathcal{I}} p(t^i)\,.$$

- Inliers are classified by $p(t^i) > T$.

Results

We apply the method to depth decoding for the Kinect v2 sensor, and compare it to the *Microsoft Kinect SDK* and to the open source driver *libfreenect2*.

- Ground truth is constructed by fusing many depth frames from 9 different camera poses into one very accurate depth image.
- Raw measurements from the ground truth pose are decoded into depth images using *Microsoft*, *libfreenect2* and our method.
- A point is counted as an inlier when a method outputs a depth estimate that correspond to a correct unwrapping, which we set to be within 30cm from the ground truth.
- We evaluate on the **kitchen** dataset with maximal depth of 6.7m and the **lecture** dataset with maximum depth of 14.6m.

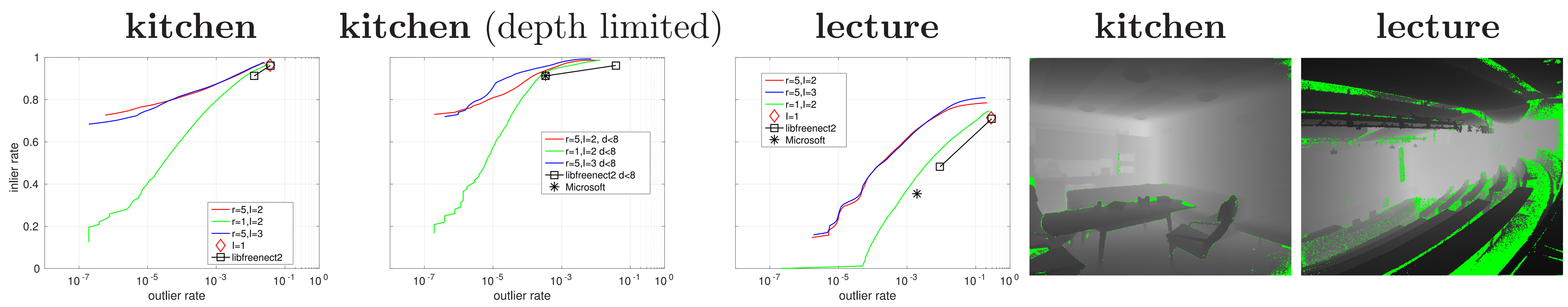

Figure 2: Inlier and outlier rate plots and corresponding ground truth depth images. Each point or curve is the average over 25 frames. In the **kitchen** (depth limited) curve the algorithms assume a maximum depth of 8m, which simplifies the outlier rejection.

- A clear improvement can be observed in the depth images, especially in large depth scenes.

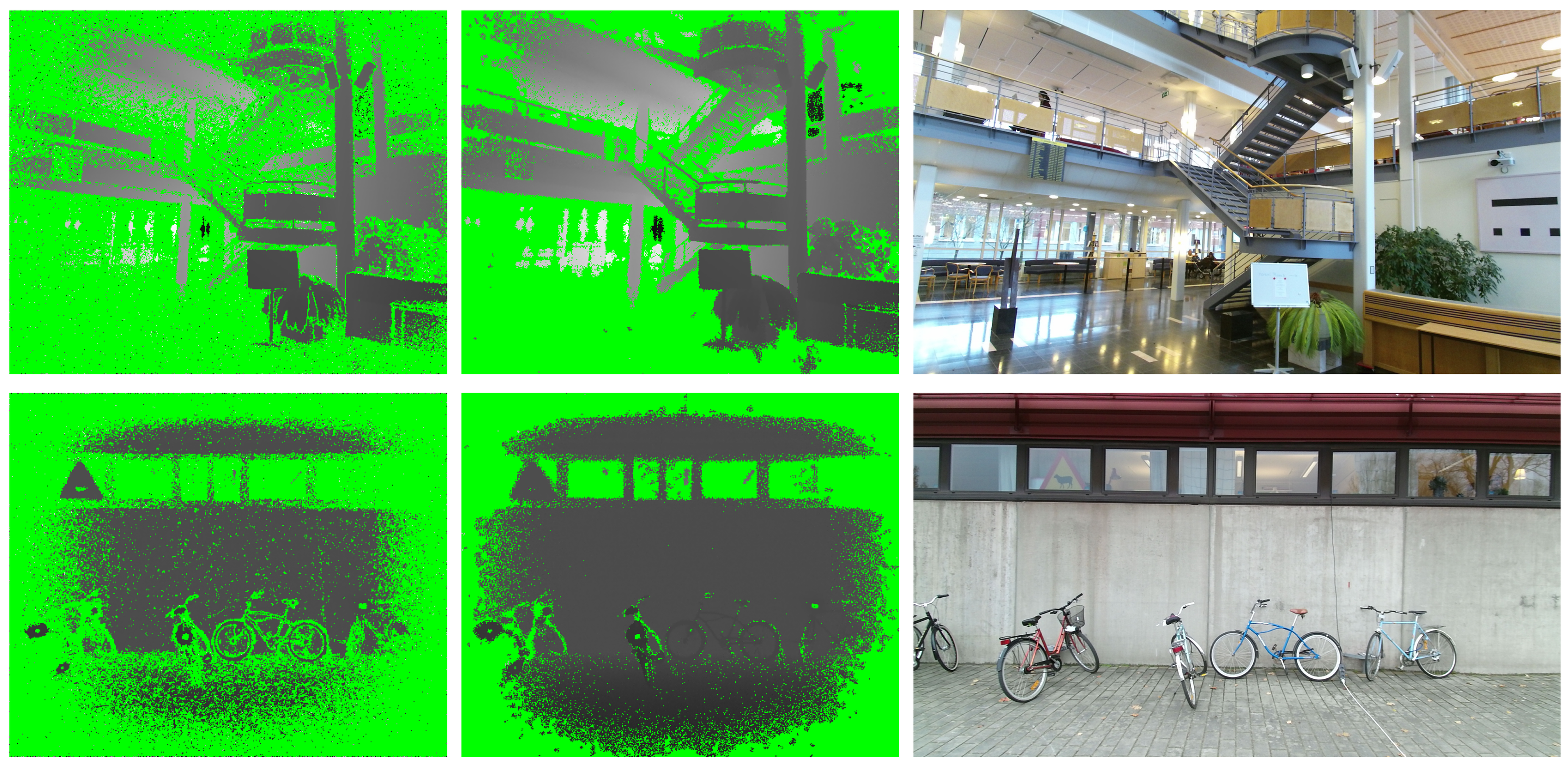

Figure 3: Single frame output. Left: *libfreenect2*, Center: proposed method. Right: corresponding RGB image. Pixels suppressed by outlier rejection are shown in green.

- Recordings of raw Kinect v2 measurements were unwrapped and passed to the Kinect fusion implementation KinFu in the *Point Cloud Library*.

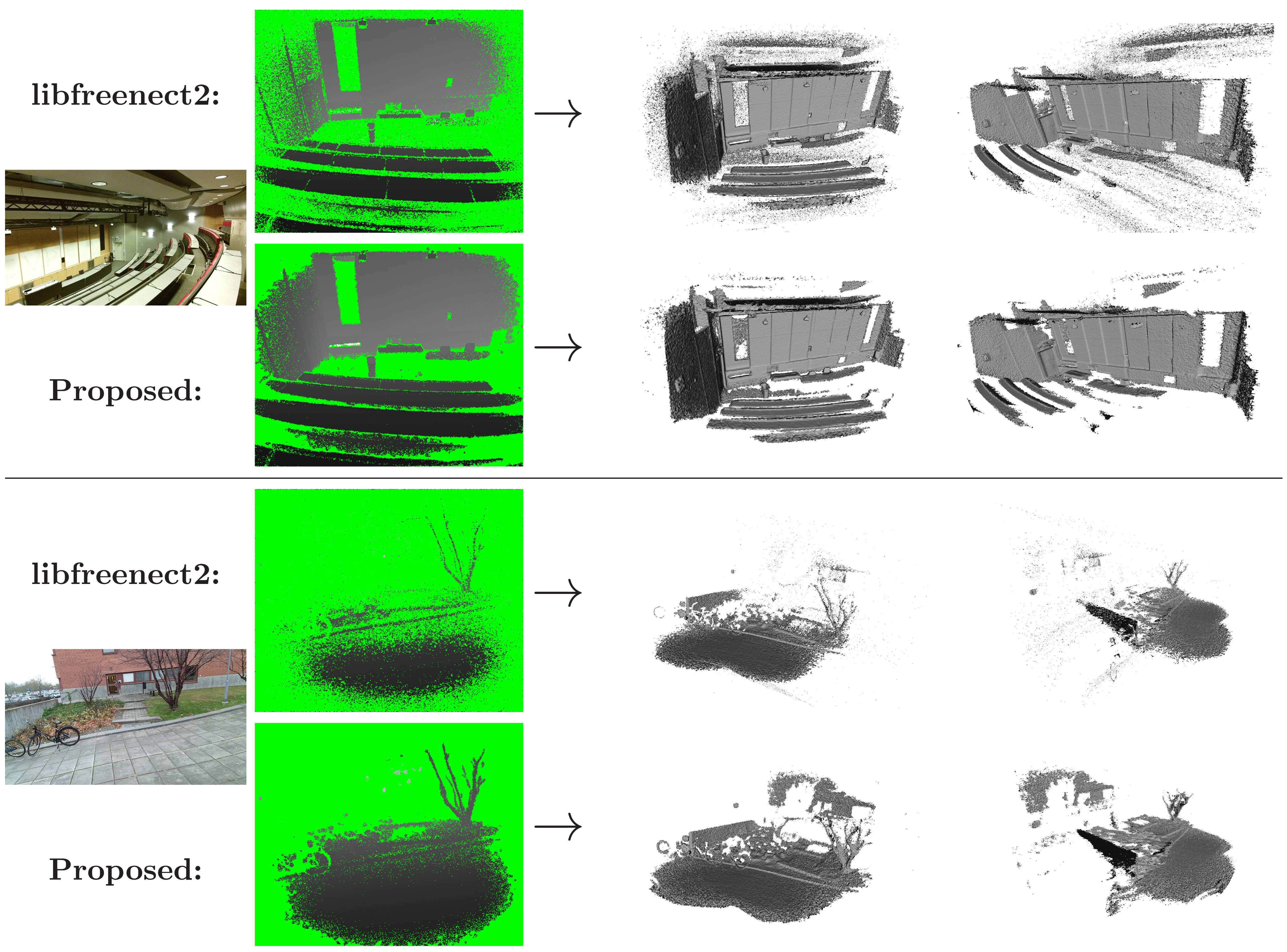

Figure 4: KinFu scans of two different scenes using depth images produced by *libfreenect2* and the proposed method. The input duration was 200 frames.