Supplementary Material

The following Sections constitute the Supplementary Material. Section A provides a pseudocode of our method. In Sec. B, a detailed profiling of our run time is presented. Section C investigates the effect of different hyperparameters for our outlier reweighting. Section D studies the impact of the convergence parameters. Section E provides visualizations of our continuous components on the DiLiGenT dataset. In Sec. F we analyze the impact of our merging operation on reconstruction error on the DiLiGenT dataset. Section G provides an ablation on the pixel connectivity used by our method. Finally, in Sec. H we discuss the limitations of our method.

A. Pseudocode of our method

A pseudocode for our method is shown in Algorithm 1.

Algorithm 1 Pseudocode of our method.

```
Require: $\theta_c$, can merge (bool), $\text{freq}_{\text{merging}}$, $\Delta E_{\max}$, $T$.
Initialize $\tilde{\mathbf{z}} \leftarrow \mathbf{0}$
Form components $\{\mathcal{C}_c^{(0)}\}$ based on $\theta_{a,b} < \theta_c$
Compute intra-component matrices
  $\mathbf{A_c}, \mathbf{b_c}, \mathbf{W_c} = \text{diag}\left(\{W_{b\to a}\}_{(a,b)\in E(\mathcal{C}_c^{(0)})}\right)$ (16)
Fill each component $\mathcal{C}_c$ in parallel:
  $\tilde{\mathbf{z}}_{\mathbf{c}}^{(0)} \leftarrow \texttt{cg}\left(\mathbf{A_c}^\top \mathbf{W_c} \mathbf{A_c}, \mathbf{A_c}^\top \mathbf{W_c} \mathbf{b_c}\right)$
converged $\leftarrow$ False, $m \leftarrow 0$, $t \leftarrow 0$, $E_0 \leftarrow \epsilon$
Form inter-component matrices $\overline{\mathbf{A}}_0$, $\overline{\mathbf{b}}_0$ (18)
while not converged do                         ▷ Relative-scale optimization
    if $t \leq 1$ then                          ▷ Alignment optimization
        Uniform weights: $\overline{\mathbf{W}}_m^{(t)} \leftarrow \text{diag}(1)$
    else                                        ▷ Discontinuity-aware optimization
        BiNI weights with outlier reweighting (20):
          $\overline{\mathbf{W}}_m^{(t)} \leftarrow \text{diag}\left(\{W_{b\to a} \cdot W_{b\to a}^{\text{out}}\}_{(a,b)\in E_{\text{inter}}^{(m)}}\right)$
    end if
    $\tilde{\mathbf{s}}^{(t)} \leftarrow \texttt{cg}\left(\overline{\mathbf{A}}_m^\top \overline{\mathbf{W}}_m^{(t)} \overline{\mathbf{A}}_m, \overline{\mathbf{A}}_m^\top \overline{\mathbf{W}}_m^{(t)} \overline{\mathbf{b}}_m\right)$
    Scale components (in parallel, via broadcasting):
      $\forall c \in \{0, \dots, |C^{(m)}| - 1\}$, $\tilde{\mathbf{z}}_{\mathbf{c}}^{(t+1)} \leftarrow \tilde{\mathbf{z}}_{\mathbf{c}}^{(t)} + \tilde{s}_c^{(t)} \mathbf{1}$
    $t \leftarrow t + 1$
    if can merge $\wedge$ $\left(t \equiv 0 \pmod{\text{freq}_{\text{merging}}}\right)$ then
        $\forall \mathcal{C}_c^{(m)}$, $(\hat{a}, \hat{b})_c^{(m)} \leftarrow \arg\min_{(a,b)\in\partial\mathcal{C}_c^{(m)}} |\overline{\chi}_{b\to a}|$
        Compute subgraph $\hat{\mathcal{Q}}_m \leftarrow (C^{(m)}, \{(\hat{a}, \hat{b})_c^{(m)}\})$
        $\{\mathcal{C}_c^{(m+1)}\} \leftarrow$ connected components$(\hat{\mathcal{Q}}_m)$
        $m \leftarrow m + 1$
        Form inter-component matrices $\overline{\mathbf{A}}_m$, $\overline{\mathbf{b}}_m$ (18)
    end if
    $E_t \leftarrow (\overline{\mathbf{A}}_m \tilde{\mathbf{s}}^{(t)} - \overline{\mathbf{b}}_m)^\top \overline{\mathbf{W}}_m^{(t)} (\overline{\mathbf{A}}_m \tilde{\mathbf{s}}^{(t)} - \overline{\mathbf{b}}_m)$
    converged $\leftarrow \frac{|E_t - E_{t-1}|}{E_{t-1}} < \Delta E_{\max} \vee t = T$
end while
return $\tilde{\mathbf{z}}$
```

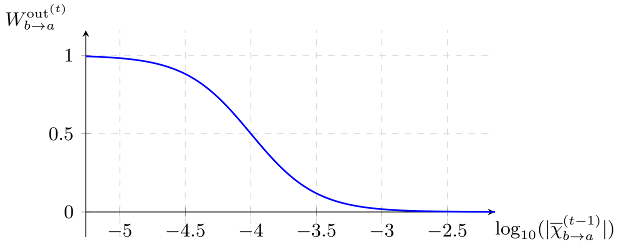


Figure 5. **Outlier reweighting** (20) **as a function of** $\log_{10}(|\overline{\chi}_{b\to a}^{(t-1)}|)$**, for** $L = 10^{-3}$ and $U = 10^{-5}$.

B. Detailed profiling of our run time

We provide a detailed profiling of the run time of our method for the parameter configuration used in our main experiments, both on the DiLiGenT dataset (Tab. 2) and on the large-scale normals maps of Fig. 4 (Tab. 3).

It is worth noting that, in both cases, two operations that account for a significant fraction of the total execution time are the two pre-processing steps of forming the intra-component matrices $\mathbf{A_c}$, $\mathbf{b_c}$, and $\mathbf{W_c}$ and filling the components. For the latter, we use the Python `joblib` library to parallelly execute multiple instances of per-component conjugate-gradient optimization. While this usually converges in few fractions of a second due to the smaller scale of the per-component optimization compared to the global optimization, larger resolutions might produce larger components, resulting in increased time for this initial step. Additionally, parallelization is capped by the number of processes that are available to the program (we set this to 4 in our experiments), thereby still requiring iterative processing. On the other hand, the formation step of the intra-component matrices $\mathbf{A_c}$, $\mathbf{b_c}$, and $\mathbf{W_c}$ is non-optimized in our current implementation, and alone contributes to a factor of up to respectively 39% (`coinskeyboard`) and 44% (`cow`) of the total execution time for the large-scale normal maps and the smaller-scale DiLiGenT dataset. The reason for the long time required to perform this step lies in the fact that the matrices $\mathbf{A_c}$ and $\mathbf{W_c}$ are represented in our implementation as sparse matrices (`scipy.sparse.csr_matrix`) of different shape, and as such cannot benefit from parallel-access optimized broadcasting operations. Implementation improvements in this direction are left to future work.

C. Ablation on outlier reweighting

To complement the definition provided in the main paper, we provide an illustration of our outlier reweighting function in Fig. 5, using the parameters from the main experiments. Additionally, we ablate on different values for its hyperparameter L and also include a variant of the reweighting which introduces a *hard* outlier thresholding, so that equations with residual magnitude $|\overline{\chi}_{b\to a}^{(t-1)}|$ are assigned weight

	bear	buddha	cat	cow	harvest	pot1	pot2	reading	goblet
Formation of $\mathcal{G}_0$	0.020	0.018	0.019	0.020	0.019	0.018	0.019	0.019	0.020
Computation of $\{\mathcal{C}_c^{(0)}\}$	0.092	0.090	0.091	0.087	0.091	0.097	0.089	0.088	0.094
Formation of $\mathbf{A_c}$, $\mathbf{b_c}$, $\mathbf{W_c}$	0.546	1.857	0.731	0.758	3.345	1.581	1.384	1.206	0.344
Filling of components	1.168	2.442	1.428	1.360	4.166	2.325	2.103	1.719	0.745
Iteration 0	1.174	2.587	1.439	1.368	4.311	2.342	2.118	1.734	0.775
Iteration 1	1.183	2.977	1.464	1.378	4.810	2.394	2.148	1.766	0.840
Iteration 2	1.193	3.490	1.484	1.391	5.747	2.471	2.187	1.822	0.858
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Convergence	1.270 [it. 8]	8.033 [it. 11]	1.500 [it. 3]	1.525 [it. 12]	18.805 [it. 17]	3.511 [it. 18]	2.655 [it. 17]	2.679 [it. 20]	1.398 [it. 49]

Table 2. **Profiling of our method on the the DiLiGenT benchmark [22].** Intermediate execution times from the start, after the completion of each step are reported [s]. $\theta_c = 3.5°$, $\Delta E_{\max} = 10^{-3}$, $T = 150$, $8-$connectivity, and outlier reweighting are used, without merging.

	bedroom	livingroom	coinskeyboard	schooldesk	seafloor	cake
Formation of $\mathcal{G}_0$	0.019	0.058	0.058	0.062	0.050	0.020
Computation of $\{\mathcal{C}_c^{(0)}\}$	0.138	0.522	0.375	0.447	0.395	0.147
Formation of $\mathbf{A_c}$, $\mathbf{b_c}$, $\mathbf{W_c}$	1.554	12.226	42.058	5.759	15.663	4.486
Filling of components	3.804	50.536	51.855	69.922	29.548	8.240
Iteration 0	3.955	51.232	54.343	70.075	29.919	8.382
Iteration 1	4.290	55.196	60.506	70.448	31.546	8.890
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Merging $m = 0$	5.876	92.018	96.427	75.205	51.690	12.382
Merging $m = 1$	6.198	102.327	102.591	75.783	57.139	12.883
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Convergence	6.796	105.081	104.630	76.648	58.524	13.374

Table 3. **Profiling of our method on the the large-scale normal maps from Fig. 4.** Columns 1 to 6 in Fig. 4 are referred to, from left to right, as: bedroom, livingroom, coinskeyboard, schooldesk, seafloor, cake. Intermediate execution times from the start, after the completion of each step are reported [s]. $\theta_c = 2.0°$, $\Delta E_{\max} = 10^{-3}$, $T = 15$, $8-$connectivity, $\text{freq}_{\text{merging}} = 5$, and outlier reweighting are used.

Reweighting type	U	θ_c	bear	buddha	cat	cow	harvest	pot1	pot2	reading	goblet
Soft	10^{-5}	None	0.02	0.13	0.03	0.09	1.35	0.40	0.14	0.19	9.37
		2.0°	0.02	0.17	0.03	0.09	1.02	0.37	0.14	0.20	9.35
		3.5°	0.02	0.10	0.04	0.10	1.10	0.39	0.14	0.10	8.08
		5.0°	0.02	0.15	0.51	0.39	1.61	0.55	0.14	0.17	4.45
Soft	10^{-4}	None	0.02	0.20	0.03	0.09	1.31	0.36	0.13	0.17	9.41
		2.0°	0.02	0.17	0.03	0.09	1.04	0.36	0.14	0.10	9.37
		3.5°	0.02	0.11	0.04	0.09	1.07	0.38	0.14	0.09	9.49
		5.0°	0.02	0.15	0.51	0.39	1.75	0.55	0.13	0.16	9.62
Soft	10^{-6}	None	0.02	0.17	0.03	0.09	0.99	0.36	0.13	0.13	9.41
		2.0°	0.02	0.13	0.03	0.09	0.88	0.36	0.14	0.10	9.40
		3.5°	0.02	0.17	0.04	0.09	1.18	0.38	0.13	0.10	7.53
		5.0°	0.02	0.18	0.51	0.39	1.70	0.55	0.13	0.15	9.62
Hard	N/A	None	0.03	0.27	0.05	0.09	1.78	0.41	0.13	0.19	9.33
		2.0°	0.02	0.42	0.05	0.09	1.13	0.39	0.14	0.22	8.19
		3.5°	0.02	0.36	0.05	0.10	1.09	0.40	0.14	0.19	3.54
		5.0°	0.02	0.16	0.51	0.39	1.42	0.38	0.14	0.24	4.42

Table 4. **Ablation on the outlier reweighting mechanism on the DiLiGenT benchmark [22].** The mean absolute depth error (MADE) $[\text{mm}]$ of our method is reported. The upper outlier reweighting threshold U is set to 10^{-3}, and soft threshold with different lower thresholds L as well as hard thresholding based on U are compared. All experiments use convergence criteria $\Delta E_{\max} = 10^{-3}$ and $T = 150$, $8-$pixel connectivity, without merging.

$W_{b\to a}^{\text{out}^{(t)}} = 0$ if $|\overline{\chi}_{b\to a}^{(t-1)}| \geq U$ and weight $W_{b\to a}^{\text{out}^{(t)}} = 1$ if $|\overline{\chi}_{b\to a}^{(t-1)}| < U$, where we set $U = 10^{-3}$.

Table 4 reports the results of the ablation. Overall, the differences across the variants for outlier reweighting are marginal for most objects. However, for objects with larger discontinuities (buddha, harvest, reading) soft reweighting achieves generally better accuracy, with a small degree of object specificity for the value of L, indicating that it might be effective particularly in controlling outlier residuals that arise due to discontinuities.

$\Delta E_{\max}$	T	θ_c	bear		buddha		cat		cow		harvest		pot1		pot2		reading		goblet*	
			Err	t	Err	t	Err	t	Err	t	Err	t	Err	t	Err	t	Err	t	Err	t
10^{-3}	5	None	0.02	4.08	0.51	10.56	0.03	5.17	0.09	2.24	1.77	17.40	0.39	9.49	0.13	4.23	0.17	4.74	9.42	5.54
		2.0°	0.02	2.49	0.18	7.92	0.03	2.91	0.09	2.37	1.24	11.65	0.38	5.26	0.13	4.05	0.14	4.75	9.37	1.27
		3.5°	0.02	1.22	0.15	4.89	0.04	1.51	0.10	1.45	1.10	7.55	0.41	2.63	0.13	2.26	0.13	1.92	9.47	0.87
		5.0°	0.02	0.86	0.18	2.65	0.51	1.11	0.39	0.98	1.78	5.30	0.56	1.96	0.13	1.67	0.18	1.33	9.56	0.75
10^{-3}	15	None	0.02	6.09	0.20	17.69	0.03	9.17	0.09	3.92	1.31	46.33	0.36	22.12	0.13	9.64	0.17	4.31	9.41	5.15
		2.0°	0.02	2.55	0.17	17.08	0.03	3.11	0.09	2.50	1.05	28.62	0.36	6.27	0.14	5.40	0.10	7.01	9.37	1.58
		3.5°	0.02	1.28	0.11	7.62	0.04	1.52	0.09	1.54	1.06	16.82	0.38	3.27	0.14	2.61	0.09	2.43	9.46	0.99
		5.0°	0.02	0.89	0.15	2.75	0.51	1.22	0.39	1.04	1.75	7.87	0.55	2.26	0.13	1.93	0.16	1.49	9.62	0.86
10^{-3}	150	None	0.02	7.39	0.20	19.56	0.03	36.98	0.09	3.35	1.31	44.35	0.36	32.51	0.13	9.83	0.17	3.40	9.41	4.78
		2.0°	0.02	2.55	0.17	19.07	0.03	3.11	0.09	2.54	1.04	28.33	0.36	6.26	0.14	5.59	0.10	6.54	9.37	2.33
		3.5°	0.02	1.27	0.11	8.03	0.04	1.50	0.09	1.53	1.07	18.81	0.38	3.51	0.14	2.66	0.09	2.68	9.49	1.40
		5.0°	0.02	0.88	0.15	2.76	0.51	1.24	0.39	1.04	1.75	7.29	0.55	5.80	0.13	1.92	0.16	1.49	9.62	0.84
10^{-5}	5	None	0.02	3.43	0.51	7.43	0.03	4.85	0.09	2.54	1.77	17.24	0.39	8.65	0.13	2.93	0.17	4.29	9.42	5.52
		2.0°	0.02	2.46	0.18	7.55	0.03	2.94	0.09	2.31	1.24	12.28	0.38	5.18	0.13	3.97	0.14	4.78	9.37	1.28
		3.5°	0.02	1.24	0.15	4.93	0.04	1.52	0.10	1.42	1.10	7.98	0.41	2.61	0.13	2.29	0.13	1.95	9.47	0.87
		5.0°	0.02	0.85	0.18	2.65	0.51	1.12	0.39	0.98	1.78	5.18	0.56	1.97	0.13	1.67	0.18	1.35	9.56	0.81
10^{-5}	15	None	0.01	10.17	0.15	25.28	0.03	8.78	0.09	4.99	1.28	53.51	0.36	25.19	0.13	7.16	0.09	14.50	9.40	12.20
		2.0°	0.02	2.93	0.17	16.94	0.03	3.49	0.09	2.64	1.05	29.43	0.36	8.36	0.14	5.46	0.09	7.93	9.37	1.59
		3.5°	0.02	1.38	0.11	8.88	0.04	1.69	0.09	1.56	1.06	15.89	0.38	3.25	0.14	2.62	0.09	2.46	9.46	1.01
		5.0°	0.02	0.95	0.12	3.90	0.51	1.21	0.39	1.05	1.69	9.50	0.55	2.32	0.13	1.93	0.15	1.62	9.62	0.84
10^{-5}	150	None	0.01	11.20	0.19	253.51	0.03	42.47	0.09	12.51	1.32	387.01	0.36	169.22	0.13	80.35	0.10	66.33	9.41	66.55
		2.0°	0.02	3.04	0.17	137.36	0.03	4.97	0.09	2.86	1.03	336.65	0.36	78.28	0.14	12.60	0.10	73.13	9.37	6.63
		3.5°	0.02	1.60	0.11	52.42	0.04	2.88	0.09	1.93	1.18	98.90	0.38	11.63	0.14	4.24	0.09	10.85	9.49	3.32
		5.0°	0.02	1.10	0.12	14.57	0.51	2.08	0.39	1.36	1.69	121.54	0.55	7.42	0.13	2.64	0.16	5.83	9.62	2.44

Table 5. **Ablation on the convergence criteria $\Delta E_{\max}$ and T on the DiLiGenT benchmark [22].** The mean absolute depth error (MADE, abbreviated as Err) [mm] and the total execution time (abbreviated as t) [s] of our method are reported. All experiments use outlier reweighting $W_{b\to a}^{\text{out}}$ (20) with $L = 10^{-5}$ and $U = 10^{-3}$, without merging.

D. Ablation on the convergence parameters

Table 5 reports the reconstruction error and run time achieved by our method for different values of the parameters $\Delta E_{\max}$ and T that control the termination of its execution. We observe that with limited exceptions, mostly restricted to our pixel-level variant ($\theta_c = \text{None}$), enforcing a stricter relative-energy convergence criterion ($\Delta E_{\max} = 10^{-5}$) produces no significant changes in the accuracy of the reconstruction, while requiring a longer run time. Notably, the same conclusion applies to the maximum number of optimization steps, for which we find that our method effectively achieves convergence in 5 or at most 15 iterations for all objects. While earlier termination of the optimization results in a slight increase in the running time, this speedup is not particularly significant for the small-scale of the normal maps from DiLiGenT, especially in the case of component-based (rather than pixel-level) optimization. For our main experiments on the DiLiGenT benchmark (*cf*. main paper), we therefore chose to use $T = 150$, to enable convergence for the baselines without outlier reweighting.

E. Ablation on the threshold for component formation

Figure 6 shows the continuous components identified by our heuristic on the DiLiGenT benchmark, for the different values of the normal similarity threshold θ_c that we use in our experiments.

We note that for each decomposition it is possible to compute the minimum theoretical reconstruction error that could be achieved by the relative scale optimization, which coincides with the global mean average depth error (MADE) that would be attained if the optimal combination of scales was applied to the individual reconstructions formed in the component filling stage. In general, each of such per-component reconstructions introduces an error, for however small, in its corresponding surface region. This is due to the approximations inherent in the models of continuity and discontinuity, as well as to the convergence of the optimization. To compute the minimum MADE, it is sufficient to compute the sum of the per-component MADEs, weighted by the number of pixels in each component.

As shown in Fig. 6, choosing a smaller threshold for θ_c results in a smaller minimum theoretical MADE, at the cost of a larger number of components. This is coherent with the fact that the minimum theoretical MADE decreases as the number of components approaches the total number of pixels, with a minimum value of 0 in the limit of per-pixel components, since by definition a perfect reconstruction could be achieved by appropriately scaling the depth of each pixel.

Importantly, we note that for all objects in the benchmark the minimum theoretical MADE achievable by our method is significantly smaller than the accuracy reached by state-of-the-art methods, by different margins depending on the exact value of θ_c. On the one hand, this validates the effectiveness of our component formation and filling, which does not compromise the best quality that the reconstruction can achieve. On the other hand, the remaining gap between

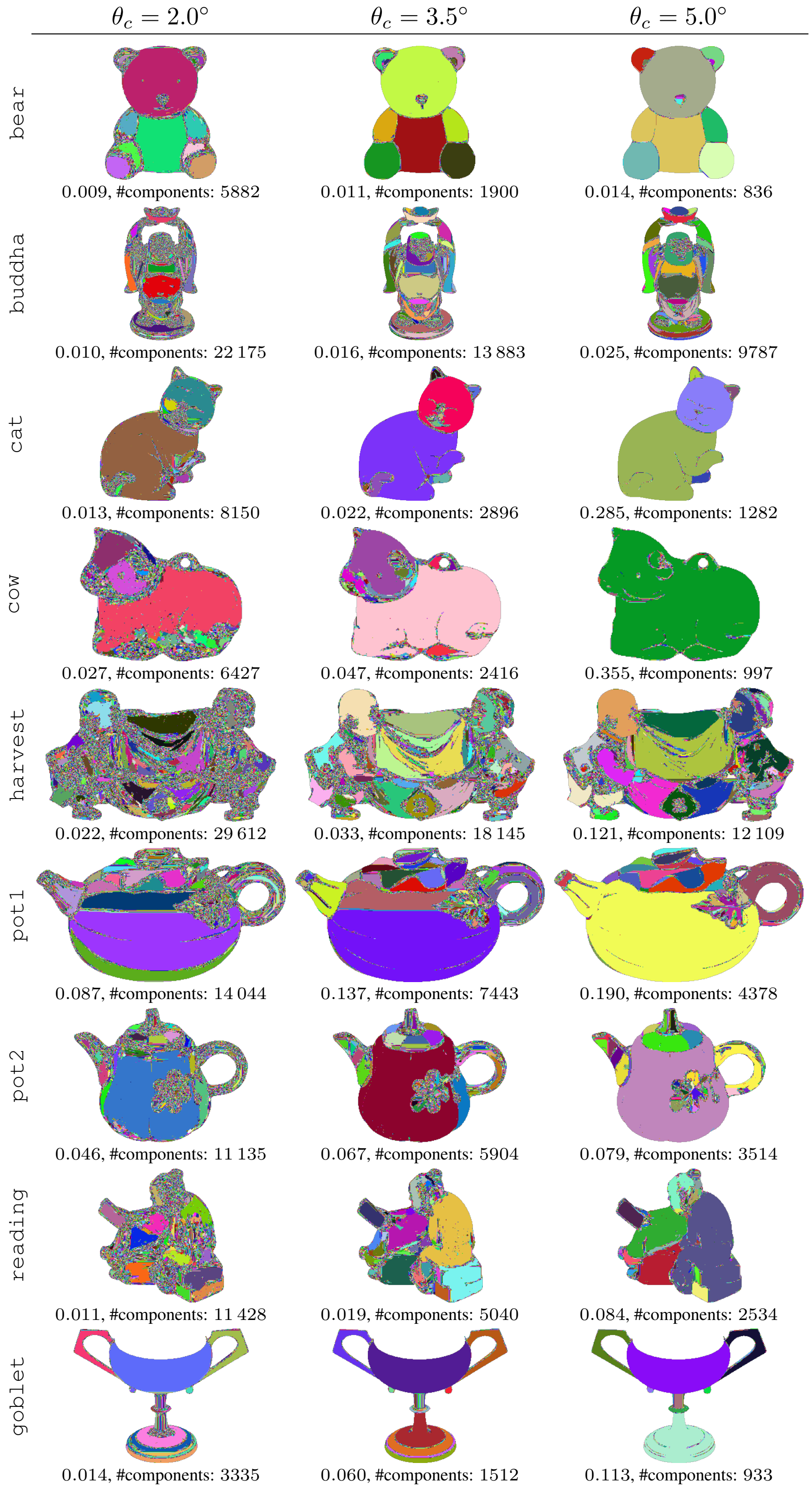


Figure 6. **Continuous components identified by our method for different normal similarity thresholds θ_c and $8-$ connectivity, DiLiGenT benchmark [22].** For each object and threshold, different colors indicate different components. Below each component image are: the minimum mean average depth error (MADE, in mm) that can be theoretically achieved by scaling the continuous components; the number of components.

the minimum theoretical MADE and the MADE actually achieved by our optimization presents an opportunity for the future emergence of more accurate models of continuity and discontinuity, which could lead to even more optimal relative scale optimization.

F. Ablation on the effect of merging in the DiLiGenT benchmark

Figure 7 shows the progression of the component decomposition, of the minimum theoretical MADE, and of the actual MADE computed from the reconstruction, at different stages of our optional merging process. The illustrated example uses a merging frequency of 5 optimization iterations; as previously noted (*cf*. Sec. D and Tab. 5), on the small-scale normals from DiLiGenT our method achieves convergence for most object already after this number of iterations. This is reflected in the fact that the reconstruction error (indicated within brackets in Fig. 7) does not change significantly after the first merge operation for most objects. However, small improvements can be observed for objects with larger discontinuities (`buddha`, `harvest`, `reading`). This indicates that in the presence of discontinuities optimization requires a larger number of iterations to converge. For this reason, merging can be a viable option to reduce the size of the optimization problem (hence also the execution time of later iterations) while allowing the convergence process to continue. As already observed (Sec. D), while the computational effect of this operation might be negligible for small-scale normal maps, its impact becomes significantly more important for larger-scale normal maps, for which a reduction in the number of variables can largely reduce the run time of the optimization (*cf*. Fig. 4).

We note, finally, that merging in general increases the minimum theoretical MADE, since it “fixes" the relative scale of neighboring components to a value that in general does not coincide with its optimal one. However, we stress that the merging operation per se does not increase the reconstruction error with respect to the previous optimization steps. This is because it does not alter the relative scales to which the optimization had converged at the preceding step, but simply relabels pixels in different components so that their scale is jointly optimized in subsequent iterations.

G. Ablation on pixel connectivity

We ablate the impact of the chosen pixel connectivity on the DiLiGenT benchmark, comparing $8-$ connectivity, which we use in our main experiments, with $4-$ connectivity. For a given configuration, we use the connectivity both in the component detection/filling stage and in the subsequent relative scale optimization. The results of the ablation are shown in Tab. 6. Overall, no major differences emerge between the two connectivities, with comparable accuracies and runtimes across all objects. A minor exception is to be found when using normal similarity threshold $\theta_c = 5.0°$, for which for some objects (`cat`, `harvest`, `reading`) $4-$ connectivity achieves better accuracy by a significant margin.

H. Limitations

Component formation and model of discontinuity. Since our heuristic for component formation relies on the similarity between neighboring normals, if the normals are continuous across a surface discontinuity it cannot detect disconnected surface regions as separate components, as depicted in the example of Fig. 8. We note, however, that this example represents a corner case more in general of normal integration methods, as previously noted for instance by [5] (Fig. 14, Supplementary). In particular, in this case the integration problem itself is ill-posed, since in this setting it is not possible to determine whether a discontinuity is present from the normal map alone.

A similar issue arises in the case of overly-smooth input normals, with indistinct boundaries, which sometimes occur in normal maps produced as output by learning-based approaches for normal estimation, such as DSINE [3]. For these normals maps, our heuristic for component formation might sometimes merge multiple surfaces into a single component, depending on the similarity threshold θ_c (Fig. 9). We note, however, that even for an incorrect component decomposition (*i.e.*, that merges surface regions separated by a discontinuity) the corresponding reconstruction can in theory still correctly capture the discontinuity if the model of discontinuity is able to describe it. This is because in the initial phase each surface patch is reconstructed using the same model of discontinuity deployed for relative scale optimization. Viceversa, if the model of discontinuity is unable to capture the discontinuity, as is the case for full discontinuities such as the foreground-background boundaries in Fig. 9, the general problem of retrieving the scale of surface regions across such discontinuities cannot be addressed without additional knowledge.

An interesting future direction is to explore alternative, more sophisticated heuristics for component formation, for instance through the use of learning-based methods that could learn priors over the discontinuities from the normal maps. In the more general case, additional input or knowledge available depending on the setting might be incorporated in the component formation phase. For instance, if an input color image was provided, as is the case in photometric stereo or surface normal estimation, edges might be extracted from the image and composed with the heuristic based only on surface normals.

Decreased benefit for highly non-smooth normal maps. The computational advantage resulting from our method

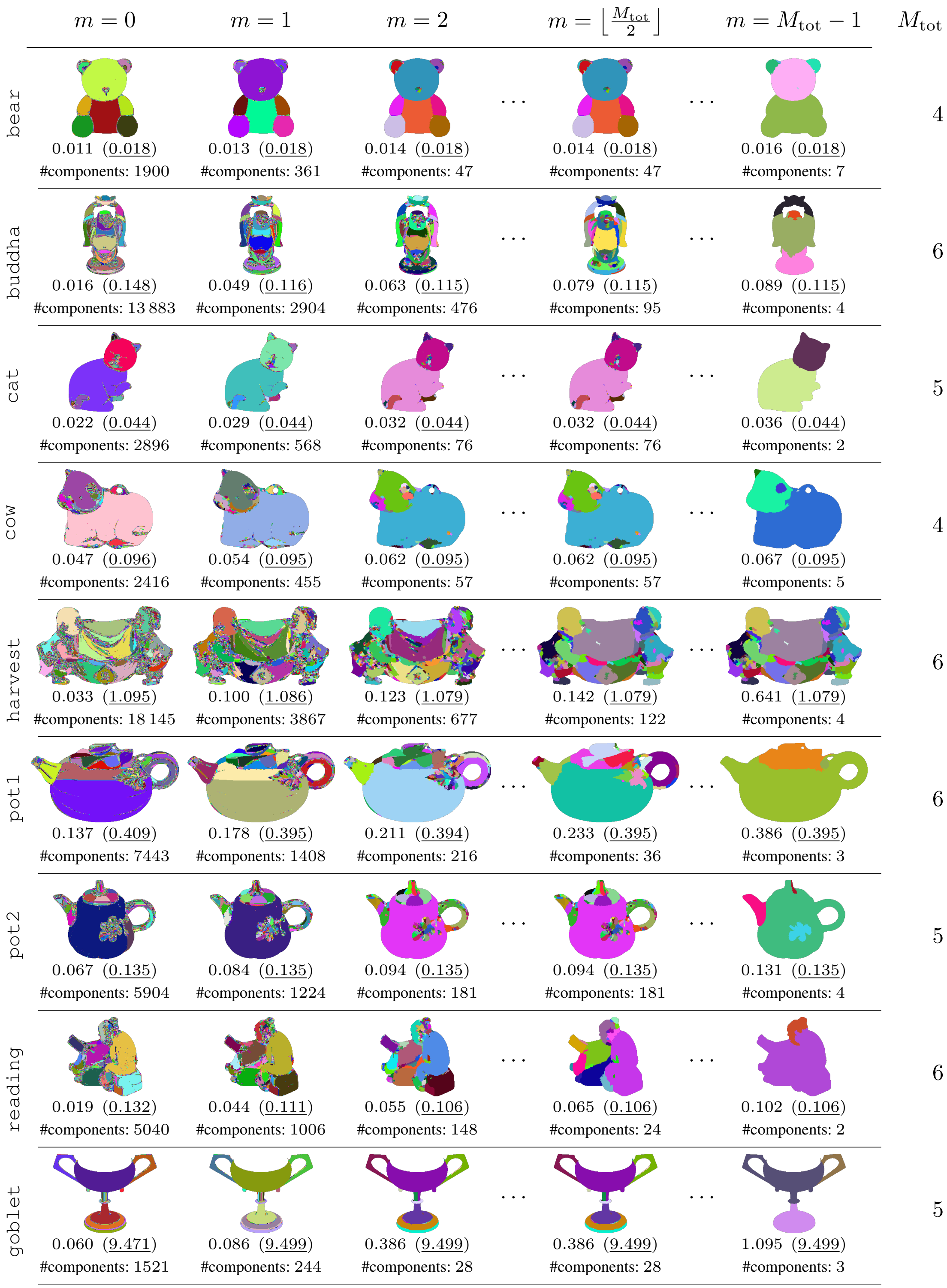
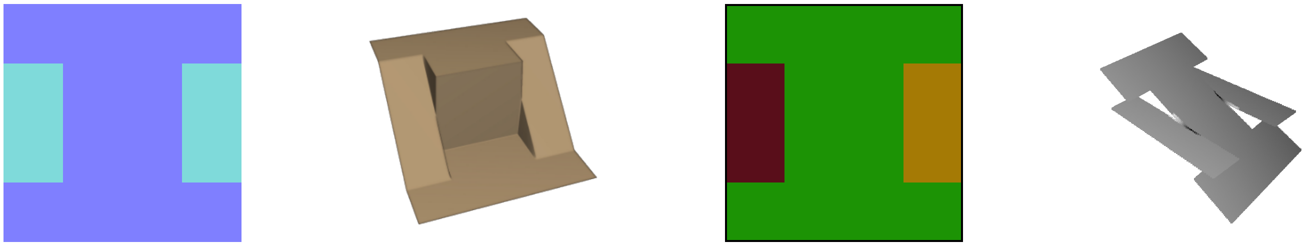


Figure 7. **Continuous components at different stages m of the merging process, DiLiGenT benchmark [22].** For each object and threshold, different colors indicate different components. Below each component image are: *first row*: the minimum mean average depth error (MADE, in mm) that can be theoretically achieved by scaling the continuous components and the MADE (mm) achieved by the optimization at the end of the merging stage, the latter in brackets and underlined; *second row*: number of components. For each object, M_{tot} denotes the total number of merging operations required to obtain a single component. For all objects, normal similarity threshold $\theta_c = 3.5°$ and 8− connectivity are used, with $\text{freq}_{\text{merging}} = 5$ and $\Delta E_{\max} = 10^{-3}$.

θ_c	Conn.	bear		buddha		cat		cow		harvest		pot1		pot2		reading		goblet	
		Err	t	Err	t	Err	t	Err	t	Err	t	Err	t	Err	t	Err	t	Err	t
None	8	0.02	7.39	0.20	19.56	0.03	36.98	0.09	3.35	1.31	44.35	0.36	32.51	0.13	9.83	0.17	3.40	9.41	4.78
2.0°	8	0.02	2.55	0.17	19.07	0.03	3.11	0.09	2.54	1.04	28.33	0.36	6.26	0.14	5.59	0.10	6.54	9.37	2.33
3.5°	8	0.02	1.27	0.11	8.03	0.04	1.50	0.09	1.53	1.07	18.81	0.38	3.51	0.14	2.66	0.09	2.68	9.49	1.40
5.0°	8	0.02	0.88	0.15	2.76	0.51	1.24	0.39	1.04	1.75	7.29	0.55	5.80	0.13	1.92	0.16	1.49	9.62	0.84
None	4	0.03	15.52	0.13	40.19	0.03	16.58	0.08	4.19	1.26	64.00	0.37	209.27	0.13	7.69	0.08	7.73	9.45	111.51
2.0°	4	0.03	2.98	0.12	26.65	0.03	9.15	0.09	2.97	1.09	29.39	0.38	8.29	0.14	27.20	0.09	17.03	9.27	1.89
3.5°	4	0.04	1.18	0.14	38.11	0.04	1.65	0.09	2.58	1.09	17.91	0.35	25.06	0.14	2.37	0.08	2.80	9.43	0.96
5.0°	4	0.02	0.77	0.12	4.46	0.03	1.23	0.09	1.14	0.80	11.11	0.57	7.72	0.13	4.84	0.10	2.21	9.51	0.73

Table 6. **Ablation on the pixel connectivity (abbreviated as** Conn.**) on the DiLiGenT benchmark [22].** The mean absolute depth error (MADE, abbreviated as Err) [mm] and the total execution time (abbreviated as t) [s] of our method are reported. All experiments use outlier reweighting $W_{b \to a}^{\text{out}}$ (20) with $L = 10^{-5}$ and $U = 10^{-3}$, convergence criteria $\Delta E_{\max} = 10^{-3}$ and $T = 150$, without merging.



(a) Input normal map

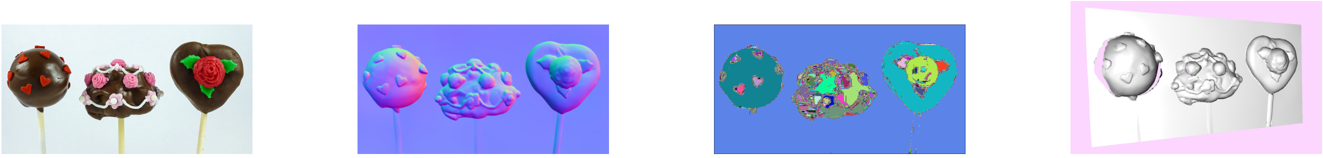
(b) Ground-truth surface

(c) Components

(d) Our reconstruction

Figure 8. **Example corner case not handled by our component-formation heuristic.** When the input normals are continuous across a surface discontinuity, our heuristic for component formation cannot detect separate components. Since the model of discontinuity that we adopt [5] cannot recover discontinuities under the same corner case, the discontinuity will not be incorporated during the component filling and the two incorrectly merged pieces of surface will jointly adjust their scale in subsequent steps of the optimization. *Source of the input normal map and ground-truth surface visualization*: [5] (Fig. 14, Supplementary).

reducing the size of the optimization problem through the use of components is partially reduced when the input normal map is highly non-smooth. In particular, if the normal vectors vary often between neighboring pixels in an area of the input map, as for instance in the case of finely-textured regions, many single-pixel components may arise (*cf*., *e.g*., the woven reed basked in the second column of Fig. 4). While our method can still be applied in such a setting, its exact computational advantage will depend on the overall balance between smooth and non-smooth regions. An interesting future direction is to design heuristics for component formation that could reduce the occurrence of such single-pixel components, for instance by detecting that highly-textured areas belong to the same connected surface region, through learned priors. Additionally, we note that in many cases the number of components is greatly increased by single-pixel components that occur at the boundaries of larger components (*cf*., *e.g*., $m = 0$ and $m = 1$ in Fig. 7). For such cases, postprocessing of the initial decomposition could be explored, for instance by merging small components into larger ones without causing the larger ones to collapse into one another.



(a) Input color image (b) Normals from DSINE (c) Components (d) Our reconstruction

Figure 9. **Example limitation due to overly-smooth input normal map.** In the case of overly-smooth input normal maps, our heuristic for component formation cannot detect separate components across the non-sharp boundaries, as is the case in this example of the boundary between the sticks in the foreground and the background. As a consequence, the foreground object is partly merged into the background. *Source of input color image*: [14].