Neighborhood filters and the recovery of 3D information

Julie Digne[1], Mariella Dimicoli[2], Neus Sabater[3] and Philippe Salembier[3]
[1] CMLA, Ecole Normale Supérieure de Cachan,
julie.digne@cmla.ens-cachan.fr
[2]Laboratory of Physiology of Perception and Action
Collège de France
maria.dimiccoli@college-de-france.fr
[3] CMLA, Ecole Normale Supérieure de Cachan,
neus.sabater@cmla.ens-cachan.fr
F-94235 Cachan
[4] Image and Video Processing Group
Technical University of Catalonia (UPC)
philippe@gps.tsc.upc.edu

March 15, 2010

Abstract *Following their success in image processing (see Chapter* **??***), neighborhood filters have been extended to 3D surface processing. This adaptation is not straightforward. It has led to several variants for surfaces depending on whether the surface is defined as a mesh, or as a raw data set point. The image grey level in the bilateral similarity measure is replaced by a geometric information such as the normal or the curvature. The first section of this chapter reviews the variants of 3D mesh bilateral filters and compares them to the simplest possible isotropic filter, the mean curvature motion.*

In a second part, this chapter reviews applications of the bilateral filter to a data composed of a sparse depth map (or of depth clues), and of the image on which they have been computed. Such sparse depth clues can be obtained by stereo vision or by psychophysical techniques. The underlying assumption to these applications is that pixels with similar intensity around a region are likely to have similar depths. Therefore, when diffusing depth information with a bilateral filter based on locality and color similarity, the discontinuities in depth are assured to be consistent with the color discontinuities, which is generally a desirable property. In the reviewed applications, this ends up with the reconstruction of a dense perceptual depth map from the joint data of an image and of depth cues.

Contents

The above abstract is enough of an introduction, and our plan follows from it. Section 0.1 reviews bilateral filters applied to 3D data set points, often organized in a triangulation (a mesh). It ends up with comparative simulations illustrating the advantage of bilateral filters on isotropic filtering. Section 0.2 considers the various cases where in an image depth values or depth cues are available, and shows that the bilateral filter used as a diffusion tool makes a good job of restoring a dense depth map.

0.1 Bilateral filters processing of meshed 3D surfaces

This section proceeds by first examining the various adaptations of bilateral filtering on meshes (triangulated 3D surfaces) and discussing their implementation, which can depend on the surface triangulation. Then it will be shown that the main bilateral filter is actually consistent on smooth surfaces with the mean curvature motion (although it behaves differently on edges). Finally several comparative experiments on synthetic and real meshes will be performed. Since a common notation is needed for all methods, we begin with a small glossary and notation summary to which the reader may refer in this section.

Glossary and notation

- $\mathcal{M}$: the mesh, namely a set of triangles
- v current mesh vertex to be denoised
- $p \in \mathcal{N}(v)$ neighborhood of the vertex made of other vertices p
- n_v, n_p, etc. normals at vertex v or point s, etc.
- $w_1(||p - v||)$, $w_2(< n_v, p - v >)$, etc.: 1D centered Gaussians with various variances, used as weighting functions applied to the distance of neighbors to the current vertex and to the distance along the normal direction at v.
- H_v, H_p, etc. curvatures at v, p, etc.

- f triangle of a mesh
- a_f area of triangle f
- c_f barycenter of the triangle f
- n_f normal to triangle f
- Π_f projection on the plane of triangle f
- V voxel containing points of the data set point
- s', v', p', n'_v processed versions of s, v, p, n_v, ...

Main adaptations proposed for meshes

The *neighborhood filter or sigma filter* is attributed to J.S. Lee [Lee83] in 1983 but goes back to L. Yaroslavsky and the Sovietic image processing theory (see the book summarizing these works [Yar85]) in 2D image analysis. A recent variant names it bilateral filter ([TM98]). The bilateral filter denoises a pixel by using a weighted mean of its similar neighbors gray levels. In the original article the similarity measure was the difference of pixel gray levels yielding for a pixel v of an image I with neighborhood $\mathcal{N}(v)$:

$$\hat{I}(v) = \frac{\sum_{p \in \mathcal{N}(v)} w_1(\|p - v\|) w_2(|I(v) - I(p)|) I(p)}{\sum_{p \in \mathcal{N}(v)} w_1(\|p - v\|) w_2(|I(v) - I(p)|)}$$

where w_1 et w_2 are decreasing functions on $\mathbb{R}^+$ (e.g. Gaussian). Thus $\hat{I}(v)$ is an average of pixel values for pixels that are similar in position, but also in value. Hence the "bilaterality". Filtering without losing the sharp features is as critical for surfaces as it is for images, and a first adaptation of the bilateral filter to surface meshes was proposed in [FDCO03]. Consider a meshed surface $\mathcal{M}$ with known normals n_v at each vertex position v. Let $\mathcal{N}(v)$ be the 1-ring neighborhood of v (ie the set of vertices sharing an edge with v). Then the filtered position of v writes $v' = v + \delta v \vec{n}(v)$, where:

$$\delta v = \frac{\sum_{p \in \mathcal{N}(v)} w_1(\|p - v\|) w_2(< n_v, p - v >) < n_v, p - v >}{\sum_{p \in \mathcal{N}(v)} w_1(\|p - v\|) w_2(< n_v, p - v >)|)} \tag{1}$$

In a nutshell, this means that the normal component of the vertex v is replaced by an weighted average of the normal components of its neighboring points which also are close to the plane tangent to the surface at v. The distance to the tangent plane takes for meshes the role that was taken for images by the distance between grey levels. If v belongs to a sharp edge, then the only points close to the tangent plane at v are the points on the edge. Thus, the edge sharpness will not be smoothed away. One of the drawbacks of the above filter is clearly the use of a mesh-dependent neighborhood. Yet, this is easily fixed by defining an intrinsic Euclidean neighborhood.

Another adaptation of the 2D bilateral filter to surface meshes is introduced in [JDD03]. This approach considers the bilateral filtering problem as a robust estimation problem for the vertex position. A set of surface predictors are linked to the mesh $\mathcal{M}$: for each triangle f the position estimator Π_f projects a point to the plane defined by f. Let a_f be the surface area and c_f be the center of f then for each vertex v, the denoised vertex is

$$v' = \frac{1}{C(v)} \sum_{f \in \mathcal{M}} \Pi_f(v) a_f w_1(\|c_f - v\|) w_2(\|\Pi_f(v) - v\|) \tag{2}$$

where $C(v) = \sum_{f \in \mathcal{M}} a_f w_1(\|c_f - v\|) w_2(\|\Pi_f(v) - v\|)$ is the weight normalizing factor and w_1 and w_2 are two Gaussians.

Thus, $w_1(||c_f - v||)$ is a weight which is small if the triangle f is close to v. Thus, this term is the classic locality-in-space term of the bilateral. Similarly, $w_2(\|\Pi_f(v) - v\|)$ measures how far the projection of v onto the plane of the triangle is from v. This weight favors the triangles f whose plane is coherent with v.

The projection on the tangent planes operator Π_f depending on the normal to f, the normals must be robustly estimated. Normals, being first order derivatives, are more subject to noise than vertex positions. Hence the method starts by denoising the normal field. To do so, the mesh is first smoothed by using the same formula as above without the influence weight w_2 and with $\Pi_f(v) = c_f$, namely an updated position

$$v' = \frac{1}{C(v)} \sum_{f \in \mathcal{M}} c_f a_f w_1(\|c_f - v\|)$$

where $C(v) = \sum_{f \in \mathcal{M}} a_f w_1(\|c_f - v\|)$. The normal for each face in the denoised mesh is then computed and assigned to the corresponding face of the original noisy mesh. It is with this robust normal field that the bilateral filter of equation (2) is applied in a second step. The bilateral filter is not iterated in this method.

In [Wan06], meshes feature-insensitively sampled are denoised using a related bilateral approach. (By feature-insensitively we mean that the mesh sampling is independent of the features of the underlying surface, like (e.g.) a uniform sampling.) The algorithm proceeds as follows: it detects the shape geometry (namely sharp-regions), denoises the points and finally optimizes the mesh by removing thin triangles. The bilateral filter is defined in a manner similar to [JDD03]. Let v be a mesh vertex, $\mathcal{N}(v)$ the set of triangles within a given range of v and n_f, a_f, c_f the normal, area and center of a facet f (a triangle). Denote by $\Pi_f(v)$ the projection of v onto the plane of f, then the denoised vertex is defined by

$$v' = \frac{1}{C(v)} \sum_{f \in \mathcal{N}(v)} \Pi_f(v) a_f w_1(\|c_f - v\|) w_2(\|\Pi_f(v) - v\|)$$

where $C(v) = \sum_{f \in \mathcal{N}(v)} a_f w_1(\|c_f - v\|) w_2(\|\Pi_f(v) - v\|)$ (weight normalizing factor).

The first step is to detect sharp regions. Several steps of bilateral filtering (as defined in [JDD03]) are applied, then a smoothness index is computed by measuring the infimum of angles between normals of faces adjacent to v. By thresholding this measurement, the sharp vertices are selected. Triangles whose three vertices are sharp and whose size does not increase during the bilateral iterations are marked as sharp. This detection done, points are restored to their original positions. Then the bilateral filtering formula is applied to sharp vertices only, and the geometry sharpness is encoded into a Hermite data collection containing normals, centers and areas of filtered triangles. Points are then restored to their original position. Each sharp vertex is moved using the bilateral filtering over the neighboring stored data units, and thin vertices are removed from the mesh (these last two steps are iterated a certain number of times). Finally a post-filtering step consists in applying one step of bilateral filtering on all non sharp edges.

In [WYC06] a two-steps denoising method combines the fuzzy C-means clustering method (see [Dun73]) with a bilateral filtering approach. Fuzzy C-means is a clustering technique that allows a piece of data to belong to two different clusters. Each point p gets a parameter $\mu_{p,k}$ which measures the degree of membership of p to a cluster k. Let m_p be the number of points in the spherical neighborhood S of a point p. If $m_p < threshold$ the point is deleted. Otherwise a fuzzy C-means clustering center c_p is associated with p. The normal at point c_p is computed as the normal to the regression plane of the data set in a spherical neighborhood of p. Fleishman's bilateral filter [FDCO03] is used to filter c_i which yields the denoised point. This hybrid and complex method is doubly bilateral. Indeed, the previous C-means clustering selects an adapted neighborhood for each data set point and replaces it by an average which is by itself the result of a first bilateral filter in the wide sense of neighborhood filter. Indeed, the used neighborhood for each point depends on the point. The second part of the method therefore applies a second classical bilateral method to a cloud that has been filtered by a first bilateral filter.

The bilateral filtering idea was also used as a part of a surface reconstruction process. In [MF04], for example, a method for reducing position and sampling noise in point cloud data while reconstructing the surface is proposed. A 3D geometric bilateral filter method for edge-preserving and data reduction is introduced. Starting from a point cloud, the points are classified in an octree, the voxel centers are filtered, representative surface points are defined and the mesh is finally reconstructed. A key point is that the denoising depends on the voxel decomposition. Indeed, the filter outputs a result for each voxel. For a voxel V, call v its centroid with normal n_v. Let w_1 and u_2 be two functions weighting respectively the distance between a point p position and the centroid location and $\delta(p, v) = \langle n_p, n_v \rangle$ be the scalar product of the normal at p normal and the normal at the centroid.

Then the output of the filter for voxel V is

$$v' = \frac{1}{C(v)} \sum_{p \in V} w_1(\|p - v\|) u_2(\delta(p, v)) p$$

where $C(v) = \sum_{p \in V} w_1(\|p - v\|) u_2(\delta(p, v))$. Here w_1 is typically a Gaussian and u_2 is an increasing function on $[0, 1]$. But this filter proves unable to recover sharp edges, so a modification is introduced: prior to any filtering for each voxel V, points of V are projected onto a sphere centered at the centroid v. Each mapped point is given a normal $\tilde{n}_p$ which has direction $p - v$ and is normalized. The geometric filtering is reduced to:

$$v' = \frac{1}{C(v)} \sum_{p \in V} u_2(\delta(\tilde{n}_p, n_v)) p \text{ with } C(v) = \sum_{p \in V} u_2(\delta(\tilde{n}_p, n_v)).$$

Although only the similarity of normals is taken into account in the above formula, the filter is bilateral because the average is localized in the voxel.

In [LYY$^+$05], the bilateral filter is interpreted as the association to each vertex v of a weighted average

$$v' = \frac{\sum_{p \in \mathcal{N}(v)} w_1(\|p - v\|) w_2(\|\Pi_p(v) - v\|) \Pi_p(v)}{\sum_{p \in \mathcal{N}(v)} w_1(\|p - v\|) w_2(\|\Pi_p(v) - v\|)}$$

where $\Pi_p(v)$ a *predictor* which defines a “denoised position of v due to p”, namely the *projection of v on the plane passing by p and having the normal* n_v. The used bilateral predictor coming from [FDCO03] is $\Pi_p(v) = v + ((p - v) \cdot n_p) n_p$. In [JDD03], the used predictor was $\Pi_p(v) = v + ((p - v) \cdot n_p) n_p$ which is the projection of v on the tangent plane passing by p. With this last predictor the corners are less smoothed out, yet there is a tangential drift due to the fact that the motion is not in the normal direction n_v but in a averaged directions of the n_p for $p \in \mathcal{N}(v)$. Therefore a new predictor is introduced:

$$\Pi_p(v) = v + \frac{(p - v) \cdot n_p}{n_v \cdot n_p} n_v$$

This predictor tends to preserve better the edges than all other bilateral filters.

The question of choosing automatically the parameters for the bilateral filter was raised in [HW05]. It was proposed to choose adaptive parameters. The adaptive bilateral normal smoothing procedure process starts by searching for the set of triangles $(T_i)_i$ whose barycenters are within a given distance of a center triangle T. (But this keeps a distance parameter anyway.) Then the influence weight parameter σ_s is computed as the standard deviation of the distance between normals $\|n(T_i) - n(T)\|$. The spatial weight parameter is estimated using a minimum length descriptor criterion (for various scales). The estimated parameters are then used to get the smoothed normal. This result is then used for rebuilding the mesh using the smoothed normals using the method described in [OBS02].

The bilateral filtering of meshes proving an efficient denoising method preserving sharp features, the idea of a trilateral filtering is a natural extension taking into account still more geometric information. For example [CT05] proposes an extension to oriented meshes of the trilateral image filter. It is a 2-pass filter: a first pass filters the normals and a second pass filters the vertex positions. Starting from an oriented mesh, a first pass denoised bilaterally the vertices normals using the following update

$$n'_v = \frac{1}{C(n_v)} \sum_{p \in \mathcal{N}(v)} n_p w_1(\|p - v\|) w_2(\|n_p - n_v\|)$$

where $C(n_v) = \sum_{p \in \mathcal{N}(v)} w_1(\|p - v\|) w_2(\|n_p - n_v\|)$. Then, an adaptive neighborhood $\mathcal{N}(v)$ is found by iteratively adding faces near v until the normals n_f of face f differ too much from n'_v. A function measuring the similarity between normals is built using a given threshold R,

$$f(v, f) = 1 \text{ if } \|n'_v - n_f\| < R; \text{ 0 otherwise.}$$

The trilateral filter for normals filters a difference between normals. Define $n_\Delta(f) = n_f - n'_v$. Then the trilaterally filtered normal n_v is

$$n''_v = n'_v + \frac{1}{C(v)} \sum_{f \in \mathcal{N}(v)} n_\Delta(f) w_1(\|c_f - v\|) w_2(n_\Delta(f)) f(v, f)$$

where $C(v) = \sum_{f \in \mathcal{N}(v)} w_1(\|c_f - v\|) w_2(n_\Delta(f)) f(v, f)$. Finally, the same trilateral filter can be applied to vertices. Call P_v the plane passing through v and orthogonal to n'_v. Call $\tilde{c}_f$ the projection of c_f onto P_v and $c_\Delta(f) = \|\tilde{c}_f - c_f\|$. Then the trilateral filter for vertices, using the trilaterally filtered normal n''_v writes

$$v' = v + n''_v \frac{1}{C(v)} \sum_{p \in \mathcal{N}(v)} c_\Delta(f) w_1(\|\tilde{c}_f - v\|) w_2(n_\Delta(f)) f(v, f)$$

where $C(v) = \sum_{p \in \mathcal{N}(v)} w_1(\|\tilde{c}_f - v\|) w_2(c_\Delta(f)) f(v, f)$ The results are similar to [JDD03] though slightly better. They are comparable to the results of [FDCO03] since both methods use the distance to the tangent plane as a similarity between points.

In [WZZY08] the authors proposed a trilateral filter with slightly different principles. A *geometric intensity* of each sampled point is first defined as depending of the neighborhood of the point

$$\delta(p) = \frac{\sum_{q \in \mathcal{N}(p)} w_{pq} < n_p, q - p >}{\sum_{q \in \mathcal{N}(p)} w_{pq}}$$

with

$$w_{pq} = w_1(\|q-p\|) w_2(\| < n_p, q-p > \|) w_h(\|H_q - H_p\|).$$

This type of filter is a trilateral filter, which means that it depends on three variables: distance between the point p and its neighbors q, distance along the normal n_p between the point p and its neighbors q, and the difference of their mean curvatures H_p and H_q)

At each point, a local grid is built on the local tangent plane (obtained by local covariance analysis), at each point of this grid, the geometry intensity is defined by interpolation. Thus, neighborhoods of same geometry are defined for each pair of distinct points and the similarity can be computed as a decreasing function of the L^2 distance between these neighborhoods.

Since the goal is to denoise one point with similar points, the algorithm proposes to cluster the points into various classes by the mean shift algorithm. To denoise a point, only points of the same class are used. This gives a denoised geometry intensity δ'_i and the final denoised position $p' = p + \delta(p) n_p$.

More recently the NL-means ([BCM05]) method which proved very powerful in image denoising was adapted to meshes and point clouds in [YBS06]. Recall that for an image $I(x)$, the NL-means filter computes a filtered value $J(x)$ as $J(x) = \frac{1}{C(x)} \int_\Omega w(x,y) I(y) dy$, an adaptive average with weights $w(x,y) = \exp -\frac{1}{h^2} \int G_a(|t|) |I(x-t) - I(y-t)|^2 dt$ and $C(x) = \int_\Omega w(x,y) dy$. Here G_a is a gaussian or a compactly supported function, so that it defines a patch. Thus, the denoised point is a mean of pixel values with weights measuring the local image similarity of patches around other pixels with the patch around the current pixel.

Consider now the adaptation to a mesh $\mathcal{M}$. Let $\Omega_\sigma(x) = \{y \in \mathcal{M} | \quad |x - y| \leq 2\sigma\}$. The smoothing is done by changing x at each step: $x^{n+1} = x^n + k(x^n) n_x^n$ with n_x the normal to $\mathcal{M}$ at x. Let S_y be the surface associated to node y. The following definitions are directly adapted from the image case (we adopt here a continous formalism for clarity):

$$k(x) = \frac{1}{C(x)} \int_{\Omega_{\sigma_2}} w(x,y) I(y) dS_y$$

$$C(x) = \int_{\Omega_{\sigma_2}} w(x,y) dS_y$$

$$I(y) = < n_x, y - x >$$

$$w(x, y) = \exp - \frac{D(x, y)}{h^2}$$

The problem is to define the similarity kernel D. Let $\sigma_2 > \sigma_3$ be the half radius of the comparison domain: σ_3 is the half radius of the neighborhood to define the geometric similarity between two points. σ_2 is the half radius of the domain where similar points are looked for. The local tangent plane at y is parameterized by t_1 et t_2. For all z of $\Omega_{\sigma_2}(y)$ the translation t is defined as $t = -(< t_1, z - y >, < t_2, z - y >)$.

A local approximation by RBF functions is built: $F_x(u, v) = p(u, v) + \sum_{w \in \Omega_{\sigma_3}(x)} \lambda_w \psi(\sqrt{u^2 + v^2})$.

RBF coefficients λ_ω are found by a linear system resolution. And the similarity kernel finally yields:

$$D(x, y) = \int_{\Omega_{\sigma_3}(y)} G_{\sigma_3}(|t|) |F_x(u_z, v_z) - I(y - t)|^2 dt$$

with $I(x - t) =< n_x, z - x >$ et G_σ a gaussian kernel.

Thus each node is compared with nodes in a limited domain around it and the weighted mean over all these nodes yields the denoised position. This results in a better feature preserving mesh denoising method, but at the cost of a considerably higher computation time.

To improve the computation time when denoising datas using neighborhood filters, Gaussian kd-trees were introduced in [AGDL09]. The method proposed was designed to compute efficiently a class of n-dimensional filers which replace a value by a linear combination of other values. The basic idea is to consider those filters as nearest neighbors search in a higher dimensional space, for example (r, g, b, x, y) in case of a 2D color image and of a bilateral filter. To accelerate this neighbor search, a Gaussian k-d tree is introduced. The non local means filter in its naive implementation has a $O(n^2 f^2)$ complexity. To apply Gaussian k-d tree, the position of a pixel is set to be the patch and the value is set to be the color value of the pixel. A simple PCA on patches helps to capture the dimensions that best describe the patches. The authors also applied the Gaussian k-d tree to perform 3D NL-means on meshes or point clouds. To produce a meaningful value to describe geometry, they use the idea of spin images; At each point sample, a regression plane can be estimated and the coordinates of the point in this coordinate system are used as values; the position of the patch is set to be a couple of values (namely the spin image which is a histogram of cylindrical coordinates).

Summary of 3D mesh bilateral filters The filters we reviewed in this section are almost all defined for meshes. Yet, with very little effort almost all of them can be adapted to unstructured point clouds by simply redefining the neighborhoods as the set of points within a given distance from the center point (spherical neighborhood). We have examined several classic variants of bilateral filters, but their main principle is to perform an average of neighboring vertices pondered by the distance of these vertices to an estimated tangent plane of the current vertex. This distance takes the role played by the grey level in image bilateral filters. This distance can be implemented in several ways by either projecting the current vertex to the neighboring triangles, or by projecting the neighboring vertices on the current triangle, or by using an estimate of the normal at the current vertex which has been itself previously filtered. An interesting and simple possibility is to directly combine distance of vertices and of their normals or even distances of vertices, normals, and curvatures (but this requires a previous smoothing to get denoised normals and curvatures). Notice that position, normal and curvature characterize the shape of cloud in a larger neighborhood. Thus at this point the obvious generalization of bilateral filters is NL-means, which directly compares point-wise the shape of the neighborhood of a vertex with the overall shape of the neighborhoods of others before performing an average of the most similar neighborhoods to deliver a filtered neighborhood.

Sticking to the simplicity of comparisons and to the essentials of bilateral filter, we shall be contented in the comparative section to illustrate the gains of the bilateral filter with respect to a (good) implementation of its unilateral counterpart, the mean curvature motion, performed by projection of each vertex on a local regression plane. The remainder of this section is divided as follows: section 0.1.1 presents experiments and comparisons on artificial shapes, and section 0.1.2 present results on some real shapes.

0.1.1 Experiments on artificial shapes

In the following experiments, we will compare the denoising of the bilateral filter as introduced in [FDCO03] with the mean curvature motion. Recall that [FDCO03] defined the update of a point as:

$$\delta v = \frac{\sum_{p \in \mathcal{N}(v)} w_1(\|p-v\|) w_2(< n_v, p-v >) < n_v, p-v >}{\sum_{p \in \mathcal{N}(v)} w_1(\|p-v\|) w_2(< n_v, p-v >)|)}$$

(see first section, equation 1 for notations). The mean curvature motion used here is the projection on the regression plane: a vertex v with normal n_v and spherical neighborhood $\mathcal{N}(v)$ is projected on the regression plane of $\mathcal{N}(v)$. In [DMMSL09], this operator was shown to be an approximation of the mean curvature motion:

$$\frac{\partial v}{\partial t} = H n_v$$

Let us first observe the effects of bilateral denoising on some artificial shapes. A 5 unit side-long cube is created with added gaussian noise with standard deviation 0.02 (fig 1).

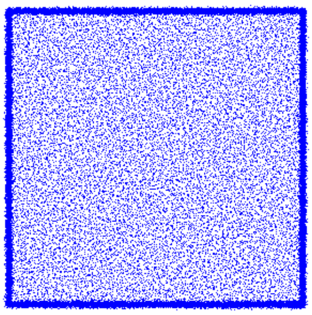
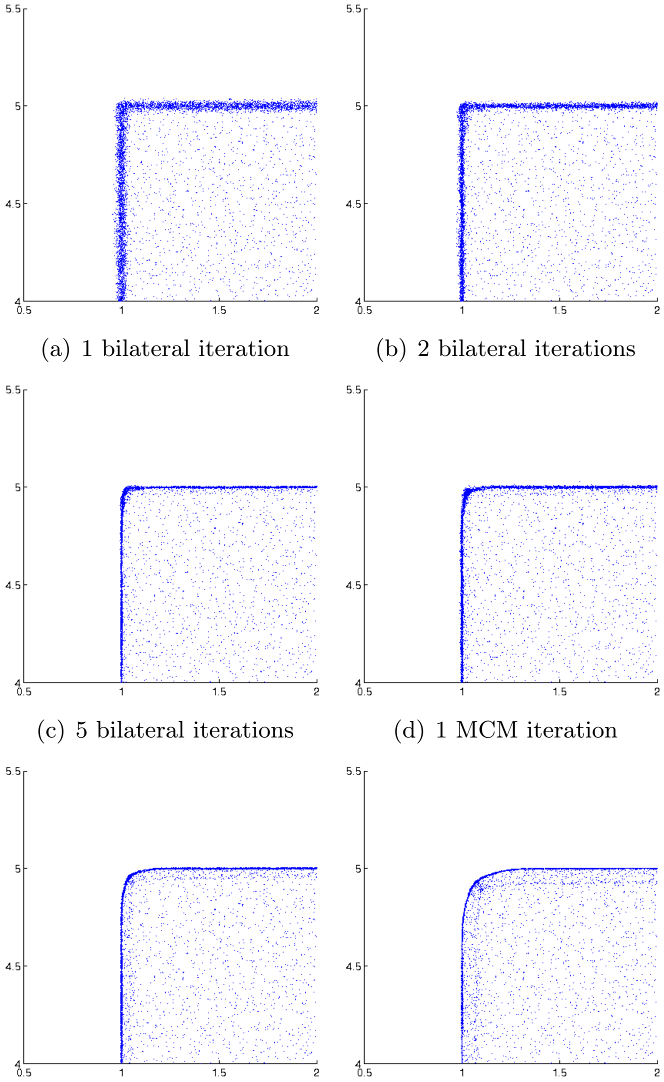


Figure 1: A noisy cube with gaussian noise



(a) 1 bilateral iteration

(b) 2 bilateral iterations

(c) 5 bilateral iterations

(d) 1 MCM iteration

(e) 2 MCM iterations

(f) 5 MCM iterations

Figure 2: Bilateral and MCM iterations on the cube corner. Notice how the sharpness is much better preserved by the bilateral filter than by the mean curvature equation.

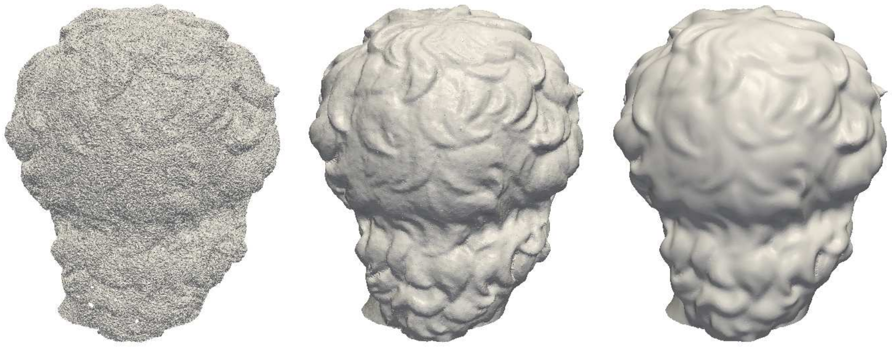
The experiments of fig 2(a, b, c) show the denoising power of the bilateral filter in term of preserving edges and should be compared with the standard mean curvature motion filter (fig 2 d, e f), the comparison is particularly interesting in the corner areas. The bilateral filter implies an anisotropic curvature motion leading to a diffusion only in smooth parts while preserving the sharp areas. Let us now see how those filters perform in case of a sharp edge. An estimation of the noise for each of the denoising methods is shown on tabular 3. These numbers tend to prove that Mean Curvature Motion, although it smoothes well the noisy flat parts also smoothes away the sharpness whereas the bilateral filter tends to preserve the sharp edges better. With few iterations the noisy parts are smoothed out, which decreases the root mean square error, then when iterating the operator, the sharpness tends to be smoothed, which increases the RMSE again. This phenomenon is of course far quicker with the mean curvature motion since this filter does not preserve the edge at all.

	Input	Iteration 1	Iteration 2	Iteration 5
RMSE (bilateral)	0.01	0.0031	0.0019	0.0035
RMSE (mcm)	0.01	0.0051	0.0085	0.0164

Figure 3: Noise estimation for the Sharp edge denoising

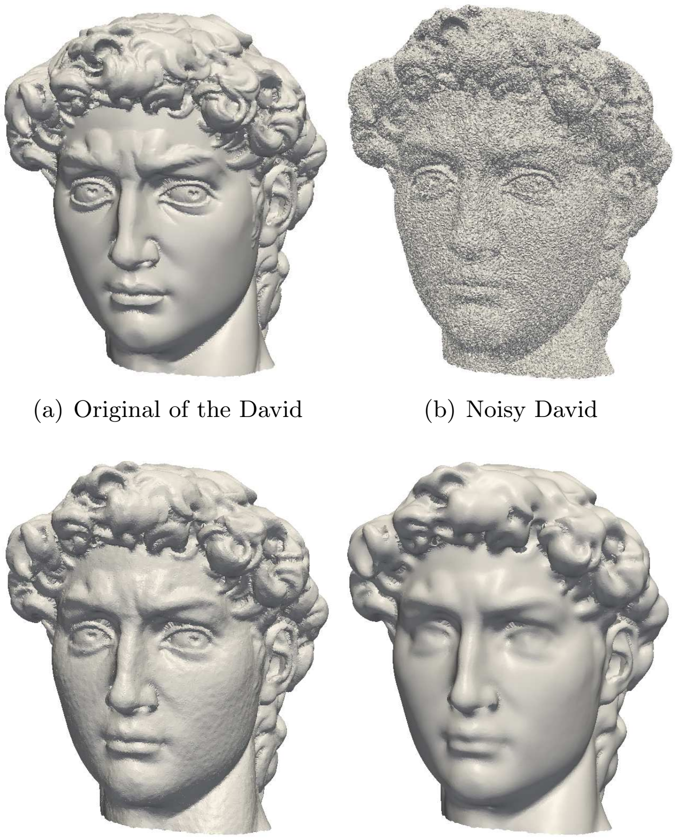
0.1.2 Experiments on real shapes

This section runs some experiments on a real shape, namely Michelangelo's David. The experiments were run on the point cloud. At each step an interpolating mesh was built for visualization.



(a) Initial noisy David (b) Bilateral denoising (c) MCM

Figure 4: Denoising of the David (back)



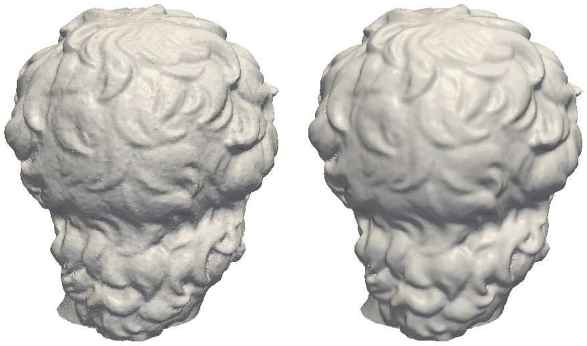
(a) Original of the David

(b) Noisy David

(c) Bilateral denoising

(d) MCM

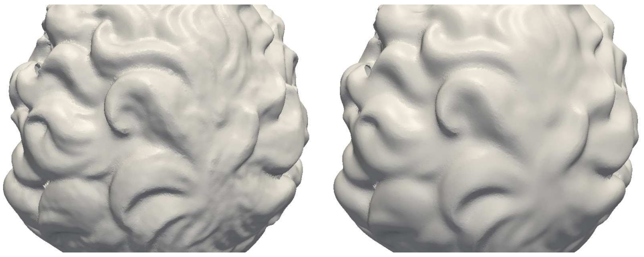
Figure 5: Denoising of the David's face



(a) Iteration 1

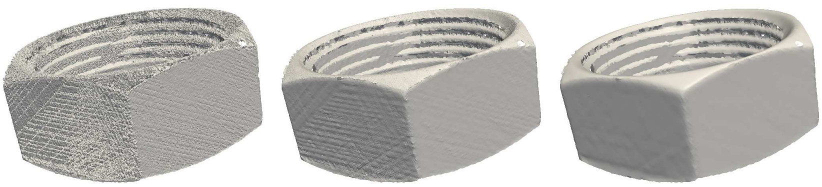
(b) Iteration 2

Figure 6: Iterating the bilateral filter on the David (back)



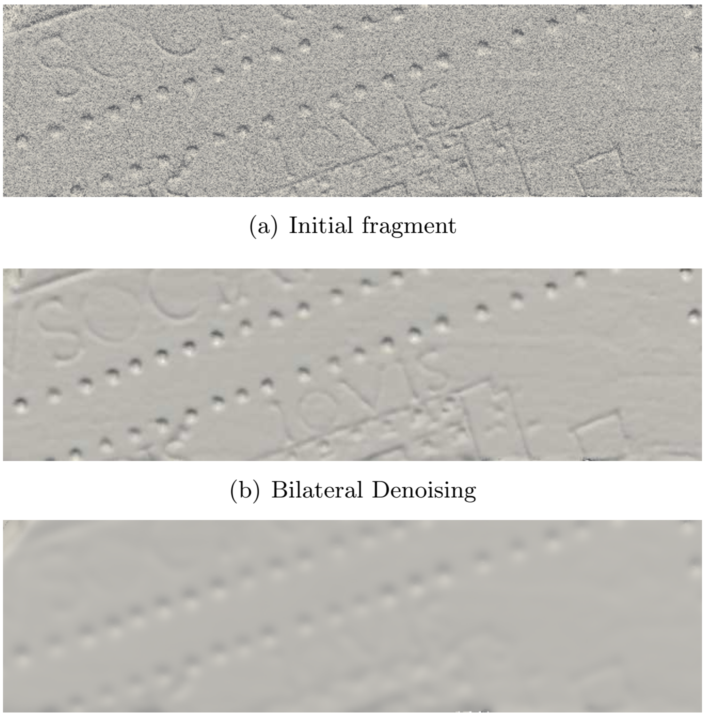
(a) Bilateral Filtering (b) MCM

Figure 7: Detail of the David



(a) Initial scan (b) Bilateral Denoising (c) MCM

Figure 8: Denoising of a screw nut driver scan.



(a) Initial fragment

(b) Bilateral Denoising

(c) MCM

Figure 9: Denoising of fragment '31u' of Stanford Forma Urbis Romae

On fig 5, denoising artefacts created by the bilateral filter can be seen.

They appear as oscillations (on David's cheek for example). These artifacts can be explained by the fact that the bilateral filter enhances structures. Added noise structures can be randomly enhanced by the bilateral filter. On fig 4, we can see that some noise remain after one iteration of bilateral denoising. We therefore iterate the bilateral filter with same parameters. Then, obviously, the remaining noise disappears at the cost of some sharpness loss (see fig 6). This can also be seen on a fragment of the Stanford Forma Urbis Romae Project fig 9 and on a noisy simple scan of a screw nut driver.

0.2 Depth-oriented applications

This section focuses on the applications of the bilateral filter and its generalized version to depth-oriented image processing tasks. The common idea to all these applications is of constraining the diffusion of depth information to the intensity similarity between pixels. The underlying assumption is that pixels with similar intensity around a region are likely to have similar depths. Therefore, when diffusing depth information basing on intensity similarity, the discontinuities in depth are assured to be consistent with the color discontinuities, which is often a desirable property [GP87, KS91].

The remainder of this section is organized as follows. Section 0.2.1 reviews the applications of the bilateral filter to stereo matching algorithms, while section 0.2.2 describes an application to the resolution enhancement of range images. Section 0.2.3 review applications to the estimation of depth in single images.

0.2.1 Bilateral filter for improving the depth map provided by stereo matching algorithms

Stereo matching algorithms address the problem of recovering the depth map of 3D scene from two images captured from different viewpoints. This is achieved by finding a set of points in one image which can be identified as the same points in the other image. In fact, the knowledge of the point-to-point correspondences permits to compute the relative disparities, which are directly related to the distance of the object normal to the image plane. The search of correspondences is usually based on a similarity measure between pixels belonging to two different images of the same scene. Due to the presence of noise and repetitive texture, these correspondences are extremely difficult to find without global reasoning. In addition, occluded and textureless regions are inherently ambiguous since the relative disparity cannot be found reliably by image matching alone.

In applications demanding accurate disparities such as structure reconstruction or multiview video coding, stereo matching algorithms provide a sparse disparity map. Improvements can be obtained through filtering or interpolation, for instance by using median or morphological filters. However, their ability to do so is limited. In [YC04], the authors have proposed a post-processing step to improve dense depth maps produced by any stereo matching algorithm. The proposed method consists in applying an iterated bilateral filter, which diffuses the depth values by relying on the gradient of the original image instead of that of the depth image. This allows to incorporate edge information into the depth map, assuring discontinuities in depth to be consistent with intensities discontinuities.

In [YK06], the color-weighted correlation idea underlying the bilateral filter has been exploited to reduce the ambiguity of the correspondence search problem. Classically, this problem has been addressed by area-based methods relying on the use of local support windows. In this approach, all pixels in a support window are assumed to have similar depth in the scene and, therefore, similar disparities. Accordingly, pixels in homogeneous regions get assigned the disparities inferred from the disparities of neighboring pixels. However, when the support windows are located in correspondence of depth discontinuities, the same disparity is assigned to pixels having different depths, resulting in a foreground-fattening phenomenon. To obtain accurate results, an appropriate support window should be selected for each pixel adaptively. This problem is addressed in [YK06] by weighting the pixels in a given support window taking into account their color similarity and geometric proximity to the reference pixel. The similarity measure between two pixels is then computed using the support-weights in the support windows of both pixels, allowing to incorporate the edge information into the disparity map. Experimental results show that the use of adaptive support-weights produces noise-removed accurate piecewise smooth disparity maps while preserving depth discontinuities.

The idea of exploiting the color-weighted correlation to reduce the ambiguity of the correspondence problem has been implemented in a parallel architecture [YWY$^+$06] [WLG$^+$06], allowing its use in real-time applications and integrated in more complex stereo systems [YWY$^+$06] [WLG$^+$06], which nowadays achieve the top rank in the Middlebury benchmark [SS].

The interpolation of disparity maps and in particular of Digital Elevation Models (DEMs) has been considered in several recent works. [FLA$^+$06] proposes to interpolate unknown areas by constraining a diffusion anisotropic process to the geometry imposed by a reference image, and coupling the process with a data fitting term which tries to adjust the reconstructed surface to the known data. More recently, [FC09] has proposed a new interpolation method which defines a geodesic neighborhood and fits an affine model at each point. The geodesic distance is used to find the set of points that are used to interpolate a piecewise affine model in the current sample. This interpolation is refined by merging the obtained affine patches with a Mumford-Shah like algorithm. The *a contrario* methodology has been used in a merging procedure. In the urban context, [LDZPD08] uses a dictionary of complex building models to fit the disparity map. However, the applicability of such a method is less evident because of the initial delineation of buildings by a rectangle fitting.

The bilateral filter averages the pixel colors, based on both their geometric closeness and their photometric similarity, preferring of course near values to distant values in space and color. [Yoo06] have used it to weight the correlation windows before the stereo correspondence search. We shall illustrate this process with experiments from [Neu]. Let q be a point in I. Consider $L_q \subset I$ the subimage where the weight is learned. For each $p \in L_q$ the weight due to color similarity and proximity are computed:

color similarity: We consider the color distance

$$d_c(u_q, u_p) = \left((R_u(q) - R_u(p))^2 + (G_u(q) - G_u(p))^2 + (B_u(q) - B_u(p))^2\right)^{1/2} ,$$

where R_u,G_u and B_u are the red, green and blue channels of u. Then the weight corresponding to the color similarity between p and q is

$$w_c(p, q) = exp\left(-\frac{d_c(u_q, u_p)^2}{h_1^2}\right) .$$

proximity: We consider the Euclidean distance between the points positions in the image plane

$$d(q, p) = \left((q_1 - p_1)^2 + (q_2 - p_2)^2\right)^{1/2} ,$$

where $p = (p_1, p_2)$ and $q = (q_1, q_2)$. Then the weight corresponding to proximity is

$$w_d(p, q) = exp\left(-\frac{d(q, p)^2}{h_2^2}\right) .$$

Therefore, the total associated weight between the two points q and p is

$$W(p, q) = \frac{1}{Z_q}\, w_c(p, q)w_d(p, q) = \frac{1}{Z_q}\, exp\left(-\left(\frac{d_c(u_q, u_p)^2}{h_1^2} + \frac{d(q, p)^2}{h_2^2}\right)\right) ,$$

where Z_q is the normalizing factor $Z_q = \sum_{p\in L_q} w_c(p, q)w_d(p, q)$. The interpolated disparity map μ_I is computed via an iterative schema

$$\mu_I(q, k) = \sum_{p\in L_q} W(p, q)\mu_I(p, k - 1) ,$$

where k is the current iteration and the initialization $\mu_I(\cdot, 0) = \mu_M(\cdot)$.

Figures 10 and 11 show the interpolated Middlebury results (100% density). The experiments demonstrate that, starting from a disparity map which is very sparse near image boundaries, the bilateral diffusion process can recover a reasonable depth map.

0.2.2 Bilateral filter for enhancing the resolution of low-quality range images

In [YYDN07], the authors have proposed a post-processing step to enhance the resolution of low-quality range images. Contrary to intensity images, each pixel of a range image expresses the distance between a known reference

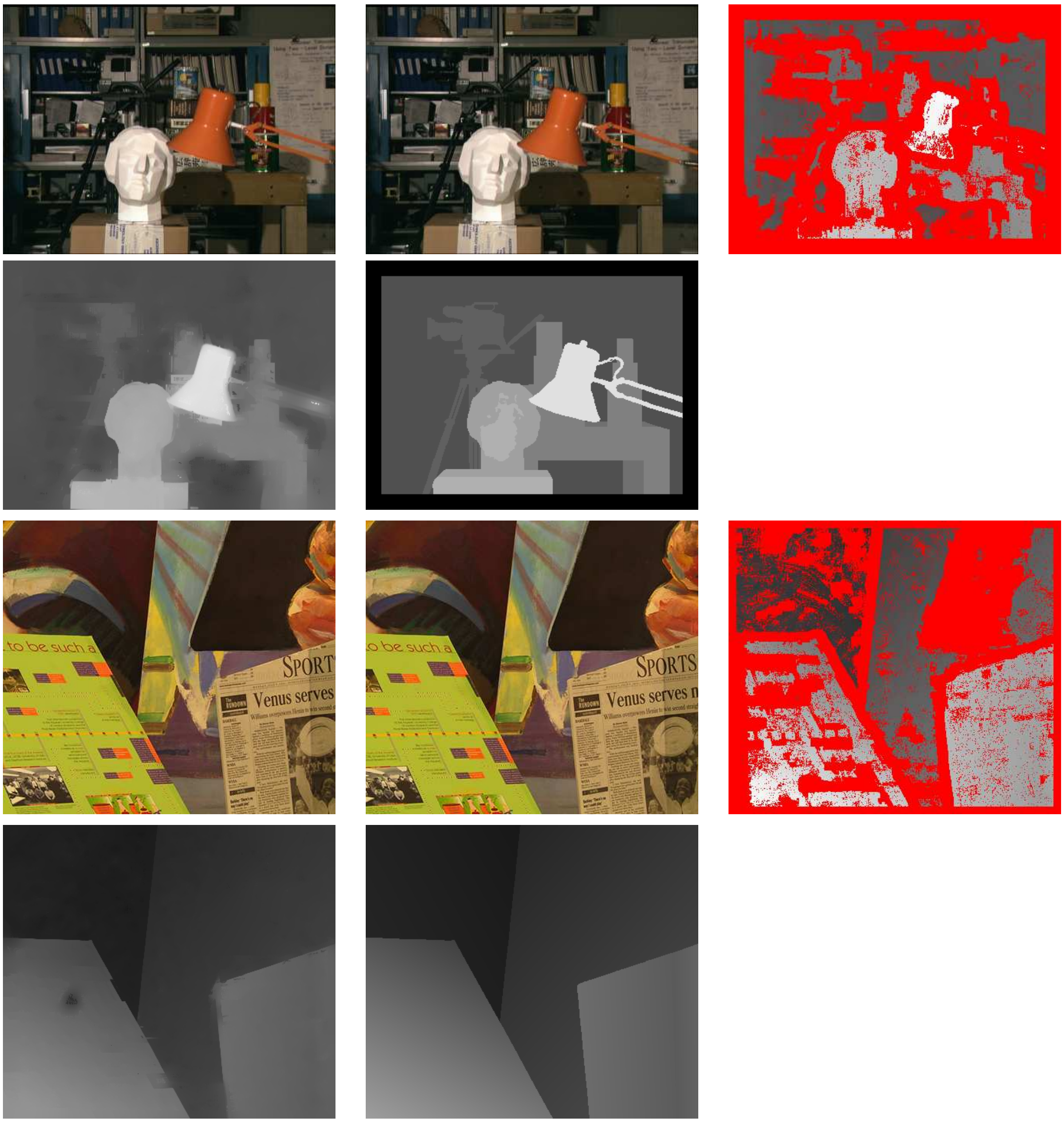


Figure 10: Tsukuba and Venus results. For each couple of images: stereo pair of images, output of a sparse algorithm retaining only sure points, points (in red in the .pdf file) are the rejected correspondences, interpolated version of these results, ground truth and signed error between the interpolated image and the ground truth.

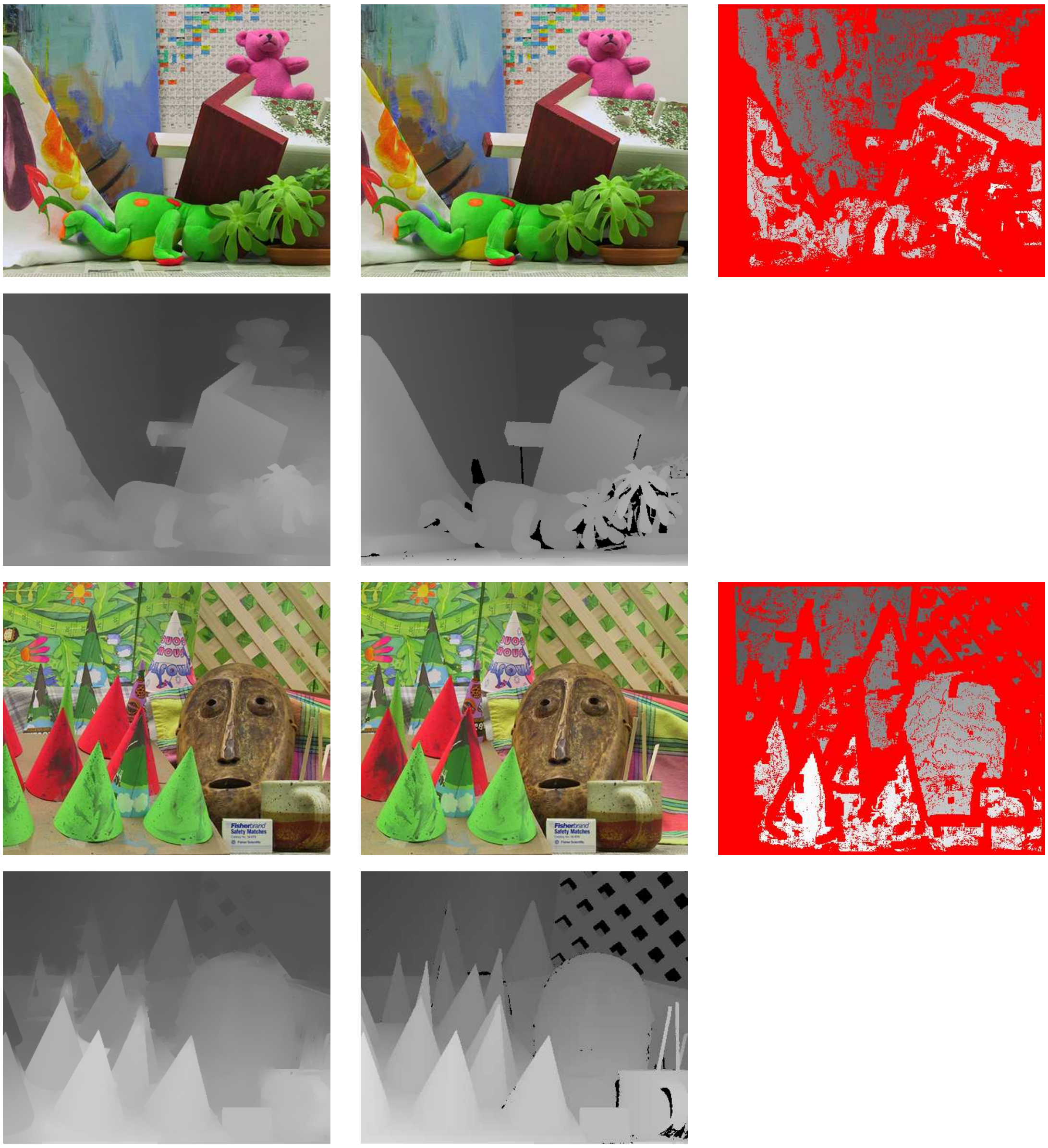


Figure 11: Teddy and Cones results. For each couple of images: stereo pair of images, output of a sparse algorithm retaining only sure points, points (in red in the .pdf file) are the rejected correspondences, interpolated version of these results, ground truth and signed error between the interpolated image and the ground truth.

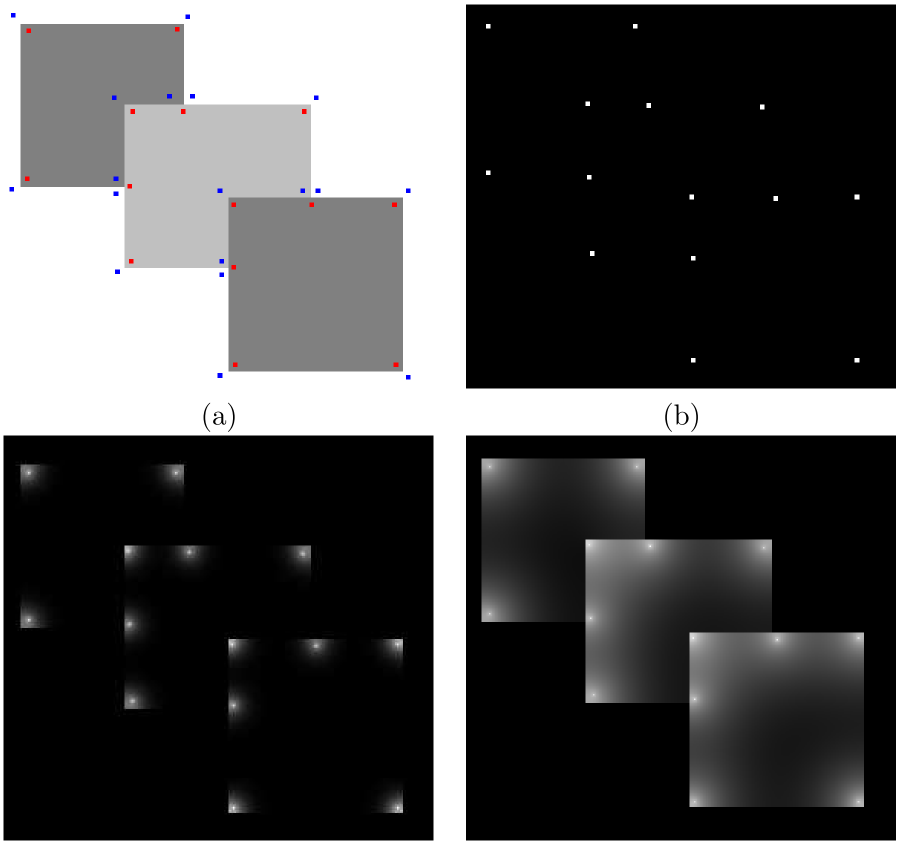
frame and a visible point in the scene. Range images are acquired by range sensors that, when acquired at video rate, are either very expensive or very limited in terms of resolution. To increase the resolution of low-quality range images acquired at video rate, the authors of [YYDN07] have proposed a post-processing step relying on an iterated bilateral filter. The filter diffuses the depth values of the low-quality range image, steering the diffusion by the color information provided by a registered high-quality camera image. The input range image z_0 is first up-sampled to the same size of the camera image and then used as the initial depth map hypothesis according to which an initial cost volume is built, c_0. An iterative process is then applied that consists in first performing a bilateral filter on each slide of the current cost volume c_i and then in generating a new depth map hypothesis z_{i+1} based on the filtered cost volume. At each filtering step, an aggregated cost volume is computed for each pixel based on the cost volume of the pixels in its neighborhood, which is defined based on the color similarity. The new depth map hypothesis z_{i+1} is then obtained by first selecting the depth hypothesis having the minimal cost volume and then by applying a sub-pixel estimation using a quadratic polynomial interpolation.

0.2.3 Bilateral filter for the global integration of local depth information

In [DMS08], an iterated neighborhood filter has been proposed as a depth diffusion mechanism to globally integrate a set of initial local depth hypothesis derived from different monocular depth cues.

With this strategy, the occlusion boundaries and the relative distances from the viewpoint of depicted objects are simultaneously recovered from local depth information without the need of any explicit segmentation and possible conflicting depth relationships are automatically solved by the diffusion process itself.

Once monocular depth cues are detected, each region involved in a depth relationship is marked by one or few points, called *source points* (see Figure 12 (a)). Source points marking the regions closer to the viewpoint are called Foreground Source Points (FSPs), whereas source points marking the regions more distant to the viewpoint are called Background Source Points (BSPs). The depth image z is initialized by assigning value 1 to FSPs and value 0 to BSPs. The rest of the image is initialized with value zero (see Figure 12 (b)). The diffusion process is applied to the depth image z by using the gradient of the original image u rather than that of the depth image to define the neighborhood. By diffusing the depth information using the gray level (color) gradient of the original image instead of that of the depth map itself, the edge information is incorporated into the depth map, assuring that discontinuities in depth are consistent with gray level (color) discontinuities.



(a) (b)

(c) (d)

Figure 12: Example of depth diffusion by using equation (3). (a) Gray level image, where BSPs and FSPs are marked in blue and red respectively. (b) Depth image, where points corresponding to FSPs are initialized with a positive value (marked in white) and the rest of the image with value zero. (c) and (d) Depth images after an increasing number of iterations of the DDF.

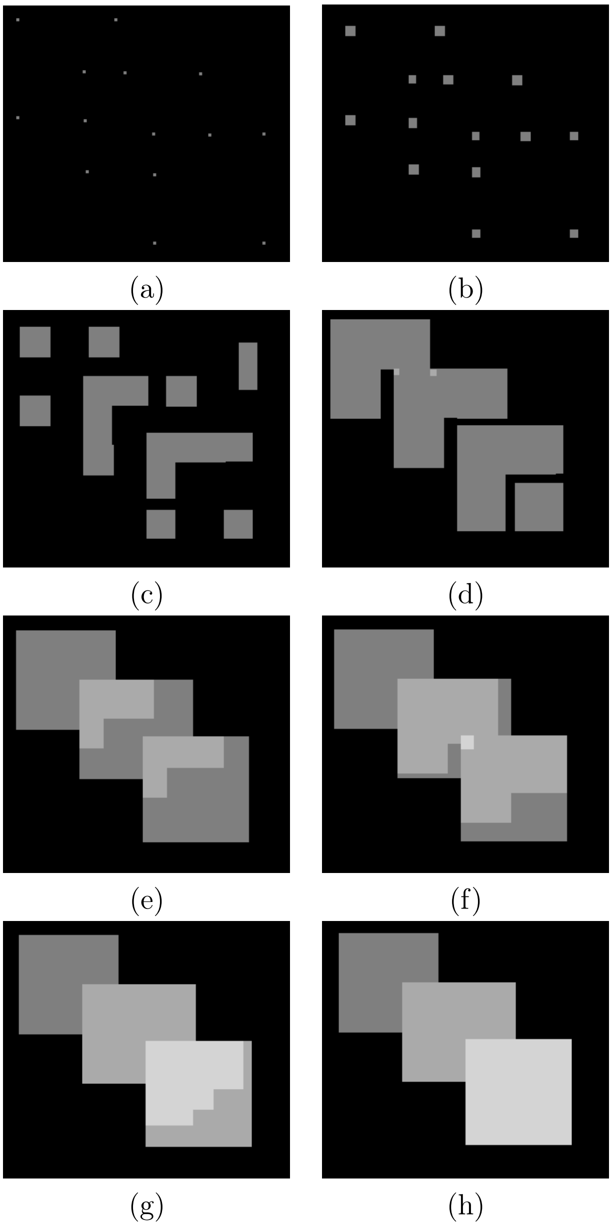


Figure 13: Example of depth diffusion by using equation 5. (a) Depth image, where FSPs have been initialized with a positive value (marked in gray) and the rest of the image with value zero. (b), (c), (d), (e), (f), and (g) depth images corresponding to an increasing number of iterations. After each iteration, the depth difference between corresponding FSPs and BSPs is forced to be at least equal to the initial depth difference Δ, by adding Δ to FSPs when the difference between corresponding FSPs and BSPs is less than Δ.(h) Final depth image.

The depth diffusion filter (DDF) proposed in [DMS08] is

$$DDF_{h,\rho}z(x) = \frac{1}{C(x)} \int_{B_\rho(x)} z(y) e^{\frac{-|u(x)-u(y)|^2}{h^2}} dy, \tag{3}$$

where $B_\rho(x)$ is a ball of center x and radius ρ, h is the filtering parameter which controls the decay of the exponential function, and

$$C(x) = \int_{B_\rho(x)} e^{\frac{-|u(x)-u(y)|^2}{h^2}} dy \tag{4}$$

is the normalization factor. We set $\rho = 1$ and $h = 10$.

Equation (3) is applied iteratively until the stability is attained. After each iteration, the values of FSPs and BSPs are modified in order to hold at least the initial depth gradient. In practice, when the difference between the values of a FSP and the corresponding BSP become less than 1, the value 1 is added to the value of the FSP. This constraint corresponds to Neumann internal boundary conditions which are understood as a pre-specified jump on the $c\frac{Dz}{Dn}$ as the boundary is crossed, where c is a positive constant and n is the normal to the boundary. This allows one to handle simple sorting when objects are located in multiple layers. Figure 12 is an example of the diffusion through the DDF. As can be sensed looking at Figures 12 (c) and (d), by taking an average of the depth values in a neighborhood, a very large number of iterations would be needed to attain the stability. To make the diffusion process faster, the following equation is used to initialize the process

$$DDF_{h,\rho}z(x) = \sup_{B_\rho(x)} z(y) e^{\frac{-|u(x)-u(y)|^2}{h^2}} dy, \tag{5}$$

while equation (3) is used only in the last iterations (see Figure 13).

In the case of occlusion there is also a depth order between the two regions separated respectively by the stem of the T. However, occlusion does not carry any information about the partial order between the objects respectively in partial occlusion and the background. This depth order can be inferred by other cues, such as convexity. When information about this partial order is present, the depth gradient between one of the BSPs and the FSPs increases. This is the reason for which force source points are forced to hold "at least" the initial depth gradient.

Experimental results on real images (see Fig. 14) have proved that this simple formulation turns out to be very effective for the integration of several monocular depth cue such as occlusion, transparency, convexity, visual completion (both a-modal and modal) and self-occlusion. In particular, contradictory information given by conflicting depth cues is dealt with correctly by the bilateral diffusion mechanism, which permits two regions to invert harmoniously their depths, in full agreement with the phenomenology.

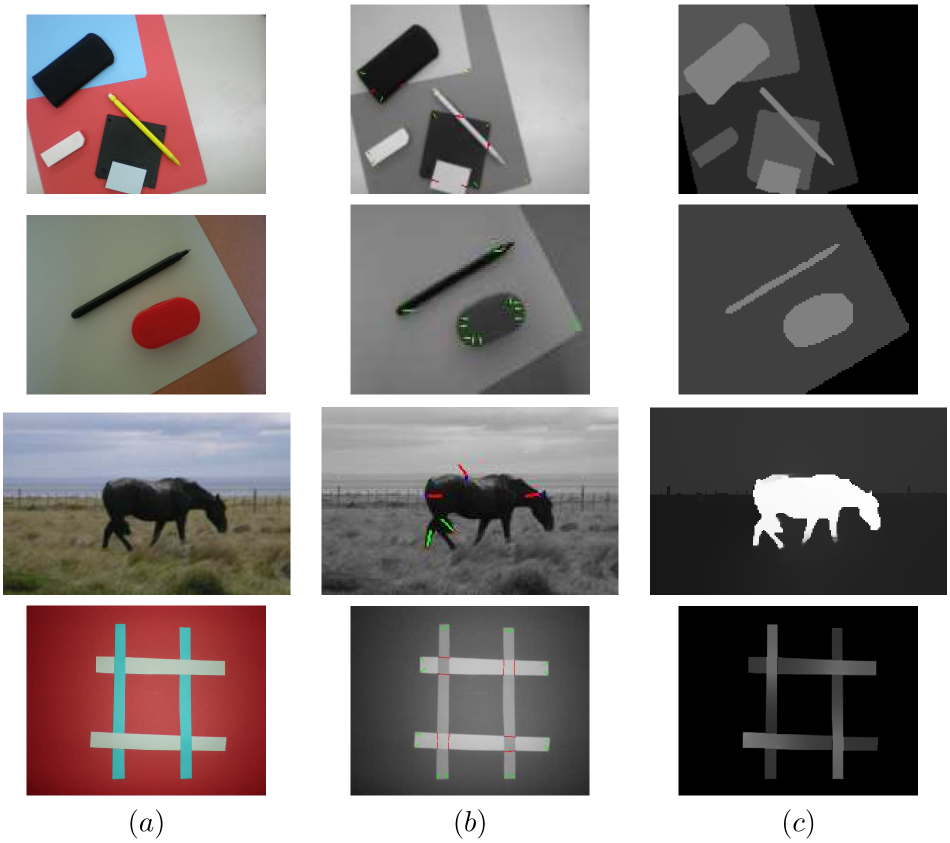


Figure 14: (*a*) Original image. (*b*) Local depth cues are represented through vectors that point to the region closer to the viewpoint. (*c*) Depth image.

For each experiment on Fig. 14 are shown four images: the original image (Fig. 14(a)); the image where the initial depth gradient at depth cue points is represented through vectors pointing to the region closer to the viewpoint (red vectors arise from T-junctions, green vectors arise from local convexity and each of them represents the point having the biggest curvature value of the connected components obtained by thresholding the curvature) (see Fig.14 (b)); the depth image obtained by performing the bilateral diffusion method (see 14 (c)). The depth map is rendered through gray level values (high values indicate regions that are close to the camera). On the first and the second rows of Figure 14 there are examples of indoor scenes, for which a proper solution is obtained. Instead, on the third row there is an example of outdoor scene involving a conflict. The T-junction detected on the back of the horse is due to a reflectance discontinuity and its local depth interpretation is incorrect. However, on the depth map, the shape of the horse appear clearly on the foreground since the diffusion process allowed to overcome the local inconsistency. In the last row there is an example involving self-occlusion: occluding contours have different depth relationships at different points along its continuum. However, the bilateral diffusion method performs well also in this ambiguous situation.

Acknowledgements The David raw point set is courtesy of the the *Digital Michelangelo Project, Stanford University*. Fragment '31u' of the *Stanford Fragment Urbis Romae, Stanford University*, the Screw Nut point set is provided by the AIM@SHAPE repository and is courtesy of Laurent Saboret, INRIA. Research partially financed by Institut Farman, ENS Cachan, the Centre National d'Etudes Spatiales (MISS Project), the European Research Council (advanced grant Twelve Labours) and the Office of Naval research (grant N00014-97-1-0839).

Bibliography

[AGDL09] Andrew Adams, Natasha Gelfand, Jennifer Dolson, and Marc Levoy, *Gaussian kd-trees for fast high-dimensional filtering*, ACM Trans. Graph. **28** (2009), no. 3, 1–12.

[BCM05] A. Buades, B. Coll, and J. M. Morel, *A review of image denoising algorithms, with a new one*, Multiscale Modeling & Simulation **4** (2005), no. 2, 490–530.

[CT05] Prasun Choudhury and Jack Tumblin, *The trilateral filter for high contrast images and meshes*, SIGGRAPH '05: ACM SIGGRAPH 2005 Courses (New York, NY, USA), ACM, 2005, p. 5.

[DMMSL09] J. Digne, J.M. Morel, C. Mehdi-Souzani, and C. Lartigue, *Scale space meshing of raw data point sets*, preprint CMLA - ENS Cachan, October 2009.

[DMS08] M. Dimiccoli, J.M. Morel, and P. Salembier, *Monocular depth by nonlinear diffusion.*, Indian Conference on Computer Vision, Graphics and Image Processing, (ICVGIP) (Bhubaneswar (India)), December 2008.

[Dun73] J. C. Dunn, *A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters*, Journal of Cybernetics **3** (1973), no. 3, 32–57.

[FC09] G. Facciolo and V. Caselles, *Geodesic neighborhoods for piecewise affine interpolation of sparse data*, International Conference on Image Processing, 2009.

[FDCO03] Shachar Fleishman, Iddo Drori, and Daniel Cohen-Or, *Bilateral mesh denoising*, ACM Trans. Graph. **22** (2003), no. 3, 950–953.

[FLA$^+$06] G. Facciolo, F. Lecumberry, A. Almansa, A. Pardo, V. Caselles, and B. Rougé, *Constrained anisotropic diffusion and some applications*, British Machine Vision Conference, 2006.

[GP87] E. Gamble and T. Poggio, *Visual integration and detection of discontinuities: the key role of intensitiy edges.*, 1987.

[HW05] Li Hou, Qibin andBai and Yangsheng Wang, *Mesh smoothing via adaptive bilateral filtering*, ch. Computational Science - ICCS 2005, pp. 273–280, Springer, 2005.

[JDD03] Thouis R. Jones, Frédo Durand, and Mathieu Desbrun, *Non-iterative, feature-preserving mesh smoothing*, SIGGRAPH '03: ACM SIGGRAPH 2003 Papers (New York, NY, USA), ACM, 2003, pp. 943–949.

[KS91] P.J. Kellman and T.F. Shipley, *Visual interpolation in object perception*, Current Directions in Psychological Science **1** (1991), no. 6, 193–199.

[LDZPD08] F. Lafarge, X. Descombes, J. Zerubia, and M. Pierrot-Deseilligny, *Automatic building extraction from dems using an object approach and application to the 3d-city modeling*, Journal of Photogrammetry and Remote Sensing **63** (2008), no. 3, 365–381.

[Lee83] Jong-Sen Lee, *Digital image smoothing and the sigma filter*, Computer Vision, Graphics, and Image Processing **24** (1983), no. 2, 255–269.

[LYY$^+$05] Yu-Shen Liu, Pi-Qiang Yu, Jun-Hai Yong, Hui Zhang, and Jia-Guang Sun, *Bilateral filter for meshes using new predictor*, vol. 3314/2005, ch. Lecture Notes in Computer Science, Computational and Information Science, pp. 1093–1099, Springer/Heidelberg, 2005.

[MF04] A. Miropolsky and A. Fischer, *Reconstruction with 3d geometric bilateral filter*, SM '04: Proceedings of the ninth ACM symposium on Solid modeling and applications (Aire-la-Ville, Switzerland, Switzerland), Eurographics Association, 2004, pp. 225–229.

[Neu] *Reliability and accuracy in stereovision. application to aerial and satellite high resolution images.*

[OBS02] Yutaka Ohtake, Alexander G. Belyaev, and Hans-Peter Seidel, *Mesh smoothing by adaptive and anisotropic gaussian filter applied to mesh normals*, VMV, 2002, pp. 203–210.

[SS] D. Scharstein and R. Szelisk, *Middlebury stereo vision research page.*

[TM98] C. Tomasi and R. Manduchi, *Bilateral filtering for gray and color images*, ICCV '98: Proceedings of the Sixth International

Conference on Computer Vision (Washington, DC, USA), IEEE Computer Society, 1998, p. 839.

[Wan06] Charlie C. L. Wang, *Bilateral recovering of sharp edges on feature-insensitive sampled meshes*, IEEE Trans. on Visualization and Computer Graphics **12** (2006), no. 4, 629–639.

[WLG+06] L. Wang, M. Liao, M. Gong, R. Yang, and D. Nistér, *High-quality Real-time Stereo using Adaptive Cost Aggregation and Dynamic Programming.*, Third International Symposium on 3D Data Processing, Visualization and Transmission, 3DPVT, 2006.

[WYC06] Lihui Wang, Baozong Yuan, and Jing Chen, *Robust fuzzy c-means and bilateral point clouds denoising*, 2006 8th International Conference on Signal Processing, vol. 2, 2006, pp. 16–20.

[WZZY08] Ren-Fang Wang, San-Yuan Zhang, Yin Zhang, and Xiu-Zi Ye, *Similarity-based denoising of point-sampled surfaces*, Journal of Zhejiang University **9** (2008), no. 6, 807–815.

[Yar85] L. P Yaroslavsky, *Digital picture processing. an introduction.*, Springer Seriesin Information Sciences, vol. 9, Springer-Verlag, Berlin - Heidelberg, 1985.

[YBS06] S. Yoshizawa, A. Belyaev, and H.-P. Seidel, *Smoothing by example: Mesh denoising by averaging with similarity-based weights*, SMI '06: Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006 (Washington, DC, USA), IEEE Computer Society, 2006, p. 9.

[YC04] J. Yin and J.R. Cooperstock, *Improving Depth Maps by Nonlinear Diffusion*, In Proc. of 12th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG), 2004, pp. 1–8.

[YK06] K.-J. Yoon and S. Kweon, *Adaptive support-weight approach for correspondence search.*, IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI) **28** (2006), no. 4, 650–656.

[Yoo06] S. Yoon, K.-J.and Kweon, *Adaptive support-weight approach for correspondence search*, IEEE Transactions on Pattern Analysis and Machine Intelligence **28** (2006), no. 4, 650–656.

[YWY+06] Q. Yang, L. Wang, R. Yang, H. Stewénius, and D. Nistér, *Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling*, IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI) **31** (2006), no. 3, 1–13.

[YYDN07] Q. Yang, R. Yang, J. Davis, and D. Nistér, *Spatial-Depth Super Resolution for Range Images.*, International Conference on Computer Vision and Pattern Recognition, CVPR, 2007.