TrimBot2020 Deliverable 5.3

Dense Deformable Mapping Software

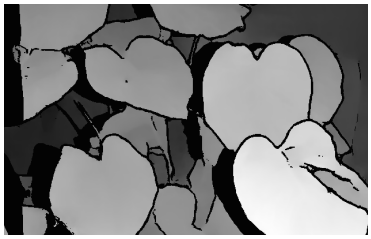
Principal Author: ALUF
Contributors: ALUF
Dissemination: PU

Abstract: We demonstrate how we can reconstruct a dynamic 3D point cloud from a sequence of stereo pairs. Based on the optical flow and disparity modules from Deliverable 5.2, points are triangulated in 3D space, each of them with an attached 3D motion vector. This enables tracking of relevant object parts like branches in 3D space over time. We demonstrate that points on an initial plant state are tracked under chaotic plant and camera motion, and points identified on a plant in motion can be back-projected onto the plant's resting state.

Deliverable due: Month 32

1 Background

left camera view



disparity

optical flow

Figure 1: **Raw components for point and surface tracking:** disparity and optical flow (provided by the scene flow software package) are used to reconstruct scenes in 3D, including plant and camera motion.

The TrimBot2020 project uses a stereo camera mounted on the arm to reconstruct the 3D shape of the plant to be trimmed, to indentify its shape and to capture deviations from the desired shape.

For visual servoing it is important that a point on the plant surface can be re-identified even if the point moves away from its initial position. Such motion can be induced by the motion of the camera, or by motion of the plant itself.

Therefore, we built a software that reconstructs a 3D point cloud together with the 3D motion vectors for each point of the point cloud from a sequence of stereo images. The reconstruction including its motion components yields a dynamical 3D model of the visible environment and can be used for visual servoing and closed-loop visual control. The software is built upon the dense scene flow software as documented in deliverable *D5.2 Scene Flow Software*. Fig. 1 shows the output of that software.

2 Software availability

The point tracking and evaluation software, as well as its dependencies, will be available at `https://github.com/lmb-freiburg/ROS-packages`.

3 Dense dynamical scene model

To avoid invalid measurements, left-right and forward-backward consistency checks are employed in the disparity and optical flow components of the scene flow module. Fig. 1 shows the effect of the left-right consistency check on the disparity map: occlusions, object boundary regrions, and overly hard to match (e.g. featureless) areas are removed. The resulting pointclouds are less dense, but cleaner and more consistent over time than when omitting the consistency checks.

Using intrinsically and extrinsically calibrated stereo camera, the disparity maps produced by DispNet are unprojected into 3D. The optical flow maps from FlowNet2 then provide temporal information about the motion of the 3D point in a plane parallel to the camera sensor. The missing depth-component of motion is retrieved from another disparity map for the second

frame. Together, full 3D motion is thus reconstructed for all visible points (modulo consistency checks).

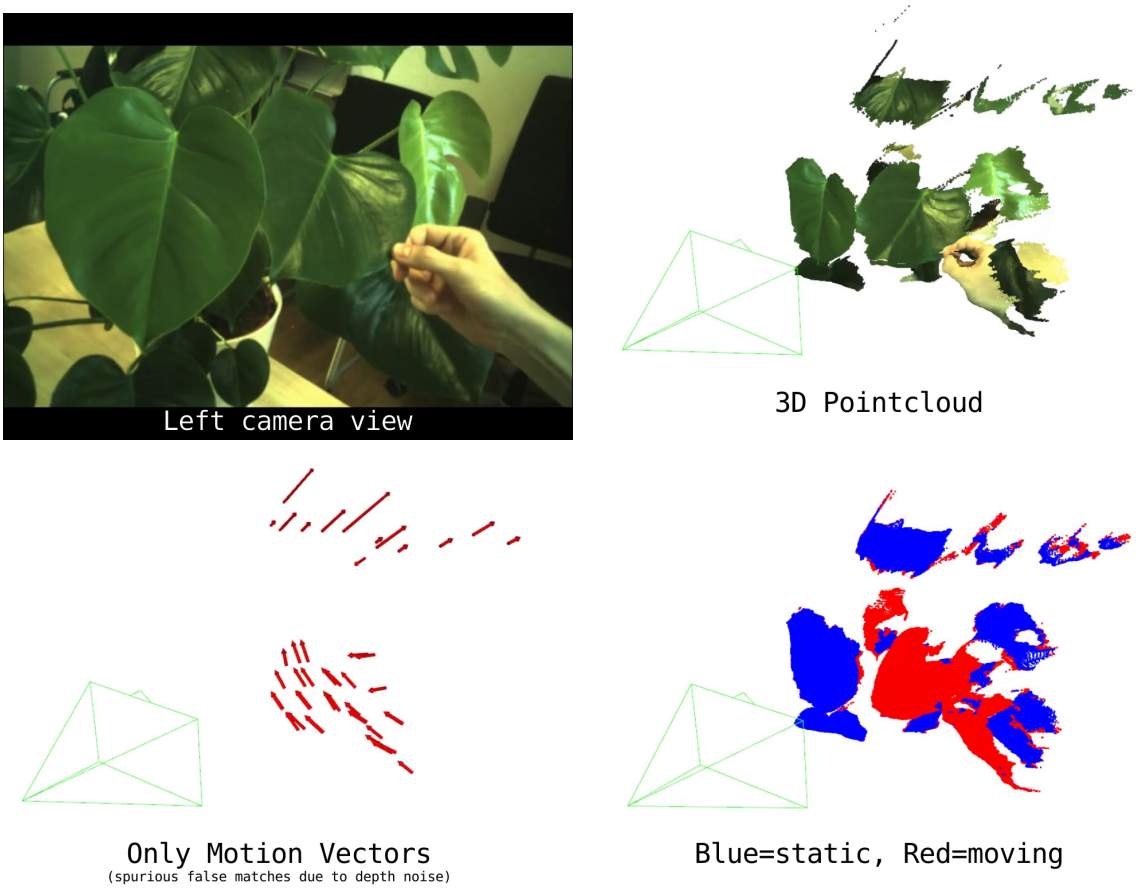


Figure 2: **Dynamic-surface tracking.**

4 Demonstration

4.1 Reconstruction consistency in dynamic scenes

Fig. 2 visualizes that points on a surface are densely tracked, and a simple motion magnitude threshold segments the scene into static and independently moving points.

This scene uses a static camera and shows deformation of a plant by means of physical contact. The video bundled with this document (`surface-tracking.mp4`) shows this scene in motion.

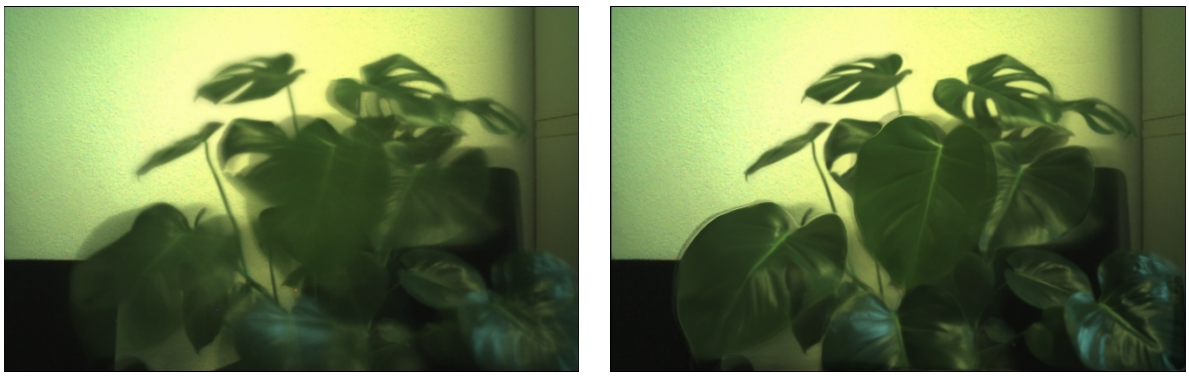
4.2 Robust point and surface tracking

Fig. 3 (bottom) shows that our recovered motion fields can be used to stabilize an observation sequence under significant irregular plant motion (i.e. the deformable surface is robustly reconstructed in its initial shape), by computing for each new frame the scene flow back to the initial

frame, and using the resulting motion vectors to backproject the new frame's pixels back into the initial frame.

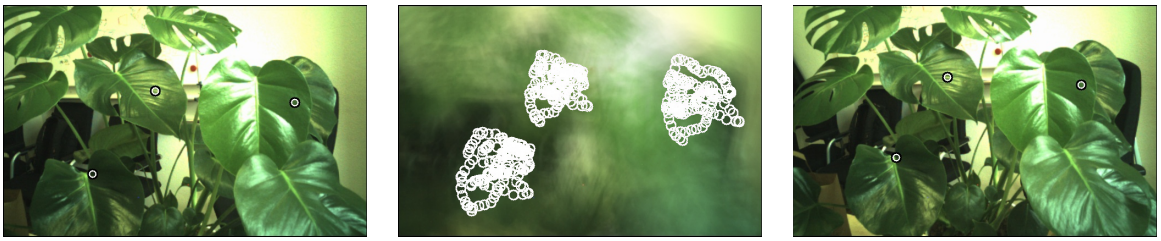
In Fig. 4, we track points frame-to-frame (via dense flow fields) instead of to an initial frame to show that tracking is stable and consistent over time.

As a dense frame-to-frame approach, this fails in case of degenerate inputs (e.g. severe motion blur) because tracking cannot be resumed once a point is lost (note that all tracked points are visible during the entire sequence).



mean image (raw input) mean image (stabilized input)

Figure 3: **Observation stabilization for surface recovery:** Despite significant plant motion, the backwarping-stabilized mean image shows that most of the surface can be robustly tracked.



initial points points during motion (overlay on mean image) final points

Figure 4: **Tracking under camera motion:** points on feature-bearing and feature-poor areas are tracked through a camera motion sequence.

4.3 Quantitative Evaluation

We evaluated tracking performance on a dataset in which the robot arm performs a scanning trajectory of a spherical bush.

Fig. 6 shows the distribution of spatial distances between initial reconstruction points and tracked points after the 130 frames shown in Fig. 5 (note that this only measures correspondences as they were tracked; there is no guarantee that the correspondences were correct).

Our tracking is based on direct scene flow estimation without point reidentification, so pixels can only be tracked as long as they stay in frame. This leads to only a small fraction of the initial point cloud still being tracked after 130 frames, as shown in Fig. 7. Still, those points that were tracked show stable positions.

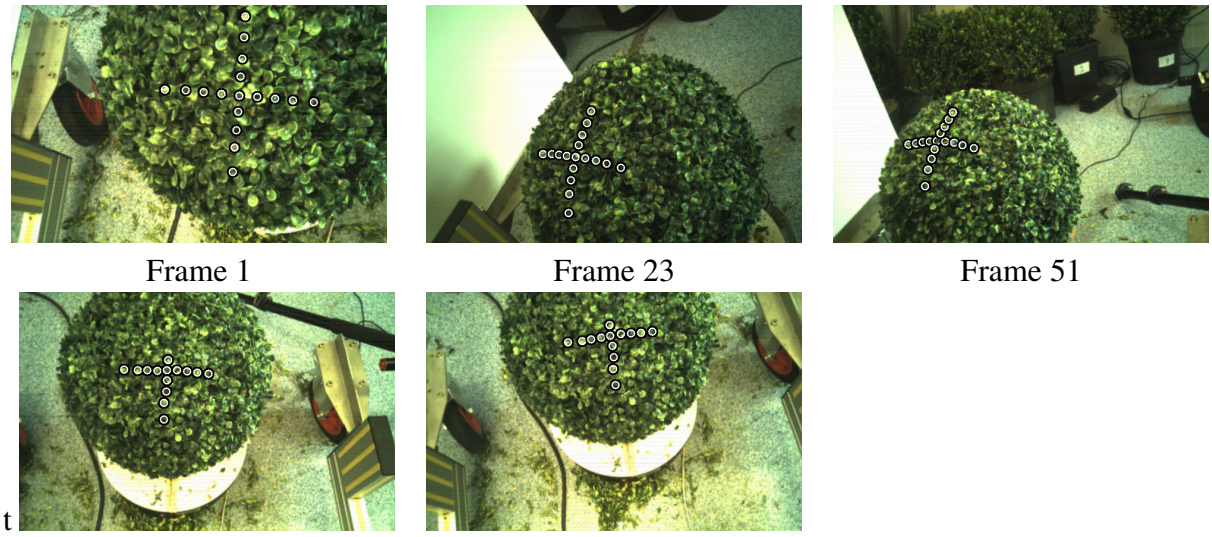


Figure 5: **Tracking sequence:** Snapshots from the dataset from which Fig. 6 and Fig. 7 were evaluated, with highlighted example points. Note how little of the initial frame stays visible throughout the sequence. One arm of the point cross is out of view between frames 51 and 112, and is subsequently lost from tracking.

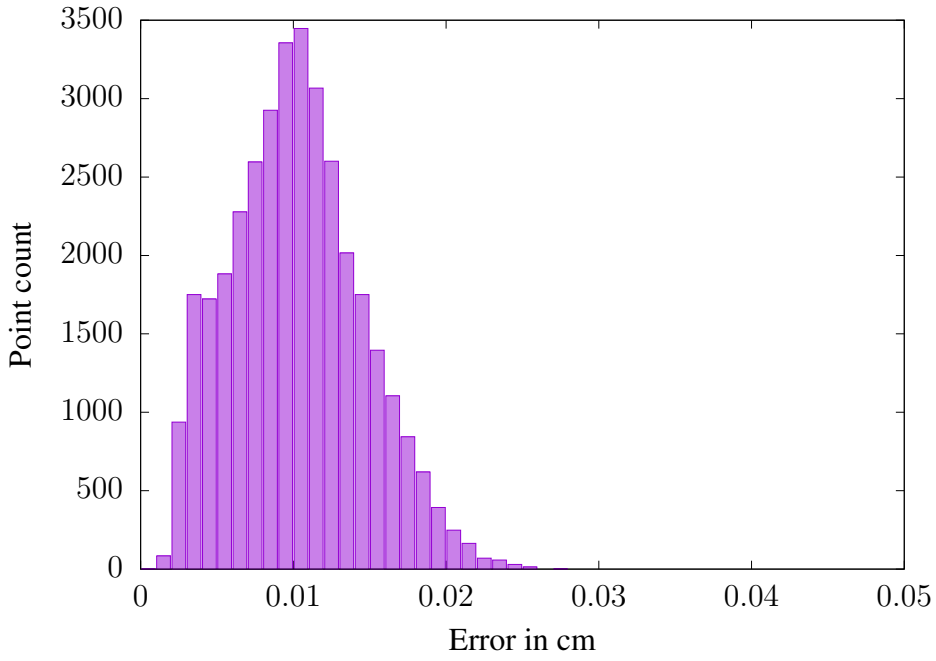
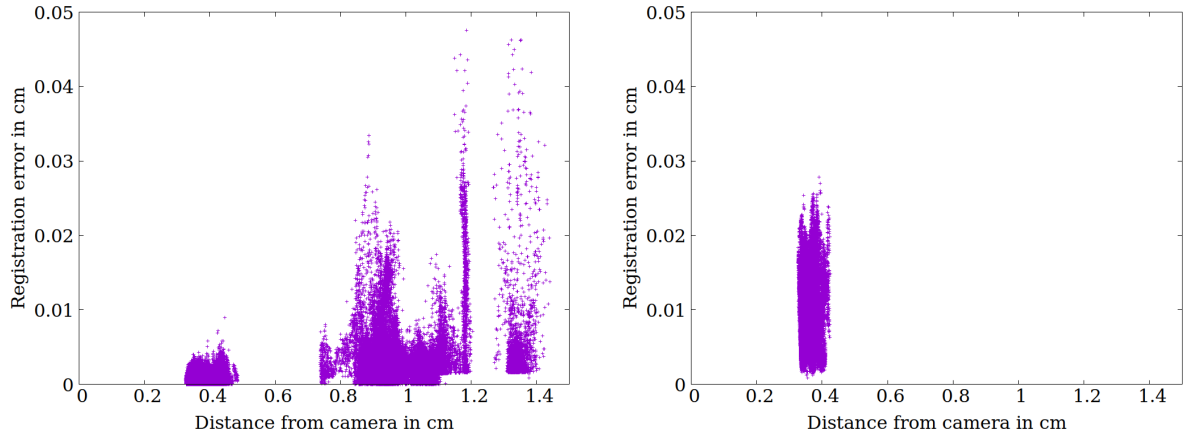


Figure 6: **Point tracking error:** Histogram of point registration errors after frame-to-frame tracking over 130 frames. The measured error is the metric distance between a reconstructed 3D point in the first frame, and its tracked correspondence after 130 frames. This measurement includes extrinsic and intrinsic camera calibration errors, as well as robot arm pose inaccuracies. **Almost all points which are still tracked after 130 frames are registered with an error of less than 2cm.** Only ca. 35k of initially ca. 180k points were tracked; the rest was lost due to occlusions, consistency checks in DispNet or FlowNet, or because they exited the camera view at some point. The number of points with error > 5cm was negligible.



(a) Frame 2 (b) Frame 130

Figure 7: **Tracking error correlation with distance to camera:** (a) Immediately after tracking starts, points further from the camera already exhibit large tracking errors; this can largely be attributed to depth noise in DispNet (which scales with depth). (b) After 130 frames, only points on the bush remain tracked (due to various reasons, see Fig. 6). These points show error accumulation, but an overall stable registration.

