Supplementary Material for A Multi-View Stereo Benchmark with High-Resolution Images and Multi-Camera Videos

Thomas Schöps[1] Johannes L. Schönberger[1] Silvano Galliani[2]
Torsten Sattler[1] Konrad Schindler[2] Marc Pollefeys[1,4] Andreas Geiger[1,3]
[1]Department of Computer Science, ETH Zürich
[2]Institute of Geodesy and Photogrammetry, ETH Zürich
[3]Autonomous Vision Group, MPI for Intelligent Systems, Tübingen [4]Microsoft, Redmond

In this supplementary document, we first show all initial datasets of our benchmark. We then give a description of the technical details of our image alignment technique. Finally, we show an experiment demonstrating the necessity of the newly proposed components of our image alignment technique.

1. Datasets

We show images of all 14 initial scenes in our benchmark in Fig. 1. They consist of 7 outdoor and 7 indoor scenes. Numerous scenes contain thin structures such as the trees in *forest*, or a rope in *playground*. Several scenes also contain large homogeneous or reflective surfaces such as the floor in *pipes* and *terrains*, the walls in *office*, or some doors in *electro*. These elements make the scenes challenging for stereo algorithms. All DSLR and multi-camera rig datasets were recorded within those scenes. Fig. 2 and Fig. 3 show example images of the high-resolution multi-view stereo datasets, while Fig. 4 shows the two-view stereo datasets. We plan to extend the benchmark with additional datasets over time.

2. Image Alignment - Technical Details

In this section, we give additional details on the image alignment process described in Section 3.2, "Refinement of Image Alignment", in the paper. We repeat some points from the paper where necessary to improve readability.

The input to the algorithm are the previously aligned colored point clouds, the images whose alignment shall be refined, and the initial values for the intrinsic and extrinsic parameters of all images. To robustify the optimization process, we leverage a multi-resolution scheme. We start from a low resolution for all images and then gradually increase it. In our implementation, we create as many image pyramid levels as is necessary for the smallest resolution to have an image area not larger than $200 * 160$ pixels. We half the image resolution in each downsampling step.

Before the optimization starts, a multi-resolution point cloud is created as described in Sec. 3.2 in the paper. Fig. 6 shows an example result for one of our datasets. Each level of the resulting multi-resolution point cloud has an associated point radius, applying to all points of this level.

As described in the paper, the optimization uses a cost function consisting of a sum of two terms: One of them rates the alignment of the images against the laser scan colors, while the other rates the alignment of the images against each other. The cost term can thus be stated as:

$$C(g_{\text{fixed}}) + C(g_{\text{variable}}) \ , \tag{1}$$

where g denotes point pair gradients and $C(g)$ is Eq. 2 from the paper:

$$C(g) = \sum_{\mathbf{p} \in \mathcal{P}} \sum_{i \in \mathcal{I}(\mathbf{p})} \rho \left[\sqrt{\sum_{j=1}^{5} \left(I_i(\pi_i(\mathbf{p}_j)) - I_i(\pi_i(\mathbf{p})) - g(\mathbf{p}, \mathbf{p}_j) \right)^2} \right] \ . \tag{2}$$

The values of $g_{\text{variable}}(\mathbf{p}, \mathbf{p}_j)$ are optimized, while $g_{\text{fixed}}(\mathbf{p}, \mathbf{p}_j)$ are pre-calculated at the beginning from the colors of the laser scan, denoted as $L(\mathbf{p})$, and remain fixed thereafter. For a point $\mathbf{p}$, the values of g_{fixed} are given by:

$$g_{\text{fixed}}(\mathbf{p}, \mathbf{p}_j) = L(\mathbf{p}_j) - L(\mathbf{p}) \quad . \tag{3}$$

In each iteration of the multi-resolution scheme, the current image pyramid level, but also all levels with smaller image resolution, are used in the cost function. The set $\mathcal{P}$ always includes all points from all levels of the multi-resolution point cloud. However, point observations are dropped if there is no suitable image pyramid level for them given the point radius, as described below.

For accessing image intensities at the projected position of points in the optimization process, we determine the scale at which a point is observed and use tri-linear interpolation: In addition to projecting a point's center to the image, the point's radius is also projected. It is used to compute the diameter d (in pixels on the highest-resolution pyramid level) of an approximate disc that would result from projecting a sphere with this radius at the point position. Then, the two pyramid levels h and $h+1$ whose pixel sizes are closest to the disc diameter are selected, *i.e.*, $2^h \leq d < 2^{h+1}$ (if no such pyramid levels are available at the current iteration of the multi-resolution scheme, the point observation is dropped). Each of these images is sampled at the projected point center position using bilinear interpolation. The results $I_h(\pi_h(\mathbf{p}))$ and $I_{h+1}(\pi_{h+1}(\mathbf{p}))$ are then linearly interpolated according to the point's scale as $(1-f) * I_h(\pi_h(\mathbf{p})) + f * I_{h+1}(\pi_{h+1}(\mathbf{p}))$, with $f = \frac{d-2^h}{2^{h+1}-2^h}$. Thus, in the first iteration of the multi-resolution scheme the *two* pyramid levels with the smallest image resolutions are used. An illustration of this process is shown in Fig. 5.

Note that the point radius as well as the image resolution differ by a factor of two between adjacent point cloud levels and between adjacent image pyramid levels, respectively. Thus, a point will be observed at an initial image pyramid level with the highest available image resolution suiting this point. If a corresponding point exists on the next point cloud level, its observation will also be on the next image pyramid level. Two images observing a point from a different distance will thus observe the same point neighborhoods from a minimum point cloud level on. Point cloud levels with a higher point radius are naturally weighted lower than levels with smaller radius since there are less points on these levels.

The optimization scheme proceeds as described in the following, and specified in Algorithm 1. For each step in the multi-resolution scheme, we run a number of optimization iterations until convergence or a maximum number of 300 iterations has been reached. We use the alternating optimization scheme proposed by [1]. In every iteration except the first, we first optimize all image intrinsics and poses using a Levenberg-Marquardt step, while keeping $g_{\text{variable}}(\mathbf{p}, \mathbf{p}_j)$ fixed. In a second step, we determine which of the points $\mathbf{p}$ are visible in which of the images, using the occlusion reasoning described in the paper. We then optimize the $g_{\text{variable}}(\mathbf{p}, \mathbf{p}_j)$ while keeping the image intrinsics and poses fixed. Afterwards, the new value of the cost function Eq. 1 is computed. If the cost did not improve over several iterations, the process converged and we continue to the next level of the multi-resolution image pyramid. Otherwise, we proceed with another iteration.

Algorithm 1 High-level specification of the optimization process.

```
for each step in the image multi-resolution scheme do
    for i := 0..max_iterations do
        if i > 0 then
            Optimize intrinsics and poses while fixing g_variable(p, p_j)
        Use intrinsics, poses and occlusion reasoning to determine observed points
        Optimize g_variable(p, p_j) while fixing intrinsics and poses
        Compute cost (Eq. 1)
        if convergence then
            break
```

3. Image Alignment Component Evaluation

Here, we complement the image alignment evaluations in the paper with an evaluation of two algorithmic components proposed in the paper:

- We verify the necessity of the gradient-based cost term by modeling constant brightness as in [1]. We therefore modify Eq. 2 to read:

$$C(c) = \sum_{\mathbf{p} \in \mathcal{P}} \sum_{i \in \mathcal{I}(\mathbf{p})} \rho \left[I_i(\pi_i(\mathbf{p})) - c(\mathbf{p}) \right] \quad . \tag{4}$$

- We verify the usefulness of the multi-resolution scheme by modifying our algorithm to operate directly on the highest resolution. In other words, we determine point neighbors within the original, full point cloud, and do not assign radii

to the points. All image accesses are performed with bilinear interpolation at the highest image resolution available in the multi-resolution scheme at a given point in time.

For simplicity, we use a single image of the multi-camera rig for this experiment, which is aligned against a laser scan. The results are shown in Fig. 7. We note that the variant using the brightness constancy assumption diverged completely due to the discrepancy of colors across modalities (*i.e.*, laser scan colors and image colors). Moreover, without the use of the multi-resolution scheme, the optimization gets stuck close to the initial state.

References

[1] Q. Zhou and V. Koltun. Color map optimization for 3d reconstruction with consumer depth cameras. In *SIGGRAPH*, 2014. 2

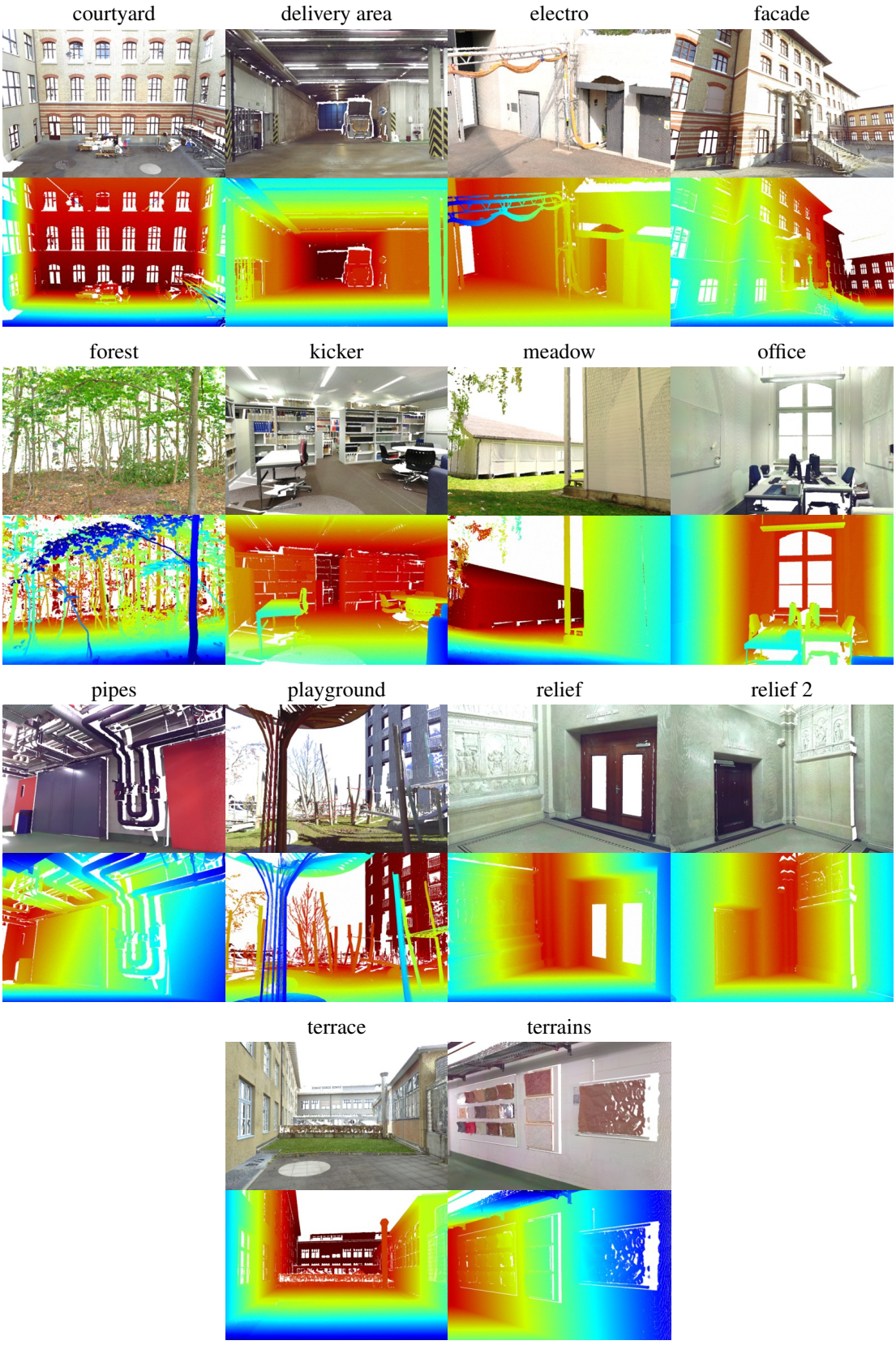

Figure 1: Colored 3D point cloud renderings of laser scans from our benchmark, with depth maps shown below.

Figure 2: Overview of the high-resolution multi-view stereo datasets, showing all images for each dataset (part 1).

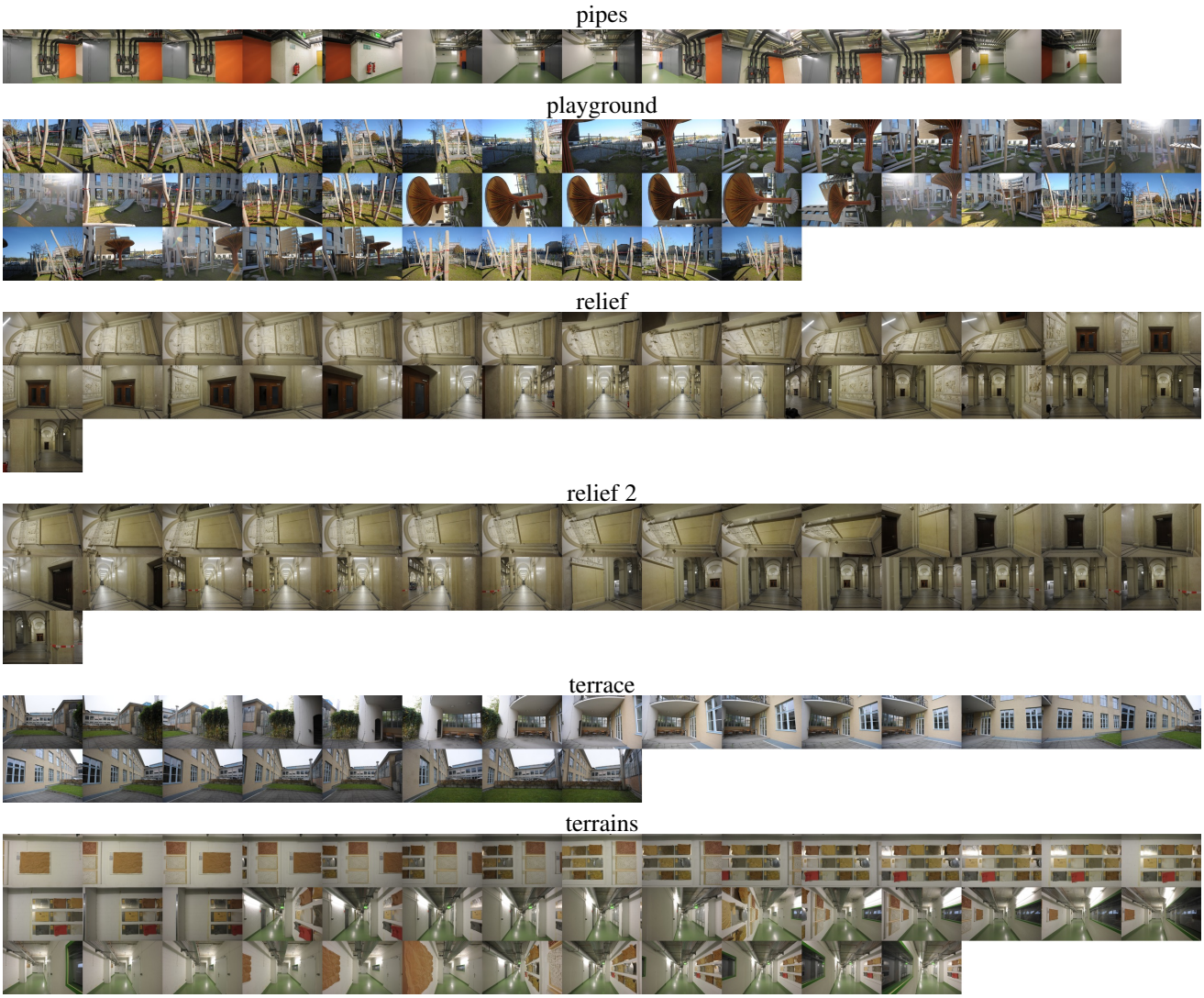

Figure 3: Overview of the high-resolution multi-view stereo datasets, showing all images for each dataset (part 2).

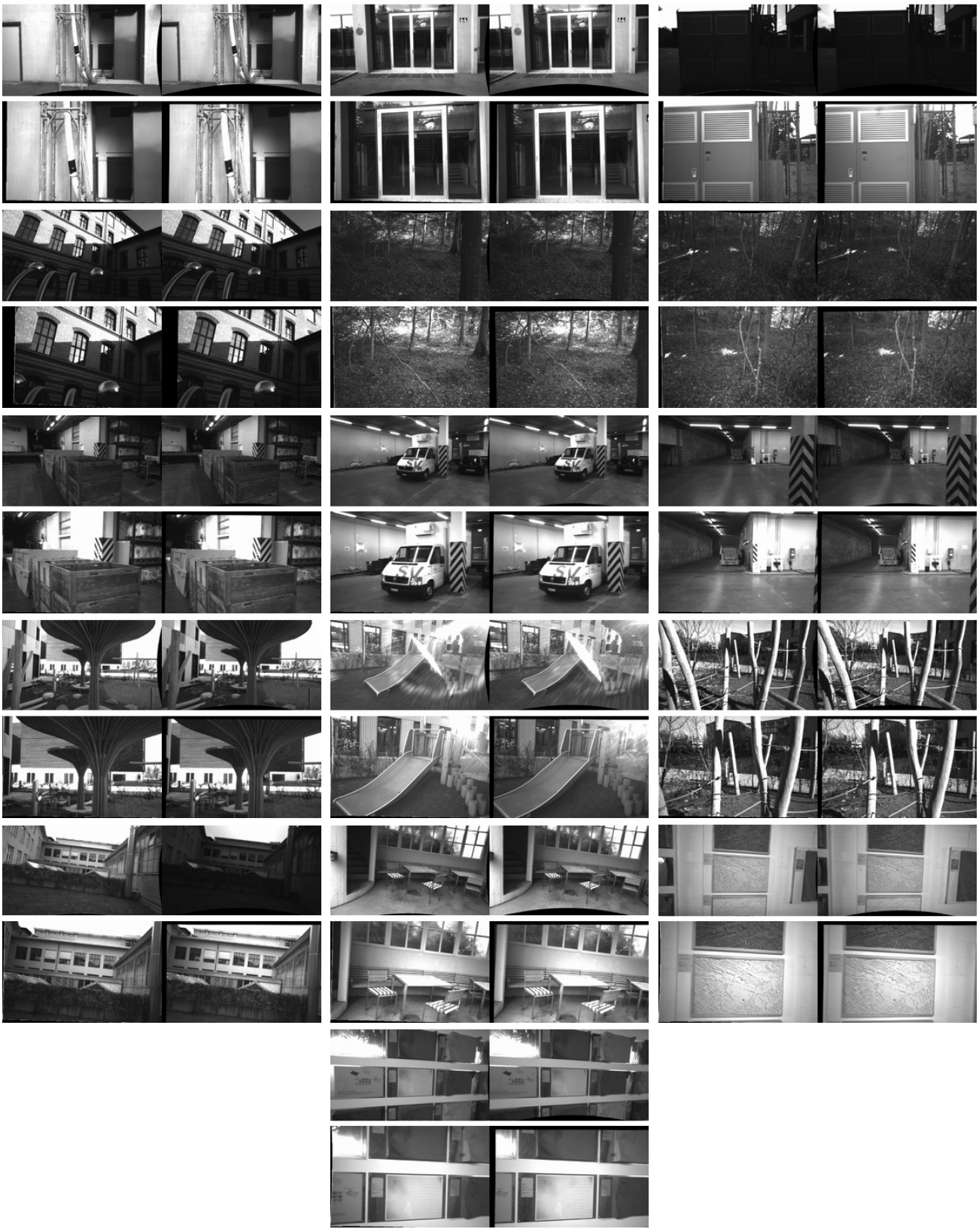

Figure 4: Low-resolution two-view stereo images, captured by two stereo pairs whose cameras have different fields-of-view. The images have been rectified.

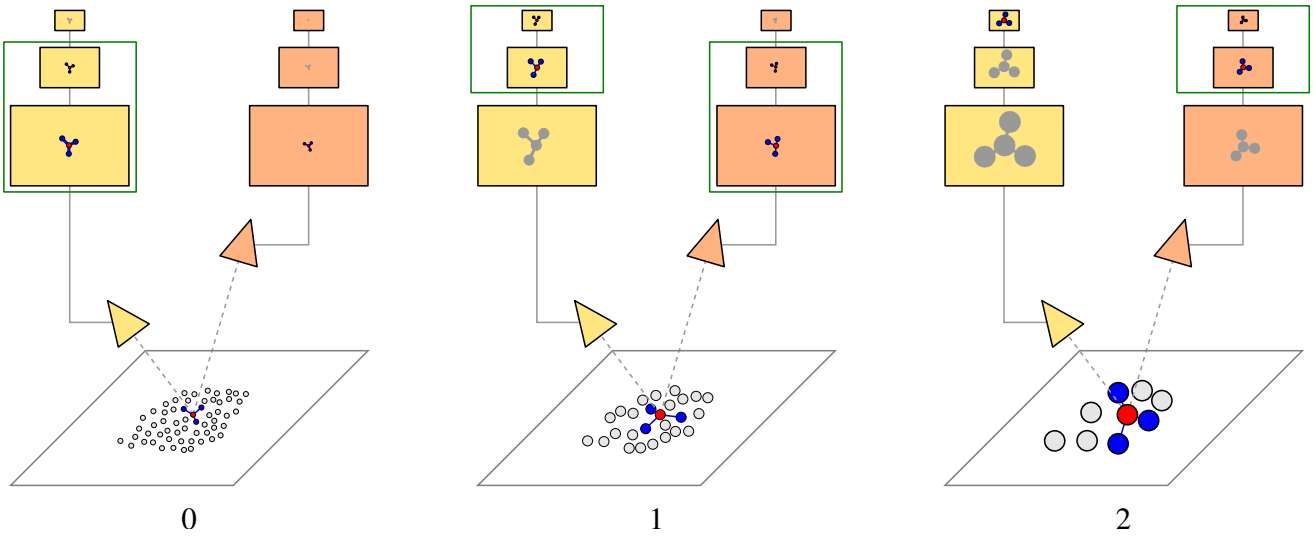

Figure 5: Sketch showing the concept of point observations in the multi-resolution point cloud for multiple point cloud levels, and their relation to image pyramid levels. Shown are 3 point cloud levels numbered 0 (left), 1 (middle), and 2 (right). Bottom: Points on a surface. The neighboring points of the red points are depicted in blue (using 3 neighbors). Two images taken by the same camera observe the red points, with the orange image having twice the distance to the point than the yellow image. Top: Image pyramids for the two images with 3 image pyramid levels. In the image pyramid levels, the red point and its neighbors are shown in color if the red point's observation scale is among the two closest ones to having a point diameter of 1 pixel, and in gray otherwise. If both of these colored scales are observed, sufficient data for tri-linear interpolation is present and the point is used for the optimization process. In these cases, the two involved image scales are framed green. In this example, the two images both observe the red point at cloud level 1 but at different image pyramid levels.

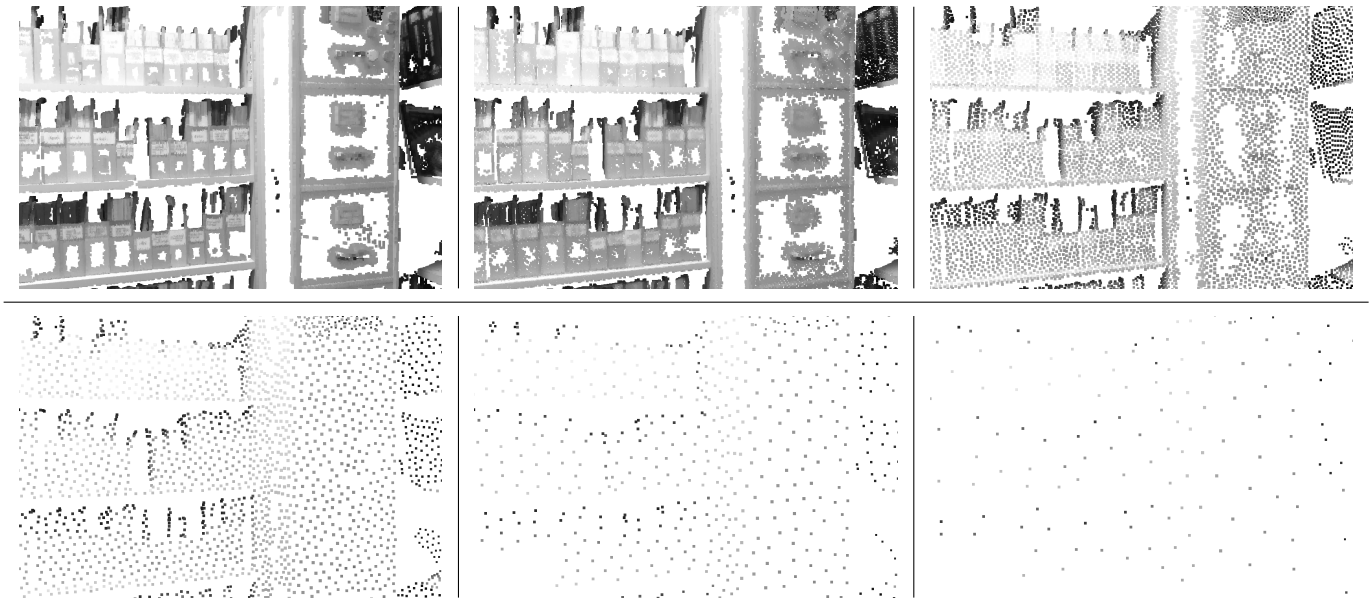

Figure 6: Example splat renderings of the multi-resolution point cloud point centers for the kicker dataset. From top left to bottom right: Point cloud levels 3 to 8.

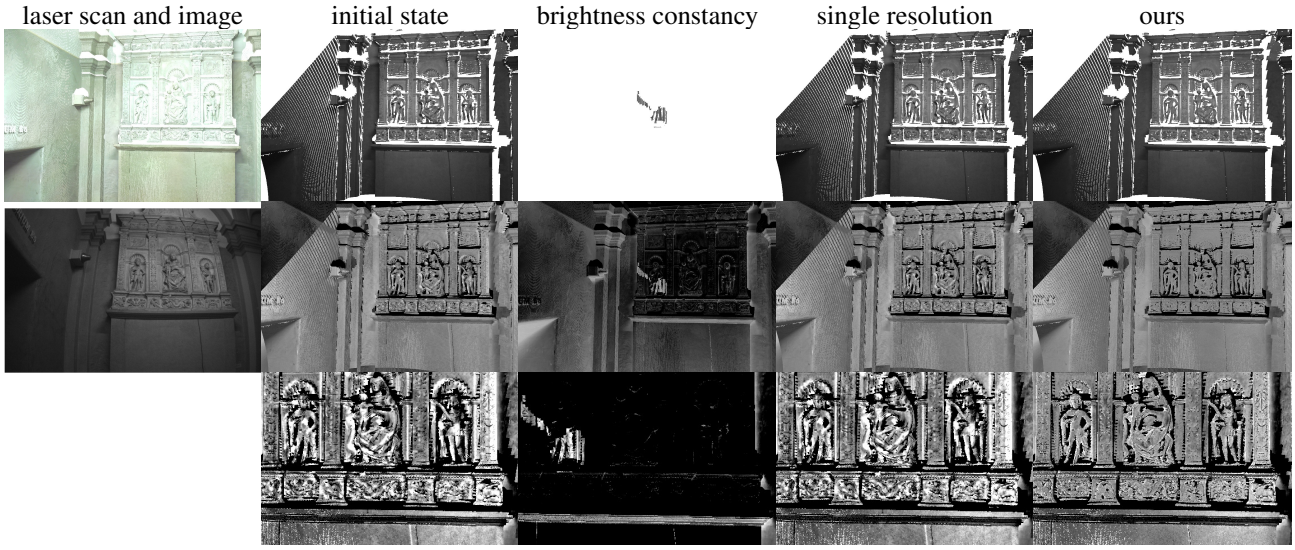

Figure 7: Comparison of different alignments for one image in the *relief 2* scene. The laser scan is shown on the top left with its original colors, while the image to be aligned is shown below it. To the right of them, the laser scan is shown in the top row with the image projected onto it in different alignments. The middle row shows the difference image to the top left image, computed on grayscale images. In the bottom row, a cutout from the middle row is displayed with increased contrast. From left to right: Original laser scan colors and image, initial state (set manually for this example), alignment assuming brightness constancy, alignment with a single-resolution point cloud only, alignment using our complete method. Since the colors of the laser scan and the image differ strongly, a darker difference image does not necessarily mean that the alignment is better. Instead, the difference image being homogeneous is an indication for a good alignment. Note that when assuming brightness constancy, the camera intrinsics drift off such that only a small patch remains visible in the image.