Interpolated corrected curvature measures for polygonal surfaces (supplementary material)

J.-O. Lachaud[†1] and P. Romon[2] and B. Thibert[3] and D. Coeurjolly[4]

[1]Université Savoie Mont Blanc, France
[2]Université Gustave Eiffel, France
[3]Université Grenoble Alpes, France
[4]CNRS, INSA Lyon, Université de Lyon, France

1. Comparison between interpolated and non-interpolated corrected normal currents

In this section we numerically compare the corrected normal current with piecewise constant **u** as studied in [LRT19] (called *constant per face CNC*) with the proposed corrected normal current with prescribed **u** at vertices and interpolation on faces (called *interpolated CNC*). We evaluate the two methods on a digitized torus as illustrated on Figure 1 (see the "PRIMAL" surface of Sect. 3 for a formal definition). For large measure radius ρ, curvature estimations are very similar, but the interpolated CNC is clearly more accurate for small measure radius. It also remains meaningful for zero measure radius, while the constant per face CNC is zero almost everywhere.

The interpolated corrected curvature measure is thus more versatile than the constant per face approach, the former being as accurate as the latter for regular measure radius while staying meaningful for radius tending toward zero.

2. Accuracy of curvatures for different discretizations of the same shape

Now we compare the sensitivity of mean curvature estimators to different polygonal approximation of the same shape, here a torus of big radius 3 and small radius 1. We evaluate mean curvature estimations of $\hat{H}^{cnc}$ and $\hat{H}^{rz}$ (not $\hat{H}^{nc}$ since we have already shown in the paper that it is less accurate than our method). We consider meshes that are sampling of the torus (meshed torus, 3-twisted meshed torus) and meshes that are solely approximation of the torus (noisy torus, digital torus). Figure 2 shows that Rusinkiewicz estimator $\hat{H}^{rz}$ is slightly better than our method on clean mesh sampling of the torus, but our estimator $\hat{H}^{cnc}$ outperforms it in more difficult situations, and is always more accurate when the measure radius is well adapted to the mesh size and the perturbation of the data.

3. Pointwise convergence of curvature estimators on discretized polynomial surfaces

In this section we evaluate quantitatively the pointwise convergence of our curvature estimators, and we compare them with Rusinkiewicz's method [Rus04] and Cohen-Steiner and Morvan Normal Cycle method [CSM03]. We have a ground-truth C^∞ polynomial surface S, here a "Goursat" polynomial $3(x^4 + y^4 + z^4) - 200(x^2 + y^2 + z^2) = 800$. In order to build meshes approximating this surface with finer and finer sampling and to vary their quality of sampling/approximation, we consider the four kinds of polygonal meshes illustrated in Figure 3.

For the sake of completeness, we first recap some definitions. For an arbitrary gristep h, the h-digitized shape Z_h of S is defined as $\{\mathbf{x} \in h\mathbb{Z}^3, S(\mathbf{x}) \le 0\}$. The "PRIMAL" surface is the topological boundary of the voxels of Z_h, seen as a union of cubes with side length h, generally called a digital surface. It does not sample S and its canonic normals are bad whatever h. The "PPRIMAL" surface is obtained by projecting "PRIMAL" vertices and cell combinatorics onto S and is thus a sampling of S. The "DUAL" surface is the dual mesh to "PRIMAL". The "PDUAL" surface is obtained by projecting "DUAL" vertices and cell combinatorics onto S and corresponds to the "Marching-cubes" surface of the polynomial implicit function.

The "PRIMAL" surface is an interesting testbed for curvature estimators, since it is not a sampling of S but just an Hausdorff approximation, with extremely poor geometric normals (six possible vectors). However each edge has a length h and each face is a square, so it is a very regular mesh. For the "PRIMAL" surface, mean and Gaussian curvature errors are displayed on Figure 4 and principal curvatures errors are displayed on Figure 5. Here the normal vector field **u** is given by digital Integral Invariant normal estimator [CLL14], otherwise results with geometric normals would be much worse and not convergent. Clearly, our estimators outperform the others by several orders of magnitude and are convergent.
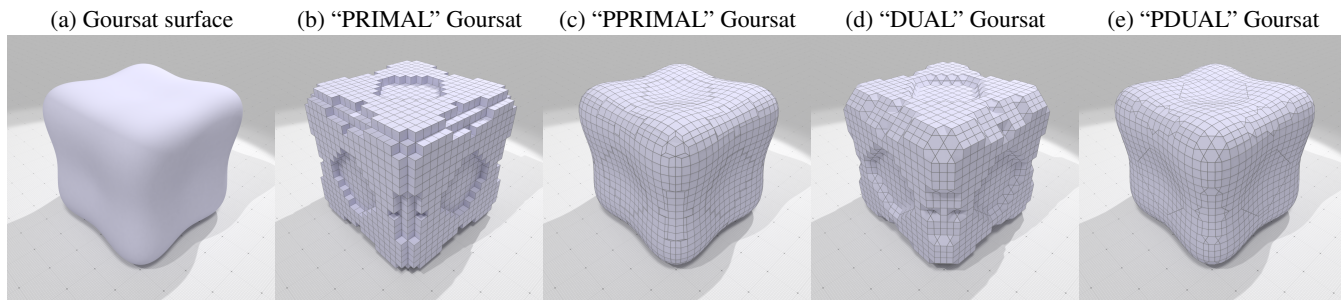
Like the "PRIMAL" surface, the "DUAL" surface is not a sampling of S but just an Hausdorff approximation (although a better one), with a limited number of possible geometric normals (but
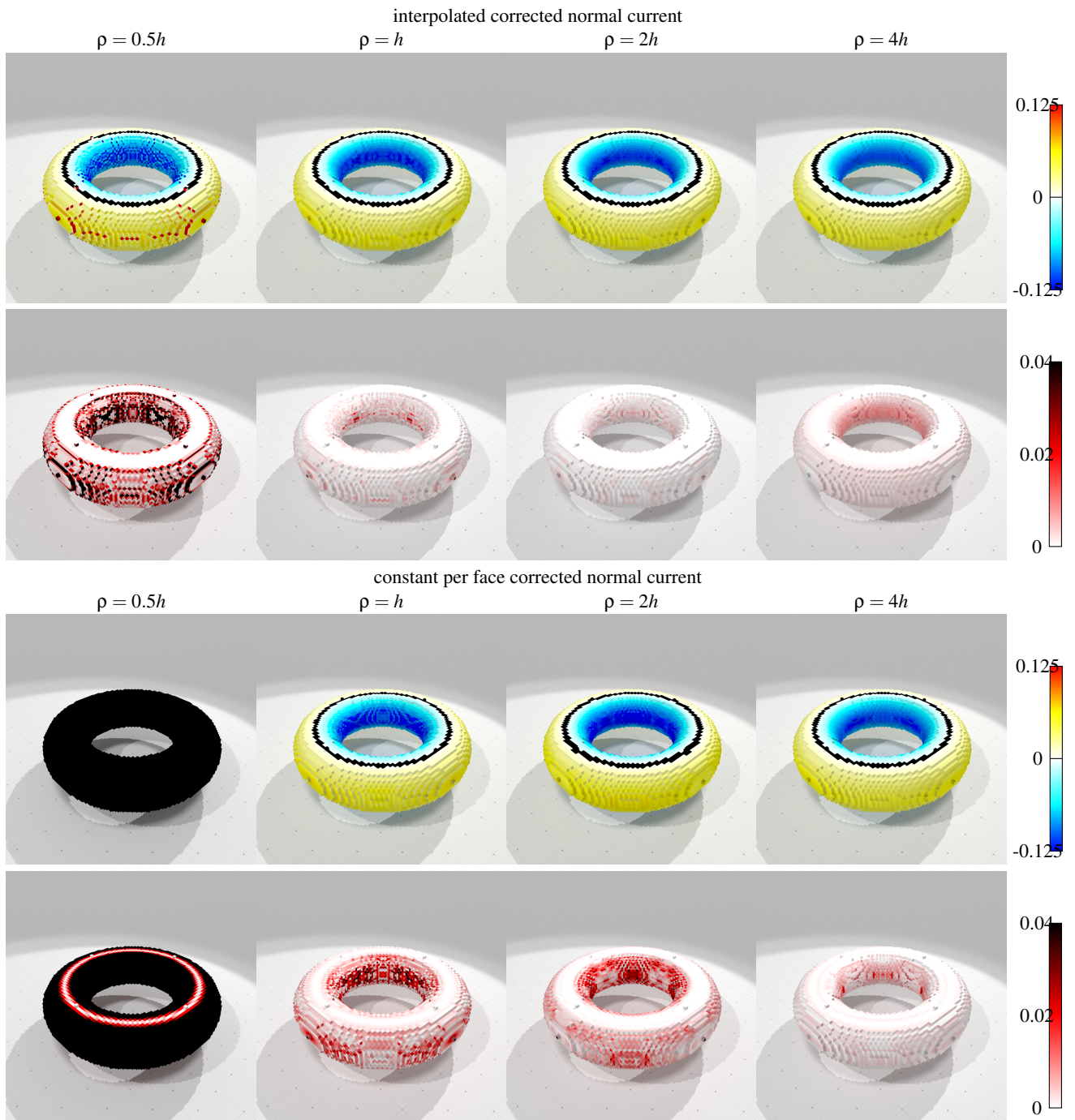
Figure 1: *Gaussian curvature estimation with true normals (expected min = −0.125, max = 0.0625) on torus shape with gridstep h = 0.25. Top two rows is the proposed interpolated corrected normal current $\hat{G}^{cnc}$, with **u** prescribed at vertices as digital Integral Invariant [CLL14], and interpolated on faces. Bottom two rows is the corrected normal current with a constant **u** per face prescribed as digital Integral Invariant, and using the formula of [LRT19] to compute curvatures accross edges and vertices. Curvature displayed in colormap: blue −0.125, white 0, red 0.125 except black zone in interval] − 0.01, 0.01[. Absolute error displayed in colormap: white 0, red 0.02, black 0.04.*
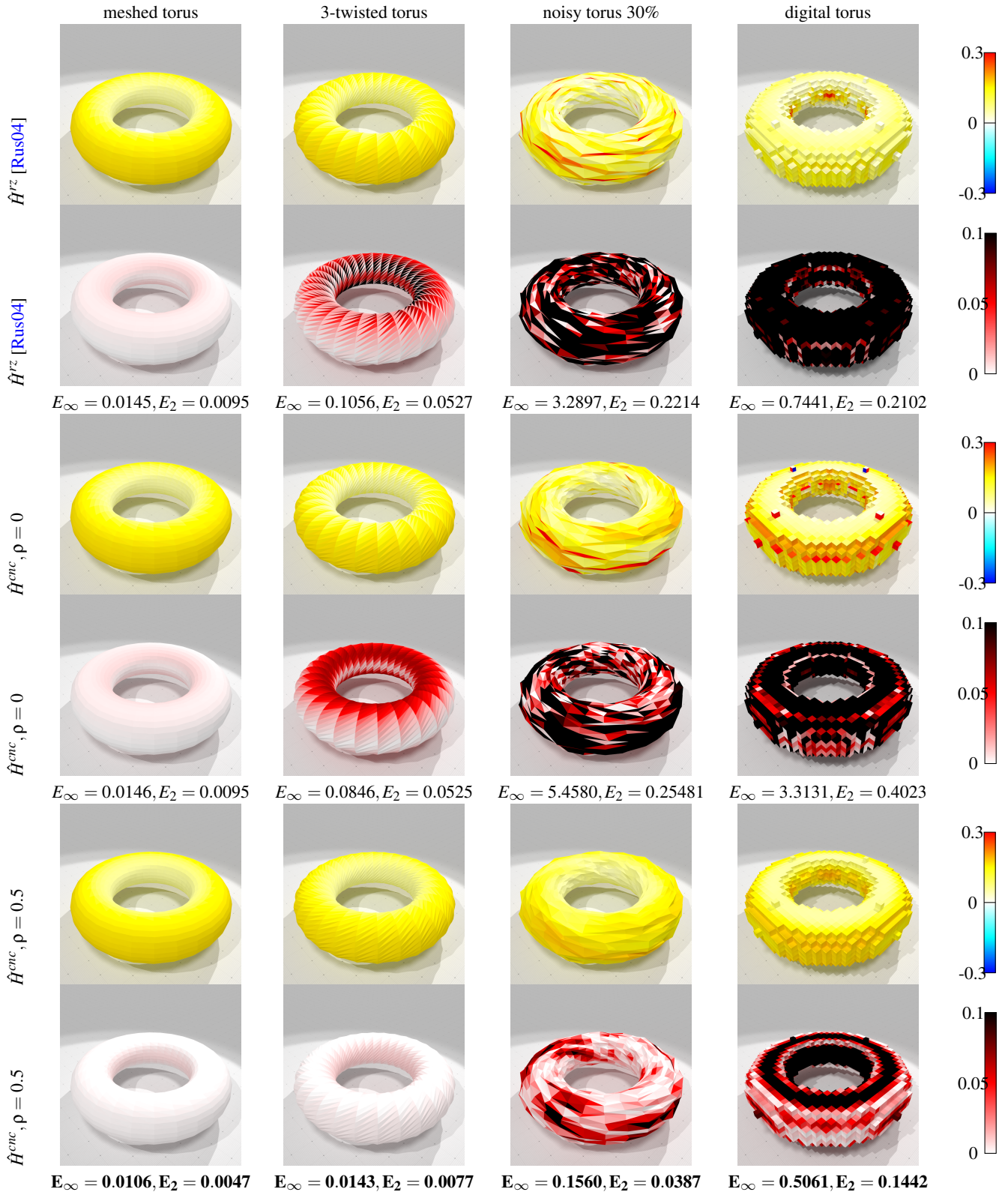
Figure 2: *Mean curvature estimation and estimation errors on a torus shape approximated by different meshes. Left and middle left, **u** is given by geometric normals. Middle right, **u** is given by averaging 4 times the geometric normals. Right, **u** is given by digital Integral Invariants. Curvature displayed in colormap: blue -0.3, white 0, red 0.3 except black zone in interval $]-0.01, 0.01[$. Absolute error displayed in colormap: white 0, red 0.05, black 0.1.*

(a) Goursat surface   (b) "PRIMAL" Goursat   (c) "PPRIMAL" Goursat   (d) "DUAL" Goursat   (e) "PDUAL" Goursat

Figure 3: *Possible input shapes for convergence tests, given input polynomial Goursat surface $3(x^4 + y^4 + z^4) - 200(x^2 + y^2 + z^2) = 800$ digitized at gridstep $h = 1$: (a) continuous polynomial surface, (b) digital surface bordering interior voxels, (c) projection of "PRIMAL" vertices onto continuous surface, (e) dual surface to "PRIMAL" surface, (e) projection of "DUAL" vertices onto continuous surface.*
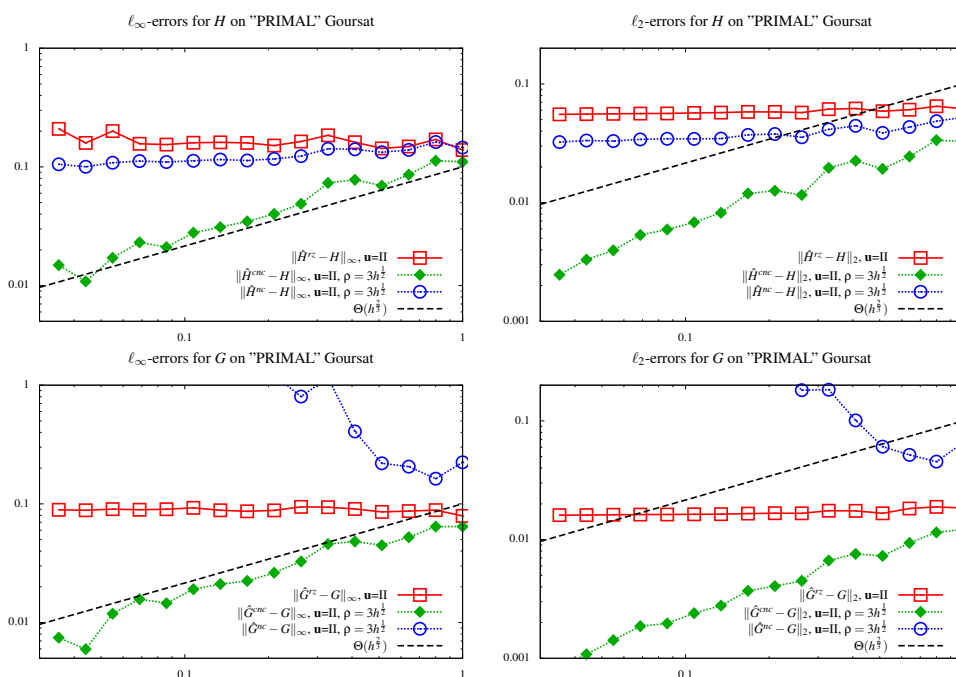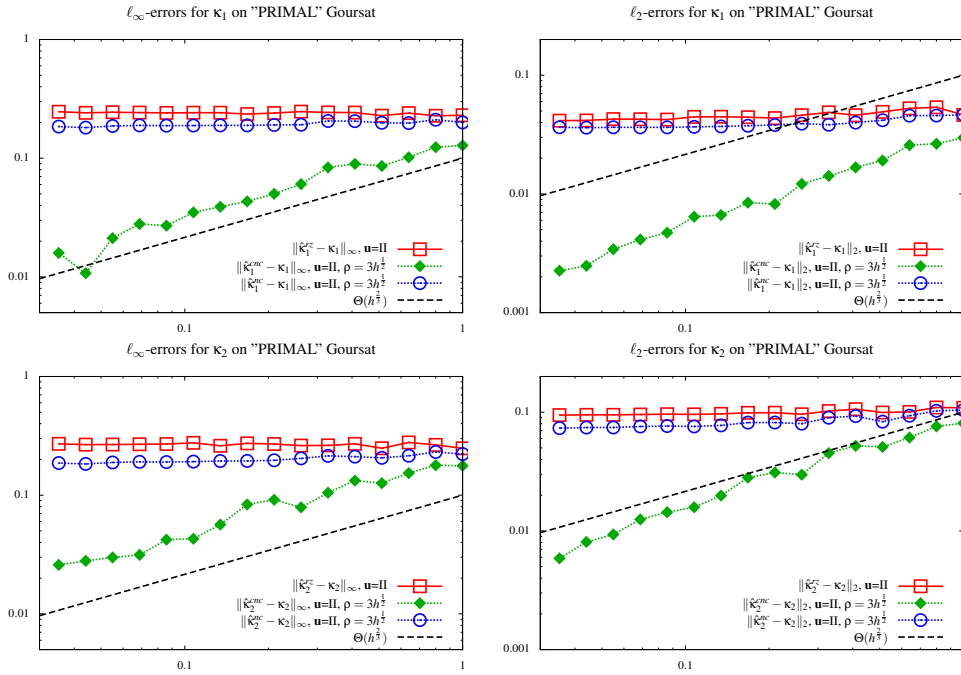

Figure 4: *Pointwise convergence for mean curvature H (top row) and Gaussian curvature G (bottom row) on "PRIMAL" Goursat: in abscissa, parameter h giving the sampling grid step (left: finest scale, right: coarsest scale). Normals* **u** *given by Integral Invariant. Left: ℓ_∞-error, Right: ℓ_2-error.*

much more than "PRIMAL". Edges are also quite regular (four possible edge lengths), so it is quite a regular mesh. For the "DUAL" surface, mean and Gaussian curvature errors are displayed on Figure 6 and principal curvatures errors are displayed on Figure 7. Here the normal vector field **u** is given by digital Integral Invariant normal estimator [CLL14], otherwise results with geometric normals would be much worse and not convergent. Clearly, our estimators outperforms the others and are convergent. The normal cycle mean curvature estimator also looks convergent (the version using **u**, not the orginal normal cycle), but less accurate than our estimator.

The "PPRIMAL" is a sampling of the continuous surface S, with vertices lying on S and quite good normals. However some edges

can be small and some faces can be very thin, which pose a lot of problems to Rusinkiewicz method. For the "PPRIMAL" surface, mean and Gaussian curvature errors are displayed on Figure 8 and principal curvatures errors are displayed on Figure 9. Here the normal vector field **u** is given by the geometric normals. Clearly, our estimators outperforms almost always the others and are convergent.

The "PDUAL" is a sampling of the continuous surface S, with vertices lying on S, with the combinatorics of "DUAL", and better normals than "PPRIMAL". It is very close to the classical Marching-Cubes surface, with corrected topology, and it is the nicest mesh to process. For the "PDUAL" surface, mean and Gaussian curvature errors are displayed on Figure 10 and principal cur-
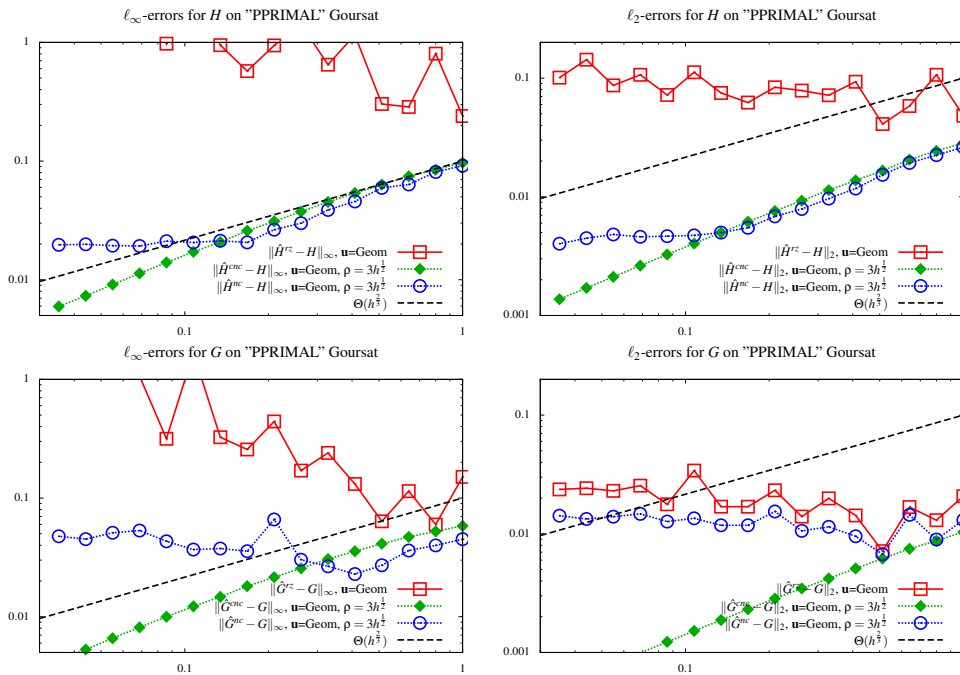
Figure 5: *Pointwise convergence for first principal curvature κ_1 (top row) and second principal curvature κ_2 (bottom row) on "PRIMAL" Goursat: in abscissa, parameter h giving the sampling grid step (left: finest scale, right: coarsest scale). Normals $\mathbf{u}$ given by Integral Invariant.* Left: *ℓ_∞-error,* Right: *ℓ_2-error.*
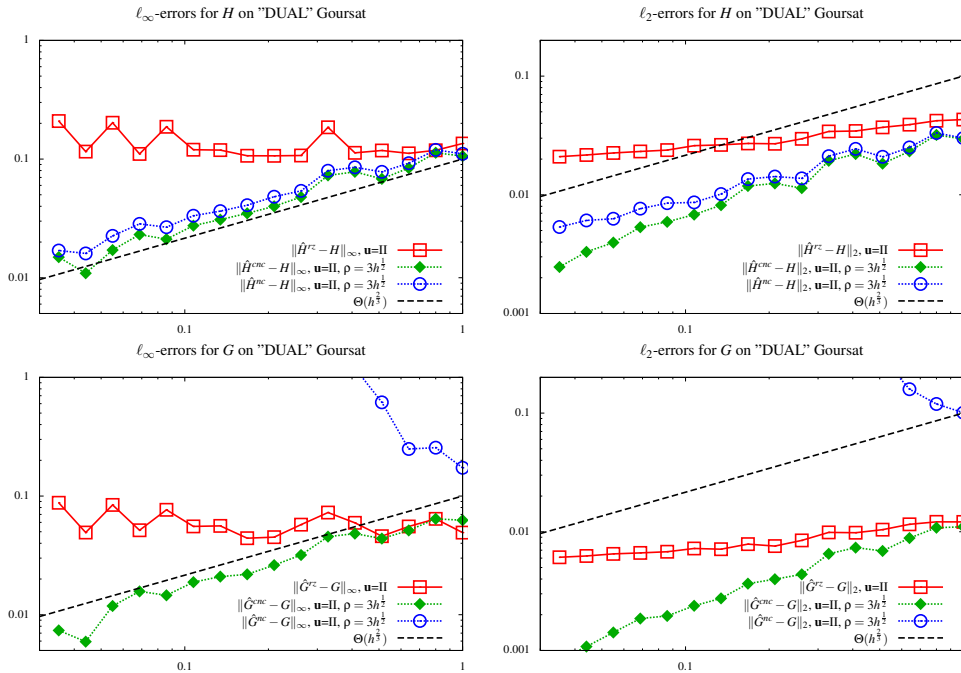
Figure 6: *Pointwise convergence for mean curvature H (top row) and Gaussian curvature G (bottom row) on "DUAL" Goursat: in abscissa, parameter h giving the sampling grid step (left: finest scale, right: coarsest scale). Normals $\mathbf{u}$ given by Integral Invariant.* Left: *ℓ_∞-error,* Right: *ℓ_2-error.*
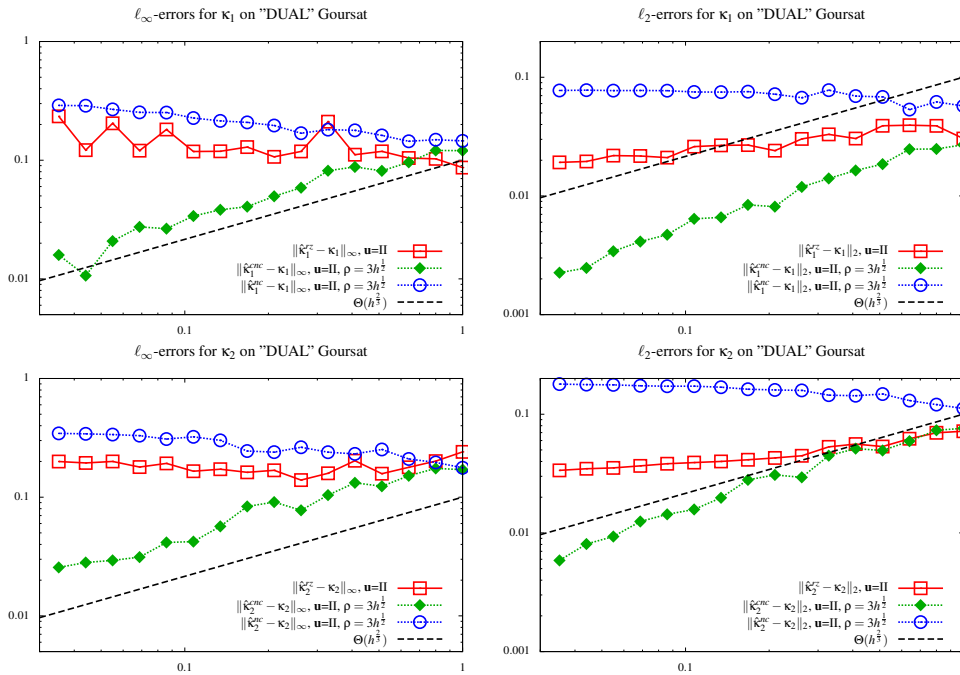
Figure 7: *Pointwise convergence for first principal curvature κ_1 (top row) and second principal curvature κ_2 (bottom row) on "DUAL" Goursat: in abscissa, parameter h giving the sampling grid step (left: finest scale, right: coarsest scale). Normals $\mathbf{u}$ given by Integral Invariant.* Left: ℓ_∞-error, Right: ℓ_2-error.
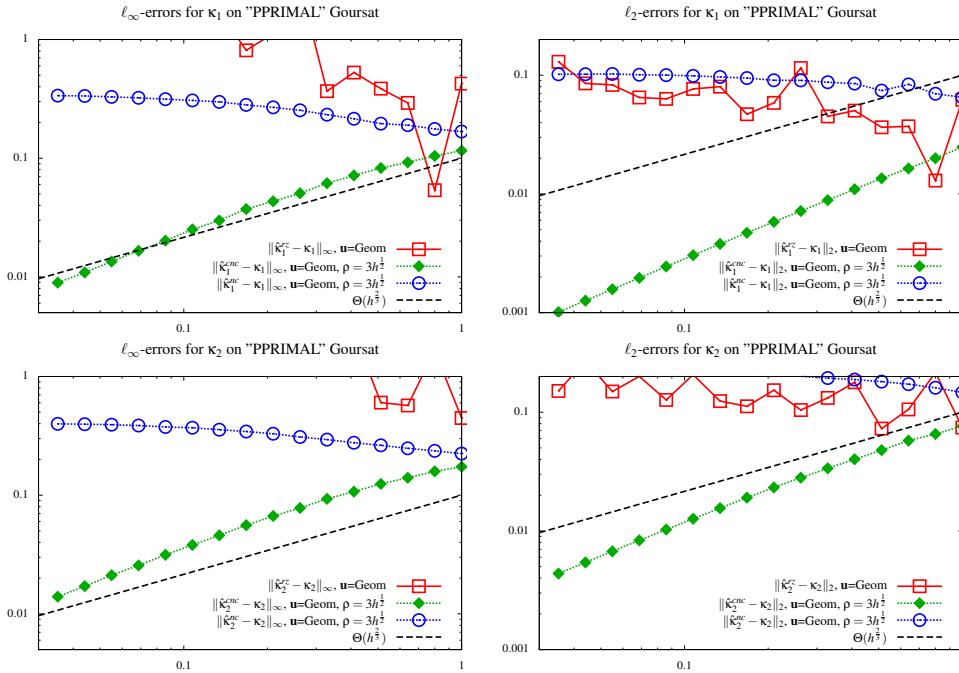

Figure 8: *Pointwise convergence for mean curvature H (top row) and Gaussian curvature G (bottom row) on "PPRIMAL" Goursat: in abscissa, parameter h giving the sampling grid step (left: finest scale, right: coarsest scale). Normals $\mathbf{u}$ given by geometry of faces.* Left: ℓ_∞-error, Right: ℓ_2-error.

Figure 9: *Pointwise convergence for first principal curvature κ_1 (top row) and second principal curvature κ_2 (bottom row) on "PPRIMAL" Goursat: in abscissa, parameter h giving the sampling grid step (left: finest scale, right: coarsest scale). Normals $\mathbf{u}$ given by geometry of faces. Left: ℓ_∞-error, Right: ℓ_2-error.*

vatures errors are displayed on Figure 11. Here the normal vector field $\mathbf{u}$ is given by the geometric normals. When the mesh is very coarse, Rusinkiewicz gives better result (essentially because the measure radius is too big at this scale). However, our estimators are convergent and outperforms it for fine enough resolution.

4. Comparisons of curvature estimations on digital surfaces

In this section, we evaluate at a given resolution how the accuracy of curvature estimators is influenced by the prescribed normal vector field $\mathbf{u}$ and by the radius of the measuring ball. Results are summed up on Figure 12. Clearly, all methods benefit from a more accurate normal vector field ($\mathbf{u}$ set to digital Integral Invariant instead of geometric normals), but our method is the most accurate when the measuring ball is set to an appropriate size. In fact, our method is already more accurate than the others for a very small measuring radius $r = h$ (for a voxel edge length of h), but convergence requires a bigger radius (see Section 3).

We also evaluate visually the accuracy of principal curvatures and principal directions on digital surfaces (only for Normal Cycle and our method since Rusinkiewicz method is too unstable for this kind of mesh). As illustrated on Figure 13, Normal Cycle first principal curvature is often overestimated while the second principal curvature is unstable (sometimes wrongly negative, and vary a lot along the sharp features of the "octaflower"). Directions are also incorrect, for instance at the tip of the "octaflower", where one can see weird alignments with axes.

We can also compare the accuracy of principal curvatures and

principal directions when a ground truth polynomial surface is known, both on a nice sampling mesh and on an approximating digital surface. As illustrated on Figure 14, our estimators $\hat{\kappa}_1^{cnc}$ and $\hat{\kappa}_2^{cnc}$ are more accurate on a polygonal surface that is a sampling of the continuous surface, but competitive results can be obtained on a coarse digital surface, provided a good prescribed normal vector field is given as input. As one can see, taking a three-times smaller integration radius for digital Integral Invariant made principal curvatures estimator much more accurate.

5. Robustness to perturbations of positions and normals

We evaluate in this section how the accuracy of estimators is degraded by perturbations in the position of vertices and how the prescribed normal vector field $\mathbf{u}$ counterbalances this perturbation. As explained in the paper, we perturb the vertices of a given mesh (here "Skull" dataset) with a uniform noise of size $0.9\bar{e}(v)$, where $\bar{e}(v)$ is the average length of edges incident to the vertex v (local 90% noise). We tested several prescribed normals at vertices (for $\mathbf{n}$ being the geometric normals): $\mathbf{u}$ is the geometrical normals $\mathbf{n}$, $\mathbf{u}$ is four times averaged the geometric normals, $\mathbf{u}$ is given by Ambrosio-Tortorelli piecewise smooth normal approximation [CFGL16] with parameter $\lambda = 0.01$ and $\alpha = 0.05$, $\mathbf{u}$ is given by Ambrosio-Tortorelli piecewise smooth normal approximation [CFGL16] with parameter $\lambda = 0.01$ and $\alpha = 0.01$, and last $\mathbf{u}$ is the geometric normals of the non-perturbed "Skull" mesh ("perfect" $\mathbf{u}$). We analyze the stability of principal curvatures estimates by extracting convex/concave/saddle zones. (See explanation of colors on Figure 12 of the companion paper.) Results are
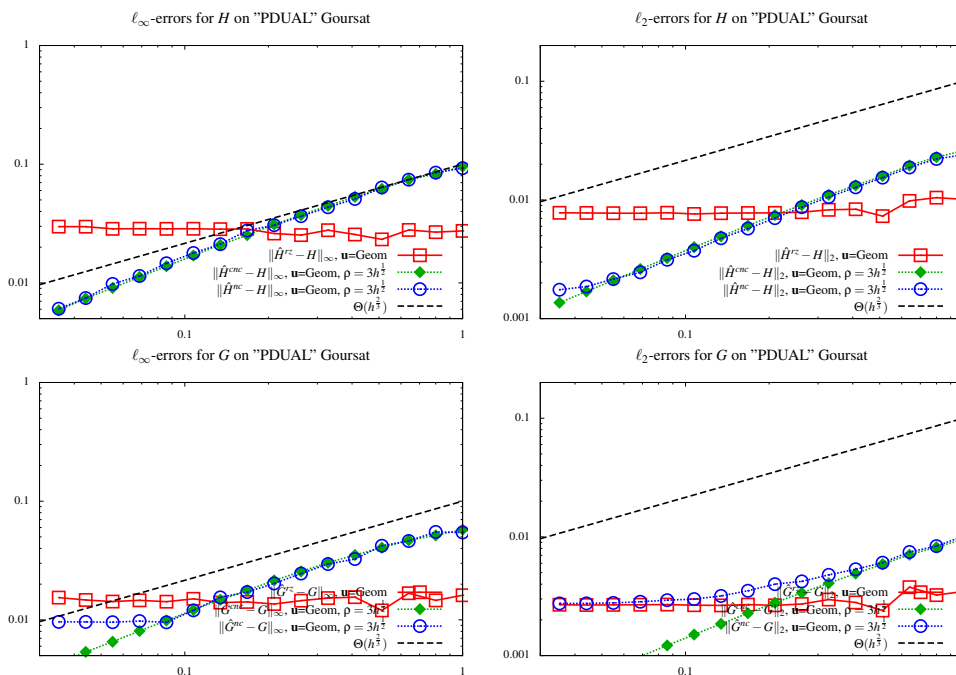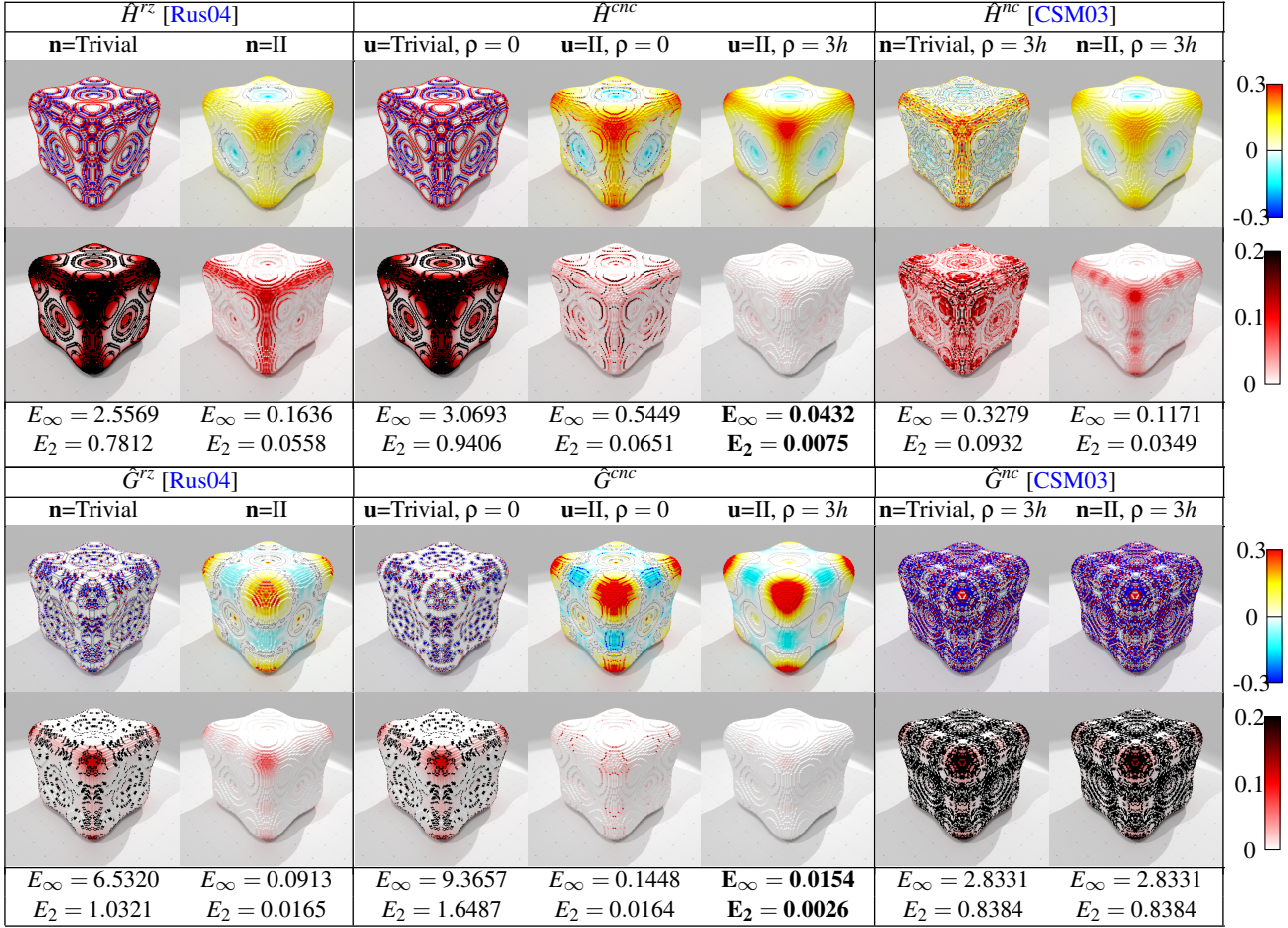
Figure 10: *Pointwise convergence for mean curvature H (top row) and Gaussian curvature G (bottom row) on "PDUAL" Goursat: in abscissa, parameter h giving the sampling grid step (left: finest scale, right: coarsest scale). Normals* **u** *given by geometry of faces.* Left: ℓ_∞-error, Right: ℓ_2-error.
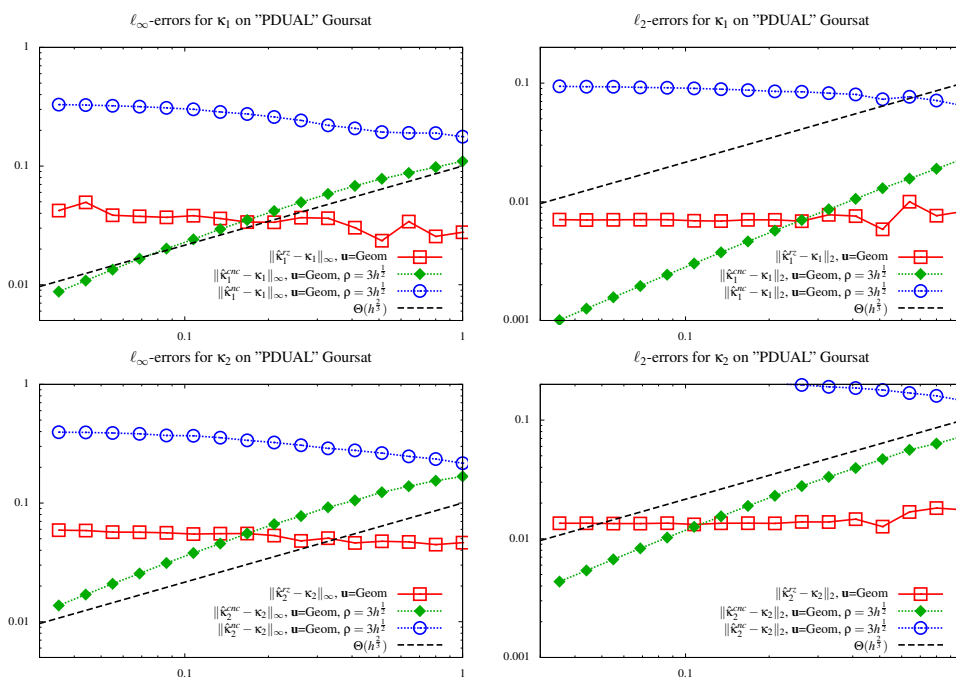

Figure 11: *Pointwise convergence for first principal curvature κ_1 (top row) and second principal curvature κ_2 (bottom row) on "PDUAL" Goursat: in abscissa, parameter h giving the sampling grid step (left: finest scale, right: coarsest scale). Normals* **u** *given by geometry of faces.* Left: ℓ_∞-error, Right: ℓ_2-error.

| $\hat{H}^{rz}$ [Rus04]		$\hat{H}^{cnc}$			$\hat{H}^{nc}$ [CSM03]	
n=Trivial	**n**=II	**u**=Trivial, $\rho = 0$	**u**=II, $\rho = 0$	**u**=II, $\rho = 3h$	**n**=Trivial, $\rho = 3h$	**n**=II, $\rho = 3h$
$E_\infty = 2.5569$ $E_2 = 0.7812$	$E_\infty = 0.1636$ $E_2 = 0.0558$	$E_\infty = 3.0693$ $E_2 = 0.9406$	$E_\infty = 0.5449$ $E_2 = 0.0651$	$\mathbf{E_\infty = 0.0432}$ $\mathbf{E_2 = 0.0075}$	$E_\infty = 0.3279$ $E_2 = 0.0932$	$E_\infty = 0.1171$ $E_2 = 0.0349$
$\hat{G}^{rz}$ [Rus04]		$\hat{G}^{cnc}$			$\hat{G}^{nc}$ [CSM03]	
n=Trivial	**n**=II	**u**=Trivial, $\rho = 0$	**u**=II, $\rho = 0$	**u**=II, $\rho = 3h$	**n**=Trivial, $\rho = 3h$	**n**=II, $\rho = 3h$
$E_\infty = 6.5320$ $E_2 = 1.0321$	$E_\infty = 0.0913$ $E_2 = 0.0165$	$E_\infty = 9.3657$ $E_2 = 1.6487$	$E_\infty = 0.1448$ $E_2 = 0.0164$	$\mathbf{E_\infty = 0.0154}$ $\mathbf{E_2 = 0.0026}$	$E_\infty = 2.8331$ $E_2 = 0.8384$	$E_\infty = 2.8331$ $E_2 = 0.8384$

Figure 12: *Influence of prescribed normal vector field* **u** *and measuring ball* ρ *on the accuracy of mean and Gaussian curvature estimators on digital surfaces. Curvature displayed in colormap: blue* -0.3, *white* 0, *red* 0.3 *except black zone in interval* $]-0.01, 0.01[$. *Absolute error displayed in colormap: white* 0, *red* 0.1, *black* 0.2.
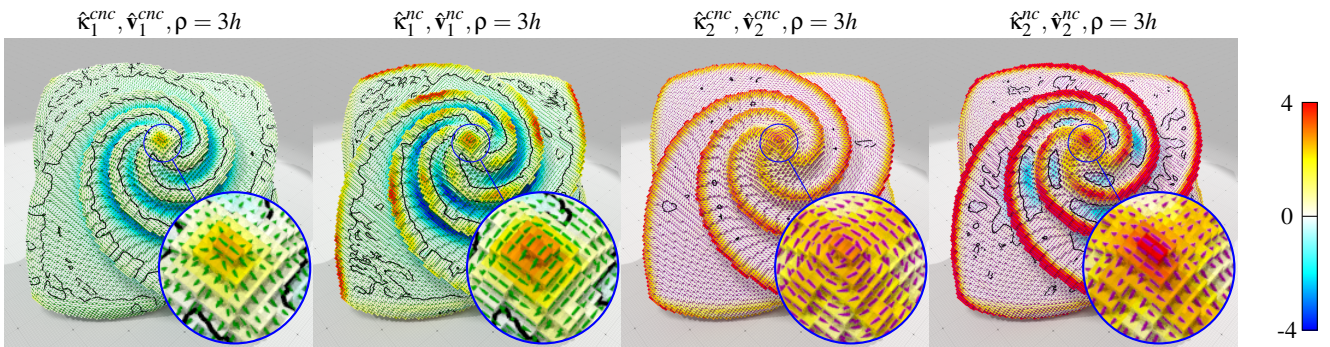
Figure 13: *Principal curvatures and principal directions on digital surface "Octaflower-128" dataset.*
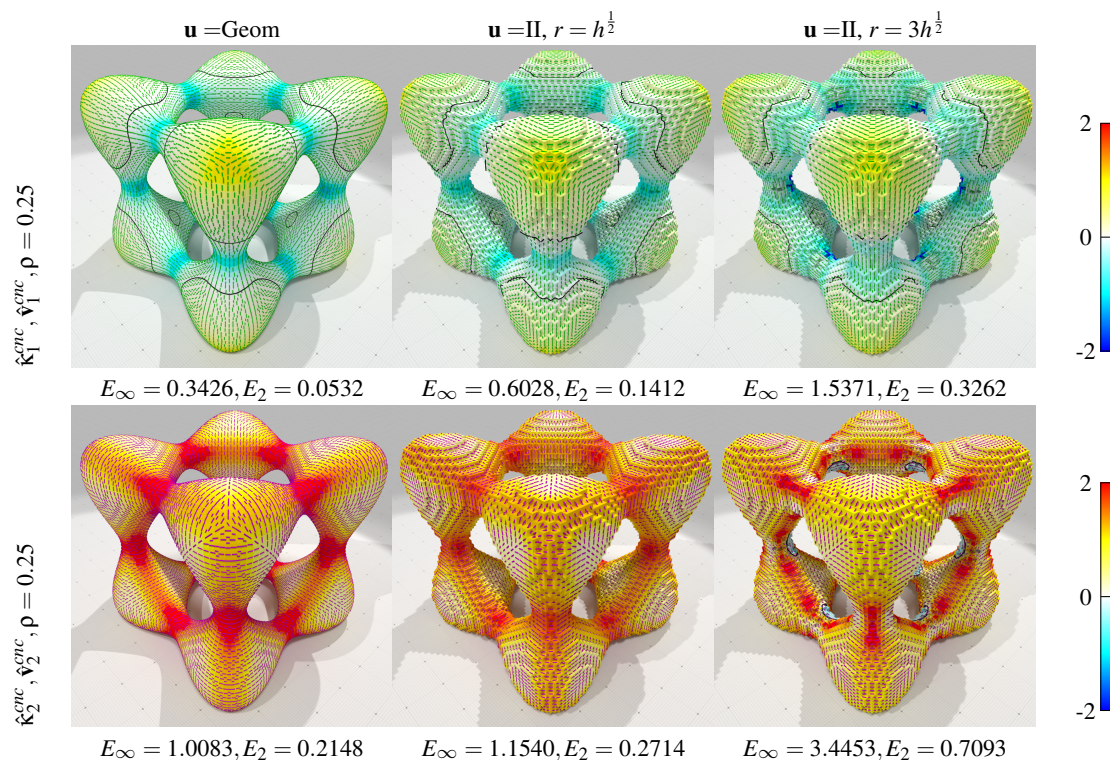
Figure 14: *Principal curvatures and principal directions estimations on different discretization of a surface "Goursat with holes" of implicit equation $x^4 + y^4 + z^4 - 8(x^2 + y^2 + z^2) + 30 = 0$. Curvatures range from -1.7589 to 3.6865.*

summed up on Figure 15 and Figure 16. Clearly, Rusinkiewicz estimators are too local when the mesh is too noisy. Normal cycle is less stable than our method and has a hard time detecting saddle zones (green). Our approach is the most stable and benefits from a more accurate prescribed vector field.

6. Tests on LIDAR data

To conclude this supplementary material, we provide two sets of experiments on LIDAR data, which present a lot of topological inconsistencies. Mean curvature estimations for different radii of measures are displayed on Figure 17. As expected, a too small radius (or zero) is too detailed and generally corresponds to noise in data. Estimator $\hat{H}^{cnc}$ and $\hat{H}^{nc}$ are visually very similar. Principal curvature estimation is illustrated on Figure 18, where we use these estimations to delineate convex, concave and saddle parts. Again, a too small radius of measure does not provide interesting insight on the shape geometry. For bigger radii, our method is generally more stable and sometimes more accurate. This is shown for instance on the upper edge of the gun, where our method hesitates between convex and cylindric parts, while Normal Cycle method sees this edge only as convex.

However both methods show their limit on such data, where manifold inconsistencies lead to the detection of weird concave zone (in blue on gun with $\rho = 0.2$). These zones are related to bad cavities in the data.

References

[CFGL16] COEURJOLLY D., FOARE M., GUETH P., LACHAUD J.-O.: Piecewise smooth reconstruction of normal vector field on digital data. *Computer Graphics Forum 35*, 7 (2016), 157–167. 7

[CLL14] COEURJOLLY D., LACHAUD J.-O., LEVALLOIS J.: Multigrid convergent principal curvature estimators in digital geometry. *Computer Vision and Image Understanding 129* (2014), 27 – 41. Special section: Advances in Discrete Geometry for Computer Imagery. 1, 2, 4

[CSM03] COHEN-STEINER D., MORVAN J.-M.: Restricted delaunay triangulations and normal cycle. In *Proceedings of the nineteenth annual symposium on Computational geometry* (2003), pp. 312–321. 1, 9

[LRT19] LACHAUD J.-O., ROMON P., THIBERT B.: Corrected curvature measures. Working paper or preprint. https://hal.archives-ouvertes.fr/hal-02193774, July 2019. 1, 2

[Rus04] RUSINKIEWICZ S.: Estimating curvatures and their derivatives on triangle meshes. In *Proceedings. 2nd International Symposium on 3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004.* (2004), IEEE, pp. 486–493. 1, 3, 9

u is **n**	**u** is 4× averaged **n**	**u** is AT,α = 0.05	**u** is AT α = 0.01	**u** is perfect	reference
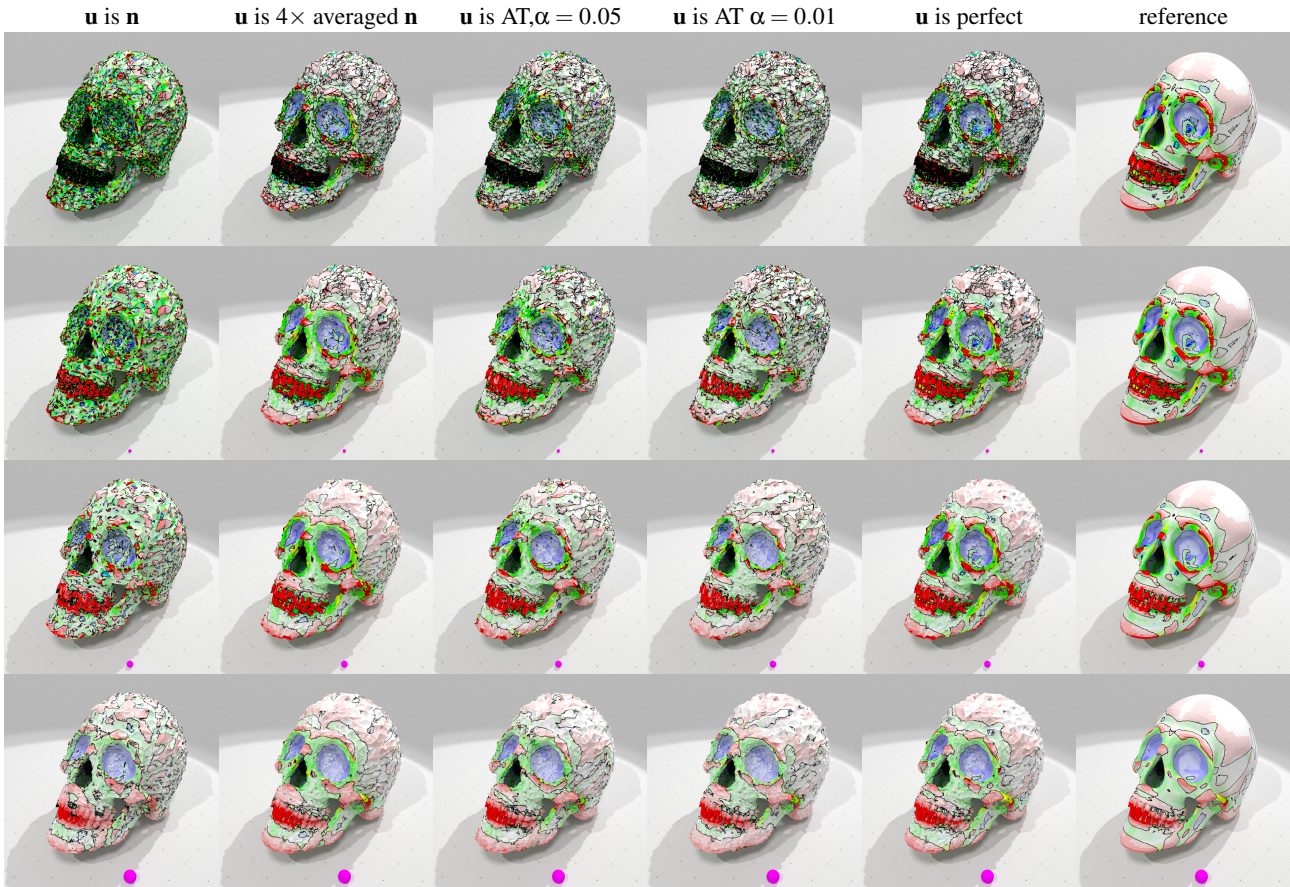
Figure 15: *Stability of convex (red), concave (blue), hyperbolic (green), concave/convex cylindric (cyan/yellow) and flat parts (white) on "Skull" dataset with increasing measure radius* $\rho \in (0, 0.05, 0.1, 0.2)$ *using corrected normal current* $\hat{\kappa}_1^{cnc}$ *and* $\hat{\kappa}_2^{cnc}$. *(See explanation of colors on Figure 12 of the companion paper.)*
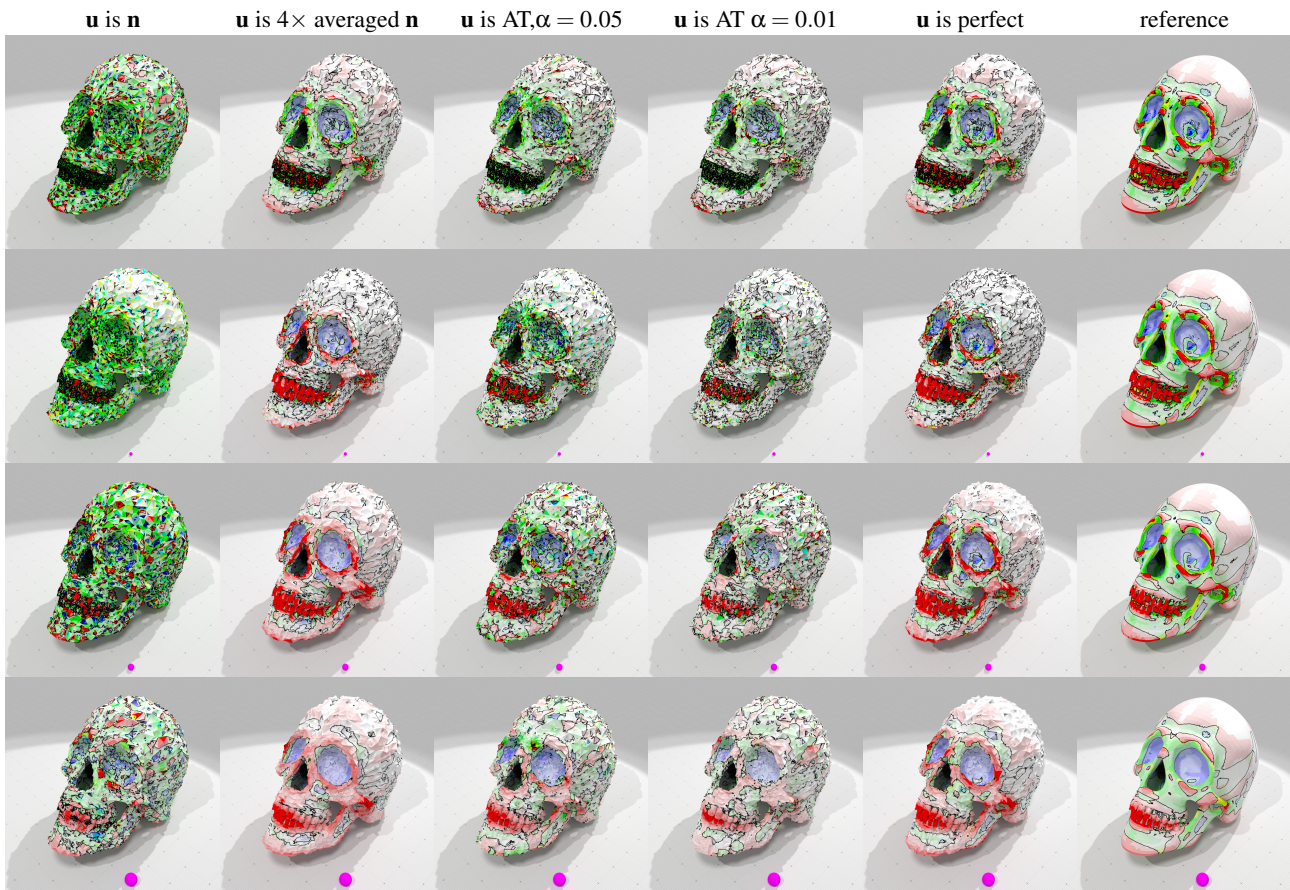
Figure 16: *Stability of convex (red), concave (blue), hyperbolic (green), concave/convex cylindric (cyan/yellow) and flat parts (white) on "Skull" dataset with increasing measure radius $\rho \in (0, 0.05, 0.1, 0.2)$ using Rusinkiewicz $\hat{\kappa}_1^{rz}$ and $\hat{\kappa}_2^{rz}$ (top) and otherwise Normal cycle $\hat{\kappa}_1^{nc}$ and $\hat{\kappa}_2^{nc}$. (See explanation of colors on Figure 12 of the companion paper.)*
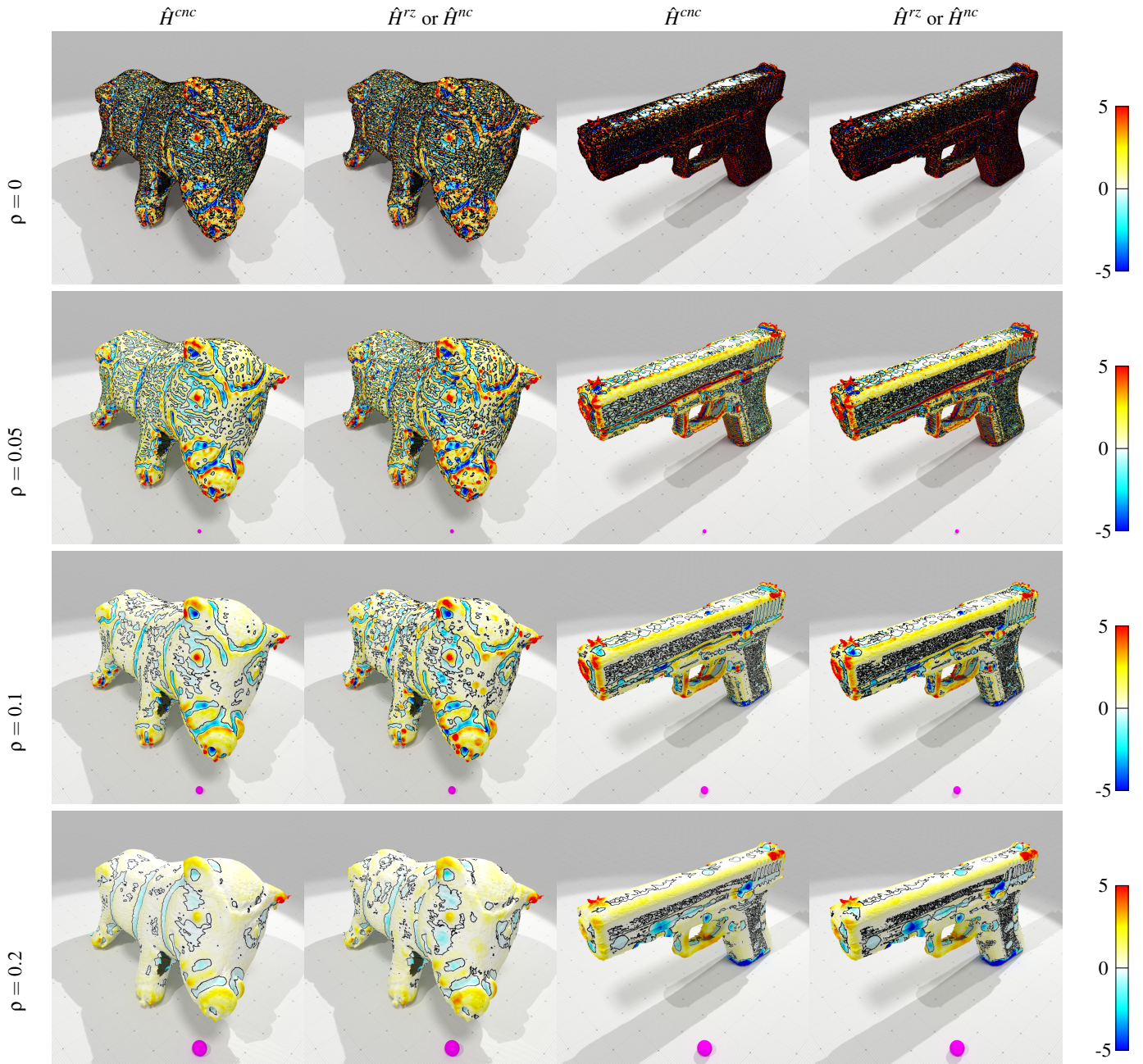
Figure 17: *Mean curvature estimation on LIDAR data according to measure radius* ρ. *Prescribed vector field* **u** *is the geometric normal vector field* **n**.
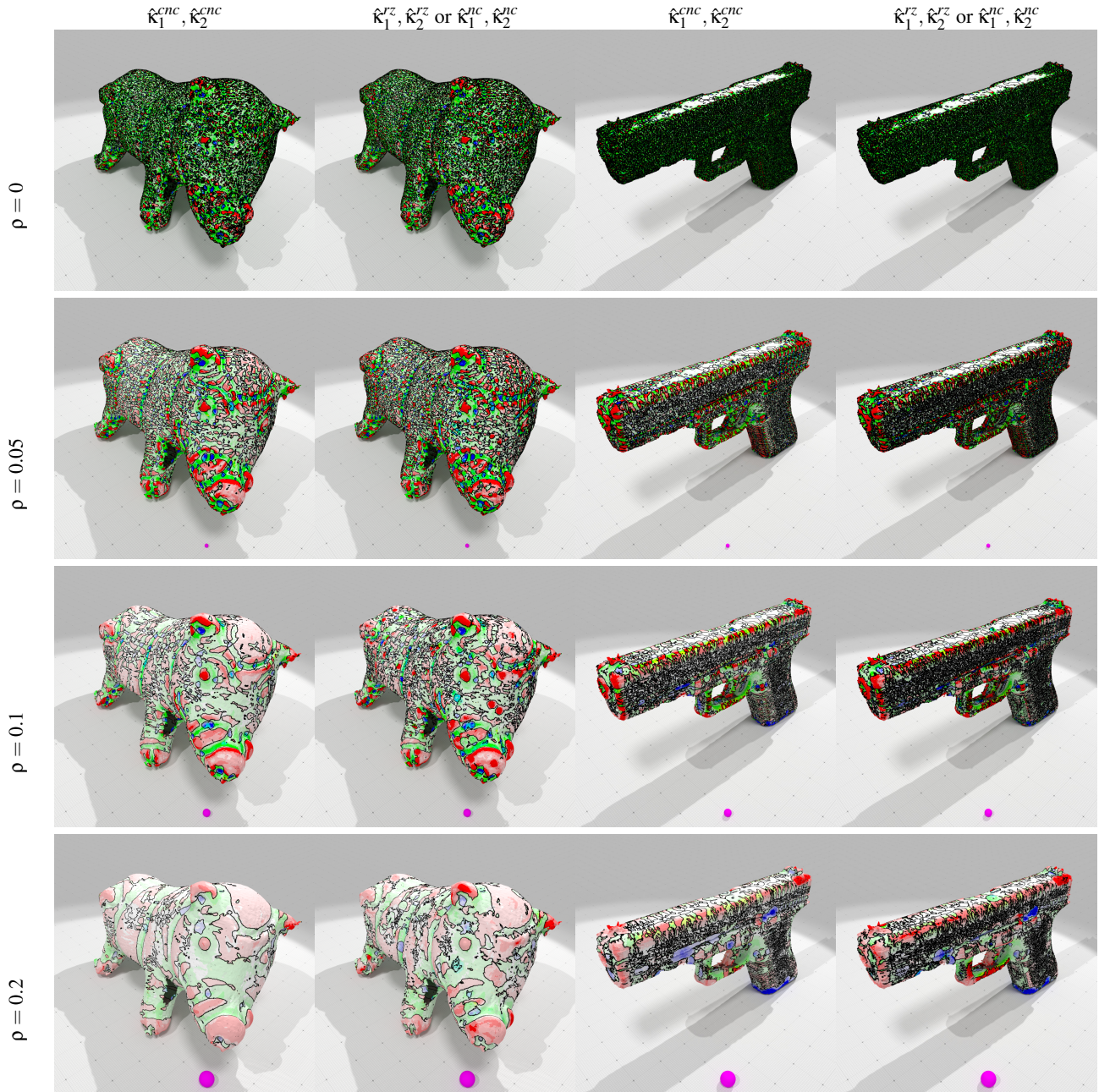
Figure 18: *Convex (red), concave (blue), hyperbolic (green), concave/convex cylindric (cyan/yellow) and flat parts (white) estimation on LIDAR data according to measure radius ρ. (See explanation of colors on Figure 12 of the companion paper.) Prescribed vector field **u** is the geometric normal vector field **n**.*