DOI: 10.2352/ISSN.2470-1173.2016.21.3DIPM-401

Efficient Filling of Disparity Holes Using Resolution Decoupling

Alexey Supikov, Maha El Choubassi, Oscar Nestares; Intel Corporation; Santa Clara, California/USA

Abstract

We propose a novel highly efficient method for filling disparity holes, regions where disparity estimation fails to produce correct result, with the most plausible values. While the filling values may not exactly match the missing disparities, the filled disparity map has cohesive and smooth areas. Such disparity map enables many applications, such as refocusing and layer effects and other new 3D photography apps, to overcome artifacts due to holes and be visually pleasant for the user. To solve for the filling disparities, we incorporate the visual saliency in our model and decouple the solution complexity from the resolution of the original disparity map. Hence, our technique strikes a good balance between perceptual quality and computational efficiency. Overall, our method produces high quality results fulfilling or exceeding the requirements of practical applications that use depth. Moreover, it is fast and hence adequate to run on the ubiquitous mobile platforms.

Introduction

There are many methods [1] to estimate depth/disparity using 2 or more images of a scene from different camera positions. Fundamentally, all methods compare image regions using a selected error measure. Such comparison might fail because of occlusion or mismatch in the large areas indistinguishable by the error measure, e.g., areas of uniform color (wall in Figure 1), or with repetitive pattern. There are methods to detect and label the areas of occlusion or mismatch but the actual value of depth/disparity remains undetermined in many regions of the map, resulting in holes (Figure 2). Disparity holes result in unacceptable visual experience in digital photography applications such as layer effects, parallax viewing, and refocus. However, many such applications do not require knowledge of the exact disparity values in the holes. They only demand a disparity map cohesive enough for plausible perceptual experience. For example, in layers effects application, initially the image appears in grayscale. As the user moves a slider, parts of the image come in color gradually, closer layers before further ones. Filling the holes with cohesive and smooth disparity guesses fixes the annoying artifacts. There are many attempts for such filling in literature [2, 3, 4, 5, 6, 7, 8, 9], usually based on propagation of known disparity values into the interior of undefined region. Recently, Lu et al [10] suggest a new method based on low-rank matrix completion to simultaneously enhance noisy images and depth maps and the depth maps may also have missing values. Overall, these methods either

a) **Use simple disparity propagation techniques but lack quality:** no smooth transitions from boundaries to the interior, results look patchy or grainy; or

b) **Produce high quality results but are computationally complex:** such methods are usually based on solving a PDE system, e.g. diffusion equation, or an optimization problem.

The reason that methods b) are computationally expensive is the huge dimensionality of the problem which equals or exceeds the pixel count in the disparity holes. The size of the problem is large because the grid resolution for the solver is the same as the disparity map resolution.

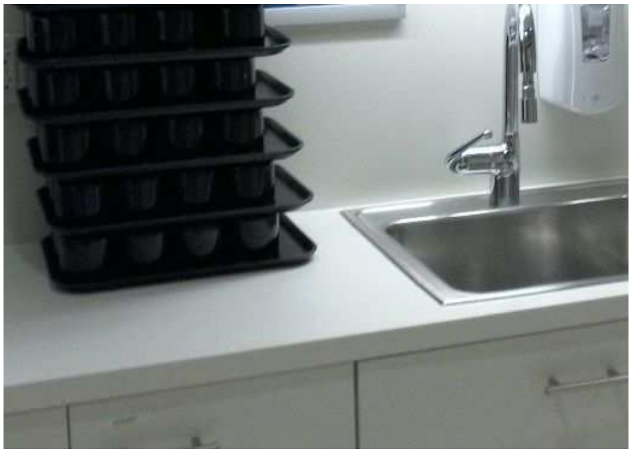


Figure 1. *Reference image for disparity computation*

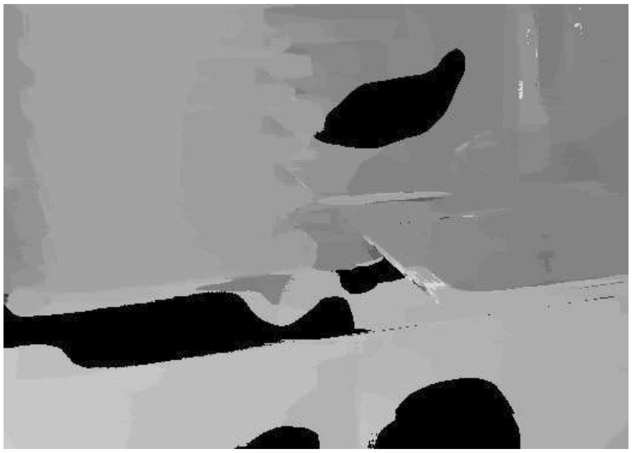


Figure 2. *Input: Original disparity map D(x,y) (black areas are disparity holes)*

Problem Statement

Our objective is to design a hole filling algorithm with the following requirements:

- it has visually satisfactory results in digital photography applications, and
- it is computationally efficient to run even on mobile devices.

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.21.3DIPM-401

For this purpose, we propose to decouple the resolution of the solver grid from the resolution of the disparity map. Such decoupling allows for solving a smaller problem on a grid with resolution lower than the resolution of original disparity map. At the same time, the method maintains the full resolution of the original disparity map, hence preserving all the details in the regions where disparity was estimated correctly. Our novel method approaches the smoothness and quality of the expensive methods at substantially lower computational cost.

Advantages of Our Approach over Existing Methods

Our suggested method is better than existing solutions [2, 3, 4, 5, 6, 7, 8, 9] in the following aspects:

- **Handling smoothness.** Our method handles correctly areas where depth changes smoothly. It produces smooth and visually plausible results unlike simpler extrapolation or propagation methods like [2, 3, 4, 5] that take a single value and propagate it across a large area. For example, our method can handle arbitrarily oriented large uniform color wall, filling its region with gradually changing disparity.
- **High performance (up to** 10 **fps on Bay Trail tablet CPU and** 38 **fps on i**7 **desktop).** Unlike methods based on dense diffusion or optimization [6, 7], our method is at least 2 orders of magnitudes faster for most of the cases. Despite the potential speedup of multigrid diffusion over dense diffusion [8, 9], these methods remain very hard to parallelize.
- **Large holes.** Our method addresses large mismatch areas of the same color or texture where the error measure has no single minimum. Other methods [4] target only disocclusion holes, usually smaller with clear discontinuity in depth and color space, and hence allowing use of 1 or 2 values to fill the hole.

Algorithm

The input to our algorithm consists of the initial estimated disparity map $D(x,y)$ (Figure 2) and the labeled areas of mismatch or occlusion\dis-occlusion indicating that the region is a hole, e.g. labeled by special reserved disparity values. For example in Figure 2, holes are marked by disparity value zero (black). Here are the steps of our algorithm:

1. First, we create holes mask H using the holes labels. The holes mask has value zero outside holes and non-zero in holes regions (Figure 3).
2. Next, we detect outer holes boundary contour for each connected non-zero region in the holes mask H. The set of contours C circumscribing the holes is the result of the operation. Note that we know the disparity values $D(x,y)$ on these contours, unlike pixels inside the holes. Then, we generate mask O with outer contours of holes (Figure 3). In mask bitmap O, all pixel values equal zero, except for the pixels on the outer parts of the contours circumscribing the holes, which are set to a non-zero value such as 1.
3. We select regular grid $G(i,j) = (x_i, y_j)$ that has lower resolution than the resolution of $D(x,y)$ (the typical grid cell size is 2-20 pixels depending on the original resolution and data) (Figure 4).

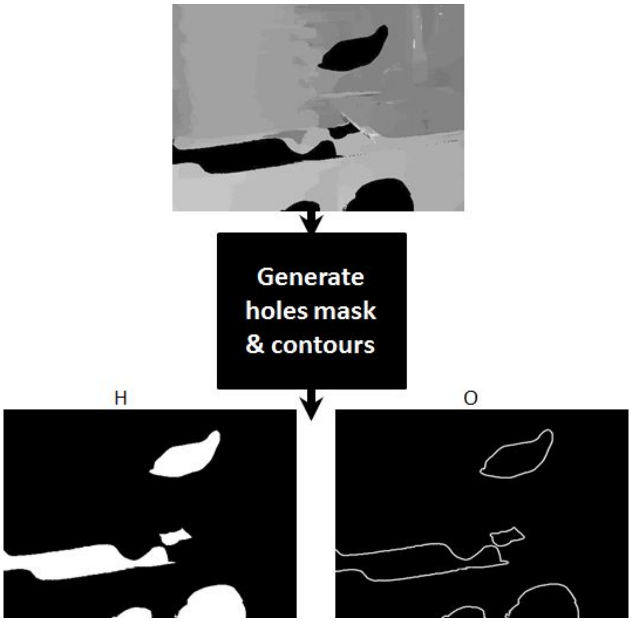


Figure 3. *Get the holes mask $H(x,y)$ and the outer contour mask $O(x,y)$ (outer contours of hole boundaries), both in full resolution.*

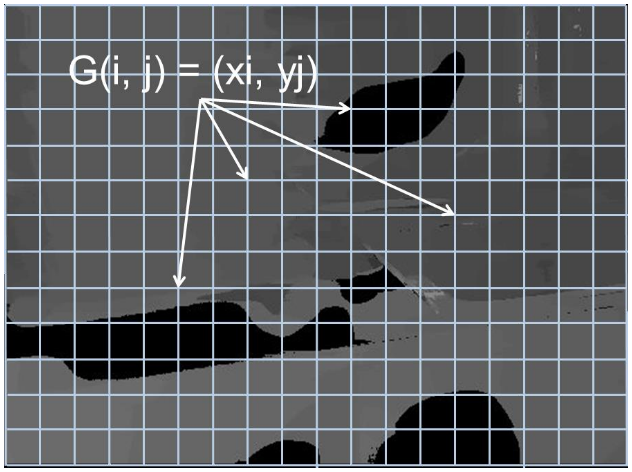


Figure 4. *Low resolution regular grid $G(i,j)$ on the disparity map $D(x,y)$.*

4. We build the sparse quadratic optimization problem in (1), where grid disparities d_{ij} at $G(i,j)$ are the unknowns. Solving (1) results in filtered disparity values at the vertices of the lower resolution grid.

$$E = E_d + \alpha E_s \tag{1}$$

Similarly to [11, 12], the quadratic cost E has 2 terms, the data term E_d and the smoothness term E_s. The data term constrains d_{ij} to be close to the known $D(x,y)$ in the holes' contours $O(x,y)$. The smoothness term is for regularizing d_{ij} to be smooth and continuous. To fill holes with plausible disparity values, we designed our problem in this way:

a. we know the original disparity $D(x,y)$ except at holes. We only want to solve for disparities inside the hole or very close to its outer boundary. Therefore, an unknown d_{ij} is included in (1) only if $G(i,j)$ is inside the

DOI: 10.2352/ISSN.2470-1173.2016.21.3DIPM-401

hole or if it is outside but very close/adjacent to some outer hole boundary (Figure 5, red circling). For such nearby outside nodes, there must be at least one point (x,y) in the vicinity of $G(i,j)$ such that $O(x,y)$ is not zero. For example, the neighborhood of $G(i,j)$ can be the 4 adjacent grid cells (quads).

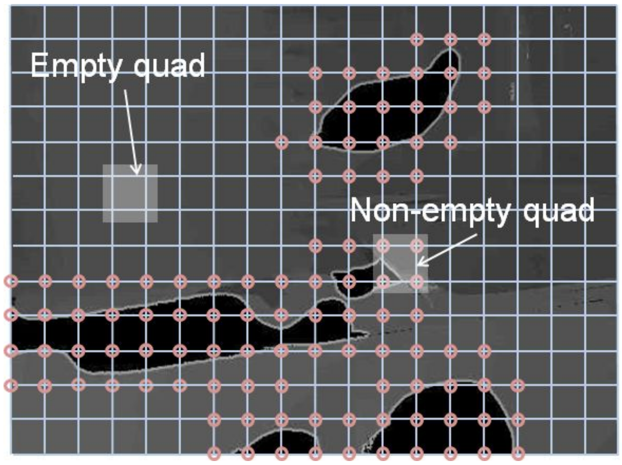


Figure 5. *The nodes given to the sparse solver (circled with red). Only non-empty quads contribute to the solution.*

b. only disparities from a hole boundary are used to build the data term E_d. Disparities in other regions are ignored because disparities at further points are more likely irrelevant to the hole unknown disparities. A point (x,y) from a vicinity of vertex $G(i,j)$ participates in the correspondent coefficient for $G(i,j)$ only if (x,y) is on the outer boundary of a hole, i.e., $O(x,y)$ is not zero . Figure 6 shows the participating regions around holes.

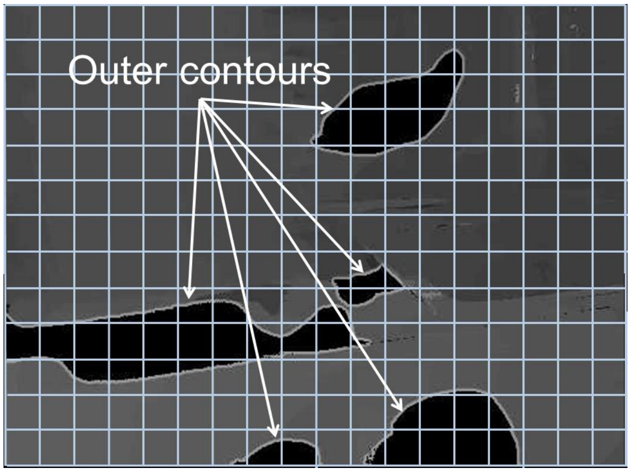


Figure 6. *Outer contours around holes highlighted in light grey. Only these regions participate in disparity filtering.*

Therefore, we formulate the data term and smoothness term as shown below. We define the data term E_d as

$$E_d = \sum_{n=1}^{N} \sum_{(x,y)\in S_n} O(x,y)(\mathbf{w_{(x,y)}}^t \mathbf{d_n} - D(x,y))^2. \quad (2)$$

S_n is a cell of the grid, which has N total cells. Each S_n has 4 vertices and $\mathbf{d_n} = [d_{ij} \;\; d_{(i+1)j} \;\; d_{i(j+1)} \;\; d_{(i+1)(j+1)}]^t$ are the unknown disparities at S_n vertices. As in (2), only cells S_n that intersect with a holes contour contribute to E_d. Moreover, we express every disparity value $D(x,y)$ in S_n that also belongs to hole contour ($O(x,y) \neq 0$) as linear combination of disparities in $\mathbf{d}_n$, i.e., $\mathbf{w_{(x,y)}}^t \mathbf{d_n}$. We then include the squared estimation error in E_d. E_d is a sum over the holes contours of the squared error between original disparities $D(x,y)$ and their estimate, $\mathbf{w_{(x,y)}}^t \mathbf{d_n}$.

The smoothness term E_s is defined as

$$\begin{aligned} E_s &= \sum_{n=1}^{N} \mathrm{Sa}_n\{(d_{ij} - d_{(i+1)j})^2 + (d_{ij} - d_{i(j+1)})^2 \\ &+ (d_{(i+1)j} - d_{(i+1)(j+1)})^2 \\ &+ (d_{i(j+1)} - d_{(i+1)(j+1)})^2\}. \end{aligned} \quad (3)$$

In E_s, for each cell S_n, we sum the squared difference between the unknown disparities d_{ij} at adjacent vertices of S_n. E_s is weighted by α, which provides a global handle on the desired smoothness in the solution. We also weight each term of E_s by the saliency Sa_n of grid cell S_n, a local handle on the amount of distortion acceptable at each cell. Intuitively, errors in d_{ij} are more visually tolerable on plain than detailed areas and the saliency weighted error models the perceived distortion. There are many options to compute image saliency [13]. We model saliency as the sum of multiresolution variance.

5. We solve (1) for d_{ij} by converting it into a linear problem and efficiently using a sparse solver. Our current prototype uses linear PCG with Jacobi preconditioner but any direct technique works as well. A resulting solution is a set of values for d_{ij} included in the system at step 4, the rest of the filtered disparities are set to 0.
6. Then, we upsample the filtered disparity values d_{ij} from the solver grid resolution to full resolution of original disparity map using any suitable interpolation technique, and write the results to $F(x,y)$ (Figure 7). In the current implementation $F(x,y)$ is the bilinear interpolation of values at vertices of a grid cell containing point (x,y).

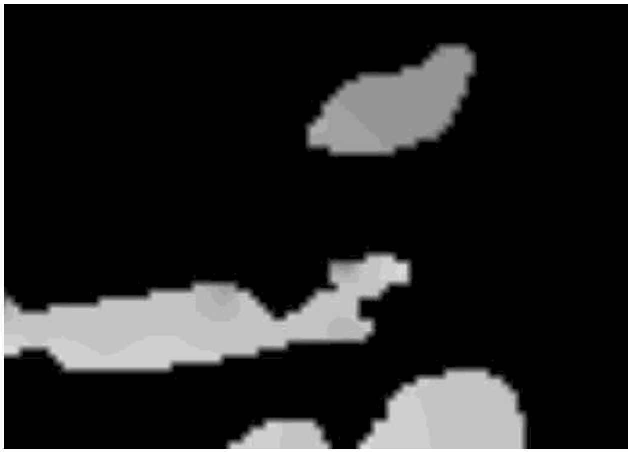


Figure 7. *The map $F(x,y)$ of the interpolated disparities produced by non-linear filtering*

7. In the final step, we combine the original disparity map D and map F using holes mask H as in Figure 8. The $D(x,y)$ is set to $F(x,y)$ if $H(x,y)$ is non-zero, thus filling only holes in D with values from F.The resulting improved disparity is shown on Figure 10.

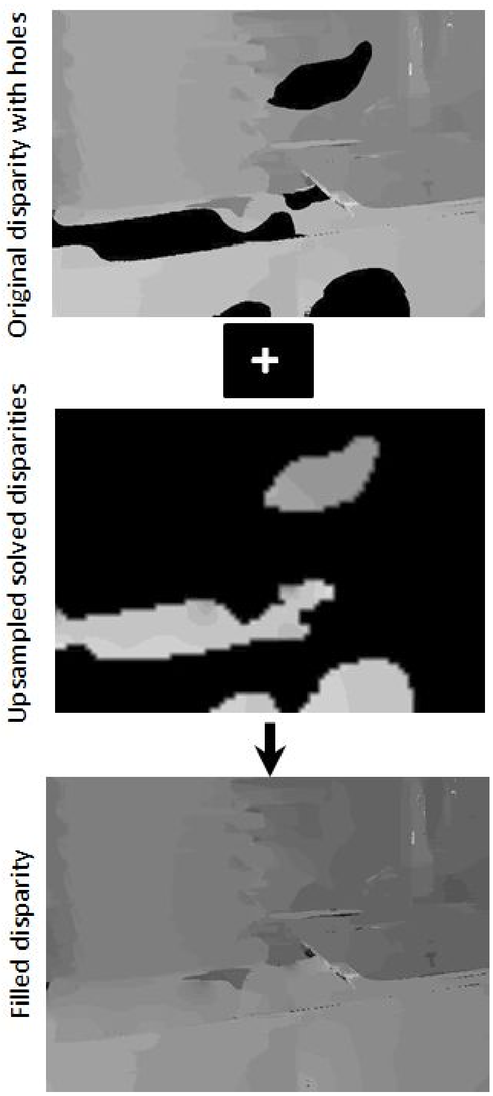


Figure 8. Final step to get the filled disparity map.

Finally, we illustrate an overview of the solver system in the diagram of Figure 9.

Results

Algorithm Performance

Our algorithm performance results are summarized in the Table below. For 1280×720 images and disparity maps, our method runs at 10 fps, on Intel® Bay Trail tablet with 2 GB RAM and 1.46 GHz Atom Quad-core CPU. On a desktop, it is approximately 4 times faster and runs at 38 fps on 3.5 GHz Intel® i7 desktop with 32 GB of RAM.

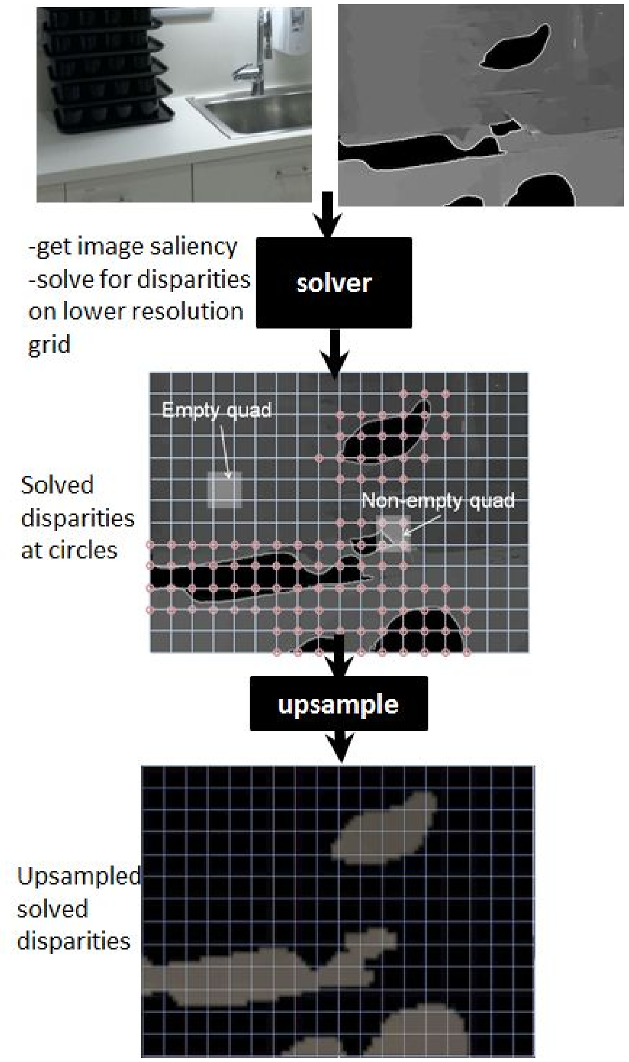

-get image saliency
-solve for disparities on lower resolution grid
solver
Empty quad
Non-empty quad
Solved disparities at circles
upsample
Upsampled solved disparities

Figure 9. Solver's system diagram. Input: original disparity map with holes and the image. Output: solved disparities in the holes at the grid vertices.

Performance

Intel® CPU	Clock	RAM	Time per frame
Atom Quadcore Tablet	1.46 GHz	2 GB	100 ms
i7 desktop	3.5 GHz	32 GB	26 ms

Enhancing Photography Applications Experience

We ran our hole filling algorithm on 40 images / disparity maps pairs. The holes are smoothly filled, e.g., see Figures 10 and 11. As emphasized earlier, in addition to running efficiently, the objective of our hole filling algorithm is to enhance the visual experience of photography applications such as layer effects. For this purpose, we verify the effectiveness of our algorithm by applying the layers effects to the 40 images before and after filling

DOI: 10.2352/ISSN.2470-1173.2016.21.3DIPM-401

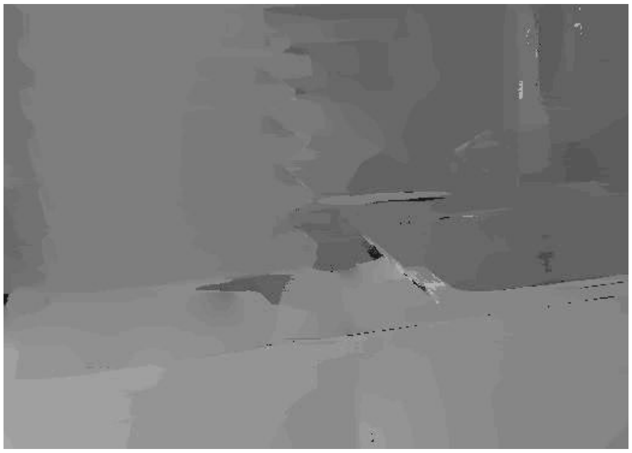
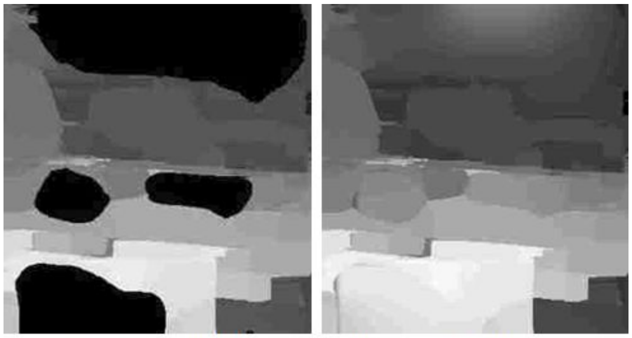


Figure 10. *Final output: disparity processed using our method. Holes/black areas are smoothly filled.*

the disparity. Figures 11 and 12 show an example. First, the raw disparity suffering from holes and the disparity filled with our algorithm are in Figure 11. Next, Figure 12 shows the layers effects app with both raw and filled disparity maps. At the beginning, all the image appears in grayscale. Gradually, closer pixels, i.e., with higher disparity, become colored. For example, the pixels on the chair and the table should get colorful before the pixels on the person and on the wall. The artifacts resulting from holes are obvious in the left column of Figure 12: the 2 spots on the table remain gray, even when the person is already in color. These spots are the holes in the raw disparity map, and since the holes have value zero, they are wrongly assumed far. Once the holes are filled by our algorithm, these artifacts disappear as in the right column of Figure 12. The results are obviously much better after applying our algorithm.



Raw disparity with holes

Disparity with holes filled

Figure 11. *Left: raw disparity with holes. Right: disparity processed using our method and holes are filled.*

Comparison to Ground Truth

For completeness, we present our algorithm's result on 'baby3' set from the 2006 Middlebury data. The image resolution is 1312×1110 and the data set provides the ground truth (GT) disparity as in Figure 13. The GT map has holes itself. But to evaluate our algorithm, we manually added 3 holes that cover 5.20% of the image. We applied our hole filling algorithm to this modified disparity map. Figure 13 shows the smoothly filled result and Figure 14 shows 3D plots of the disparities for further illustration. In the 3 introduced holes, the GT disparities range from 137 to 152 with average 150.17 and median 150. The results of comparing the GT values in the 3 holes with our algorithm's filling disparities are in the Table below. The quantitative results shown are encouraging, although our main goal is enabling visually appealing computational photography applications rather than obtaining as precise as possible disparity estimates.

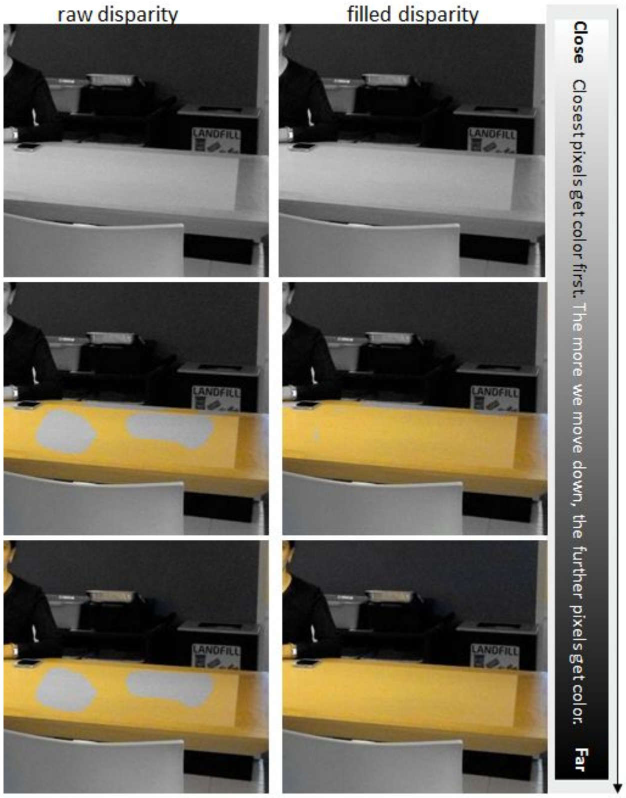

raw disparity
filled disparity
Close
Closest pixels get color first. The more we move down, the further pixels get color.
Far

Figure 12. *As we move a slider from top to bottom, parts of the image should come in color gradually with closer layers before further ones. Left: using raw disparity with holes, artifacts are clearly visible. Right: using disparity with filled holes, with the artifacts heavily reduced.*

Comparison with ground truth

Median of absolute error	1
Mean squared error	2.15
PSNR	40.31 dB
Median of relative absolute error	0.66%
Average of relative absolute error	0.61%

DOI: 10.2352/ISSN.2470-1173.2016.21.3DIPM-401

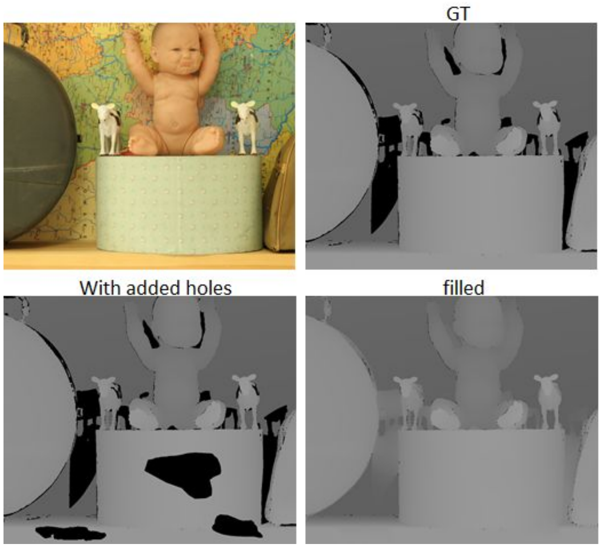


Figure 13. Top left: baby3 Middlebury view1 image. Top right: ground truth disparity (it already has holes). Bottom left: disparity with 3 manually added holes. Bottom right: disparity filled with our algorithm.

Discussion, Conclusion, and Future Work

We have presented an innovative algorithm that reduces the computation cost of high quality hole filling by using a non-linear filtering technique on a low resolution grid to propagate values from a disparity hole boundary into its interior. Instead of filtering the original disparity map, only the areas in the vicinity of holes are filtered. The output map $F(x,y)$ with filling candidate values may not precisely match the original unknown disparities in the holes, yet it creates a smooth continuation of known disparity values into the regions of missing disparity. Therefore, applications using the filtered disparity map will benefit from the improved visual quality. All depth sensing methods based on disparity in images are subject to ambiguities or errors in the disparity estimation process and thus require some method of fixing those errors using a best guess. Given that our method allows improvement of the depth map at a very low computation cost, it could be used as a part of disparity computation stack on mobile platforms, e.g., Intel tablets and phones, which have less computational power than PCs. In the future, we will develop our method further:

1. using RGB boundaries to refine hole detection and avoid having two different holes merged together, e.g., holes on adjacent surfaces of different depth.
2. leveraging confidence estimation for better identification of holes and also better hole filling quality.
3. understanding the scene in the image and identifying key regions. For example, recognizing and localizing the sky helps us immediately fill the disparity values in this region with zero.

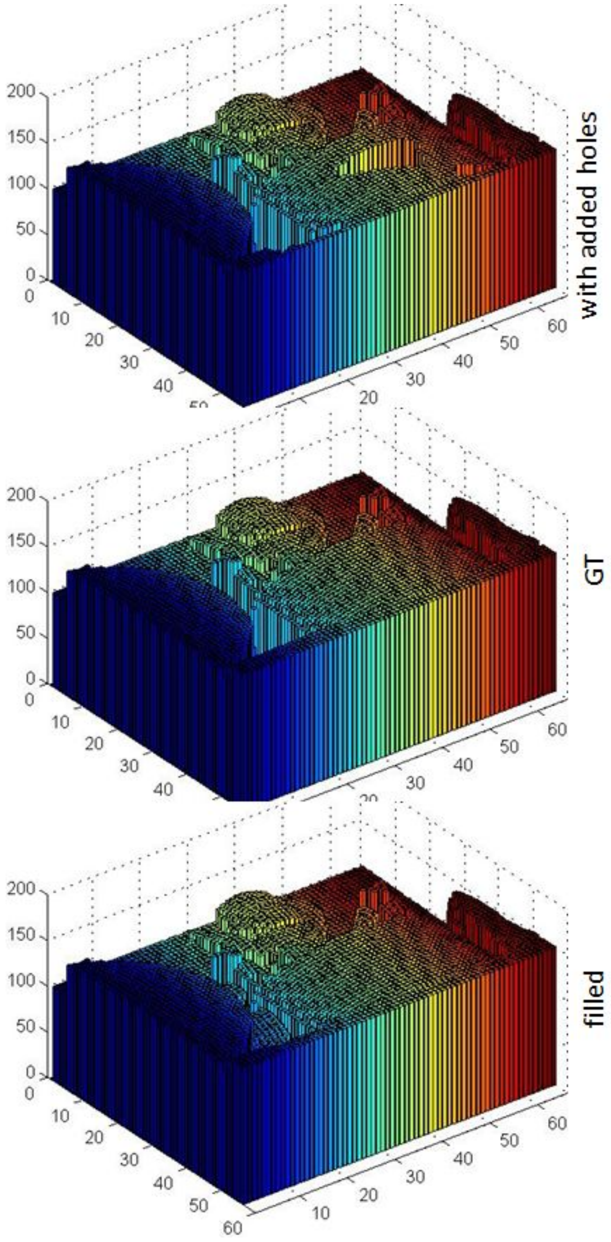


Figure 14. 3D plot of the disparity maps. Top: disparity with 3 manually added holes. Middle: ground truth disparity. Bottom: disparity filled with our algorithm.

References

[1] Richard Szeliski, Computer Vision: Algorithms and Applications, Springer-Verlag, New York, NY, 2010.

[2] Heiko Hirschmuller, Stereo Processing by Semiglobal Matching and Mutual Information, IEEE TPAMI, 30, 328 (2008).

[3] Xing Mei, Xun Sun, Mingcai Zhou, Shaohui Jiao, Haitao Wang, and Xiaopeng Zhang, On Building an Accurate Stereo Matching System on Graphics Hardware, Proc. ICCV Workshops, pg. 179 (2011).

[4] Carlos Vázquez, Wa James Tam, and Filippo Speranza, Stereoscopic Imaging: Filling Disoccluded Areas in Depth Image-Based Rendering, Proc. SPIE 3D TV, Video, Display V, (2006).

[5] Na-Eun Yang, Yong-Gon Kim, and Rae-Hong Park, Depth hole filling using the depth distribution of neighboring regions of depth holes in the kinect sensor, in Proc. ICSPCC, pg. 658 (2012).

[6] Anat Levin, Dani Lischinski, and Yair Weiss, Colorization Using Optimization, in Proc. SIGGRAPH, pg. 689 (2004).

[7] Patrick Pérez, Michel Gangnet, Andrew Blake, Poisson Image Editing, ACM Transactions on Graphics, 22, 313, (2003).

DOI: 10.2352/ISSN.2470-1173.2016.21.3DIPM-401

[8] Scott T. Acton, Multigrid Anisotropic Diffusion, IEEE Trans. on Image Proc., 7, 280, (1998).

[9] Leo Grady, A Lattice-Preserving Multigrid Method for Solving the Inhomogeneous Poisson Equations Used in Image Analysis, Proc. ECCV, pg. 252, (2008).

[10] Si Lu, Xiaofeng Ren, Feng Liu, Depth Enhancement via Low-rank Matrix Completion, Proc. CVPR, pg. 3390, (2014).

[11] Nikolce Stefanoski, Oliver Wang, Manuel Lang, Pierre Greisen, Simon Heinzle, and Aljosa Smolic, Automatic View Synthesis by Image-Domain-Warping, IEEE Trans. on Image Proc., 22, 3329 (2013).

[12] Feng Liu, Michael Gleicher, Hailin Jin, and Aseem Agarwala, Content-Preserving Warps for 3D Video Stabilization, ACM Transactions on Graphics (SIGGRAPH), 28, 44:1 (2009).

[13] Chenlei Guo, Qi Ma, and Liming Zhang, Spatio-Temporal Saliency Detection Using Phase Spectrum of Quaternion Fourier Transform, Proc. CVPR, pg. 18, (2008).

Author Biography

Alexey Supikov (Soupikov) has received MS (1997) and was PhD student (1998-2000) in Mathematics in Nizhny Novgorod State University (named after N.I. Lobachevsky). In 1997-2000 (NSTL ZAO) he worked on computer graphics, computational geometry and CSG contract projects for a number of companies (including Intel Corp., Toshiba, Bricks). He joined Intel Labs in 2000 working on computational displays, ray tracing/global illumination, physics simulation, advanced architecture modeling and workload analysis.

Maha El Choubassi received her BE in Computer and Communications Engineering from the American University of Beirut (2003) and her Master's and PhD in electrical engineering from University of Illinois at Urbana-Champaign (2005 and 2008). She is currently a research scientist at Intel Corporation in Santa Clara, CA. Her work focuses on image/video processing and computer vision.

Oscar Nestares received his M.S. (1994) and Ph.D. (1997) in Electrical Engineering from Universidad Politecnica de Madrid. After that he was a Fulbright Visiting Scholar at Stanford University and consultant at Xerox PARC (1998-2000) and a tenured Research Scientist at the Institute of Optics, Spanish National Research Council (2000-2003). In 2003 he joined Intel Labs focusing on statistical approaches to video enhancement (de-noising, super-resolution, stabilization), its applications to consumer electronics, workload analysis, and computational photography.