DOI: 10.2352/ISSN.2470-1173.2016.21.3DIPM-406

High-Fidelity Time-of-Flight Edge Sampling Using Superpixels

Thomas Hach, Johannes Steurer; Arnold & Richter Cine Technik (ARRI); Munich, Germany
Sascha Knob; Hochschule Rhein-Main; Wiesbaden, Germany

Abstract

This paper focuses on depth map boundary reconstruction by providing a novel goal-oriented Time-of-Flight depth map superresolution approach. State-of-the-art RGB-guided depth map upscaling uses accompanied high-resolution RGB information to upscale depth maps while leaving a minor but non-negligible amount of blurred flying pixels which are physically incorrect in depth maps. These flying pixels highly deteriorate consecutive applications which need a high-definition depth key. Our approach evaluates RGB superpixel segmentation principles to precisely indicate RGB boundaries and to finally transfer this boundary information to the depth map by assigning the maximum-likelihood estimate for each region. Thereby, we achieve discontinued and physically correct depth edges with noticeably reduced flying pixel artifacts. The proposed method overcomes previous algorithms by more than 50% in our dedicated evaluation on real RGBD camera data. This highly accurate depth edge information can be used in future applications relying on depth-based keying.

Introduction

Capturing depth together with two-dimensional RGB color information has accessed numerous applications. Plenty of them like depth-image-based rendering [42], object tracking [2], image segmentation [39] or deep compositing [22] are improved, facilitated or emerge when a depth channel becomes available.

A dense depth map for real image content can be captured using stereovision techniques or Time-of-Flight (TOF) sensors. While stereovision systems deliver high-resolution depth maps with only minor noise, they lack of mechanical and computational simplicity. Contrary, TOF sensors are cheap and easy to use in practical environments but deliver only low-resolution depths maps in the domain of a few hundred pixels per dimension, containing a noticeable amount of noise. Due to the operational benefits, captured depth maps by TOF systems are, however, very interesting for conventional industrial operation as well as novel fields like movie productions, where hand-made or computer-generated depth layers are well-known. Therefore, we consider so-called RGBD cameras, which capture RGB images with matching, probably upscaled, depth maps.

In order to match the higher RGB sensor resolution, information of the color image is used to refine the depth map. However, problems in typical RGB-guided depth map upscaling arouse when color and depth edges are misaligned, blurred or color differences at depth discontinuities are varying a lot. Then, depth edges either bleed over the aimed high-resolution edge position or an effect called texture copying transfers texture to depth structure. Considering RGB-guided upscaling filters, there is an important aspect to captured depth maps that has not yet been handled appropriately. This aspect is a known phenomenon called flying pixels.

Flying pixels are sampling artifacts due to the low TOF sensor resolution. That is, due to the low spatial sampling rate, depth discontinuities generate mixed depth measurements of the object and foreground or background at its boundaries, yielding a displaced distance measurement somewhere in between. This issue is visualized in Figures 1 and 2. In dependence of the usage within a filter, either the native resolution depth map or a naively upscaled depth map is used as input. Figure 1 shows the native resolution depth map, where a noticeable amount of pixels is displaced as flying pixels. Unfortunately state-of-the-art algorithms, like the often used joint bilateral filter [19], don't treat flying pixels correctly with respect to the obtained application-based consequences. Typically flying pixels are used as valid inputs. Hence, there are two cases. First, if the flying pixel represents the center pixel of the kernel, it is included with a high weight leading to a biased result. Second, if the flying pixel is within the considered filtering neighborhood, it can't be fully suppressed as a computed weight of zero is very unlikely. This case also leads to a biased results in the sense of depth.

Our contribution in this paper is twofold. First, we introduce an upscaling algorithm which incorporates color-guided superpixel clustering to indicate depth discontinuities as we assume that object boundaries induce superpixel boundaries. Thereby, we obtain high-resolution depth maps with lower boundary classification errors compared to state-of-the-art upscaling algorithms while softening the demands for smoothness. Our observation, for example in our recent lens effect simulation using RGBD data [13], clearly states that precise object boundaries are more important compared to strict smoothness of surfaces. This argument is underlined by the fact that important visual information is located at object boundaries, which the human visual system is highly sensitive to. Hence, we propose our filter in particular for depth-image-based rendering applications, whose success is dependent on depth edge reconstruction.

Our second contribution is a detailed comparison of different highly potential superpixel algorithms, which build the basis of our approach and, for example, other general superpixel-guided approach like the proposed depth upscaling algorithms by Matsuo et al. [24] or Soh et al. [35].

This paper concludes with a comparison of our approach and significant state-of-the-art filters, focusing on correct edge classification. Therefor, we provide a new metric, which is suitable for noise-aware depth edge classification.

Prior Art

In general, there are specialized methods for *identification* or *combined identification and correction* of flying pixels.

Pure *identification* methods are proposed by May et al. [26], Swadzba et al. [36] and Huhle et al. [17]. These solely remove identified flying pixels from the depth map.

DOI: 10.2352/ISSN.2470-1173.2016.21.3DIPM-406

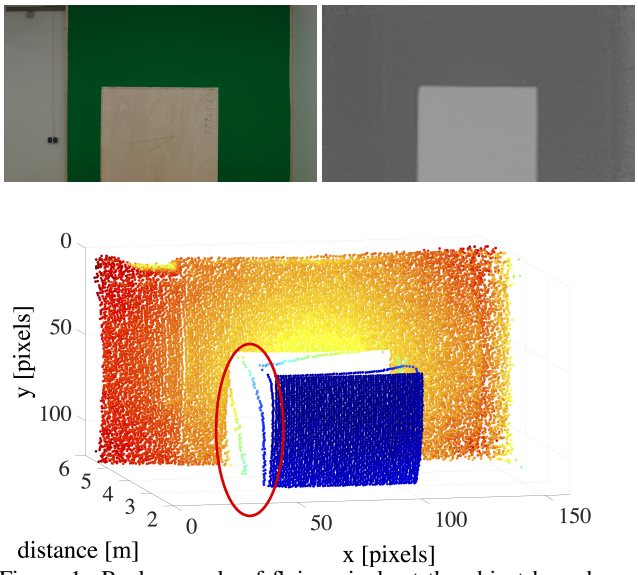

Figure 1: Real example of flying pixels at the object boundary of the box in the foreground and the wall in the background. (top) RGB and depth map; (bottom) false color point cloud of the depth map with red circle indicating flying pixels

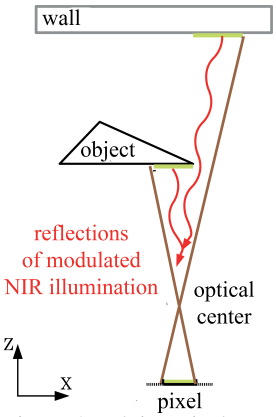

Figure 2: Flying pixels are generated because of the low sampling rate resulting in pixels covering portions of background and foreground at object boundaries.

May et al. proposes to span triangles from the physical position of the sensor pixel to voxels of a neighborhood surrounding the candidate voxel. The voxels are based on the TOF measurement. If an angle in this triangle exceeds a user-defined threshold, the voxel is marked as flying pixel.

Similarly, Swadzba et al. propose a 3x3 neighborhood filter that marks the central pixel to be a non-flying pixel, if the distance to more than 2 pixels of the neighborhood is smaller than a threshold.

While the last two methods solely work on the depth map, Huhle et al. employs a system based on an RGB image and its corresponding TOF depth map. Therein, the RGB image is used to differentiate homogeneous and inhomogeneous regions in terms of color. Then, using a color-weighted Gaussian distance, pixel distances within a neighborhood are evaluated towards a threshold to determine flying pixels.

Identification and correction methods are proposed by Sabov et al. [32] and Richardt et al. [31].

Sabov et al. introduce a so-called score value that measures the distance of the central pixel to its neighbors based on the depth map, similar to the approach of Huhle et al.. If the score value is larger than a threshold, the pixel is referred to as flying pixel. In order to correct the flying pixel, the same neighborhood is considered. The flying pixel is assigned to the pixel value of the neighboring pixel with the smallest score value distance while being a valid pixel. The authors also provide a second approach, which fits line, free or jump segments in the depth map, which is divided into vertical and horizontal scanlines. A flying pixel is defined as a pixel which lies within a jump segment. The correction step verifies whether the flying pixel can be projected onto a neighboring line segment. Compared to our approach, they try to correct flying pixels in the low-resolution domain, which is not appropriate due to the sampling theorem, when subsequent upscaling is applied.

Richardt et al. propose an RGB plus depth camera workflow based on an RGB camera and a depth camera mounted as side-by-side system. A processing step considers edge pixels in the depth map, which are either determined using a thresholded gradient image or surface normals, depending on the given geometry. The marked flying pixels and the occluded pixels, because of the required viewpoint warp, are then filled-in by a multi-scale approach using the joint-bilateral filter [19]. Subsequently a cross-bilateral filter is used to refine the edges with respect to the higher resolution RGB image. Compared to our approach, the subsequent application of the cross-bilateral filter re-introduces flying pixels based on the Gaussian filter kernels, which do not force outliers to be zero-weighted. Thus this blur generates new flying pixels.

Next to these specialized approaches, we also consider typical upscaling as a way of flying pixel correction because upscaling inherently tries to refine edges. As there are numerous color-image-guided depth map upscaling approaches [4, 6, 40, 14, 29, 8, 19, 28, 41, 28, 30, 20, 27, 23, 3], we especially stress the approach by Soh et al. [35]. Therein, they propose to use a superpixel segmentation, generated on the color image and transferred to the depth map, which is similar to our approach. However, reintroducing flying pixels as the approach is followed by a Markov-Random-Field framework to overcome plane-fitting artifacts.

The approaches of Matsuo et al. also consider RGBD data by employing a superpixel segmentation on the color image, transferring the segmentation to the depth map and fitting planes similar to Soh et al. [24, 25]. However, Matsuo et al. extend plane fitting to connecting planes, which suffice a given smoothness constraint.

Van den Bergh et al. approach depth map upscaling and refinement by an extended SEEDS segmentation algorithm [38]. Their main application is to provide closed and dense object silhouettes for robotic vision and recognition tasks.

Method

This section starts with an overview of the employed superpixel clustering algorithms, followed by the description of our approach.

Superpixel Segmentation Candidates

As we provide a recommendation of suitable superpixel algorithms, we present a subset of algorithm candidates first. The chosen superpixel algorithms are selected due to they reported

DOI: 10.2352/ISSN.2470-1173.2016.21.3DIPM-406

performance and scientific popularity.

Contour-Relaxed Superpixels

Contour-Relaxed Superpixels (CRS) by Conrad et al. is a statistical Maximum-A-Posteriori (MAP) approach for superpixel segmentation [5]. Therein, they define the image segmentation $\mathbf{Q}$, the parameter ensemble θ and the random variable z, which denotes the image. The aim is to find Q and θ that optimize the joint probability density $p(z,Q,\theta)$. Regarding the problem using MAP principles leads to

$$J = p(\theta,Q|z) = p(z|Q,\theta)\cdot p(Q,\theta). \tag{1}$$

The maximum is found assuming a Gibbs random field with discrete two-element cliques employing potentials V from the Potts model and isolating the problem to a region-specific maximum likelihood estimation as $Q = \{R_1,...,R_n\}$ is fixed during each iteration. The solver, which is also proposed by Conrad et al., efficiently maximizes Eq. 1 by variation of pixel labels. Therefor, only grid points that are located on region contours and their neighbors are adduced. Changing the region label directly alters V and thus Eq. 1. This process is done for each segment R_i individually, because it is assumed that knowledge of the entire image texture does not change the knowledge of the texture within a segment R_i. Finally, a compactness term that typically controls the shape of the superpixels is added yielding the energy

$$L = J + \kappa \sum_{\mathbf{x}_i \in R_j} (\mathbf{x}_i - \mathbf{m}(R_j))^T (\mathbf{x}_i - \mathbf{m}(R_j)), \tag{2}$$

where the second term expresses the squared distance of each pixel coordinate vector $\mathbf{x} = (x,y)$ within region R_j to its center $\mathbf{m}(R_j)$.

Entropy Rate Superpixels

Entropy Rate Superpixels (ERS) porposed by Liu et al. considers superpixel segmentation as a graph partioning problem in the real of graph theory [21]. Assuming a graph $G = (V,E)$, this approach seeks for a subset of edges $A \subseteq E$, such that the resulting graph $\hat{G}_k = (V,A)$ consists of k connected sub-graphs. The edge potential E is defined as the weighted sum of the entropy rate $H(G)$ of a random walk within G and a regularization term $B(G)$, denoted as

$$\begin{aligned} E(G) =& H(G) + \lambda B(G),\ with \\ H(G) =& -\sum_{n\in V} \frac{w_{m,n}}{\sum_{m\in V} w_{m,n}} \sum_{m\in V} p_{m,n} log(p_{m,n}), \\ B(G) =& -\sum_{i=1}^{K} \frac{|S_i|}{N} log(\frac{|S_i|}{N}). \end{aligned} \tag{3}$$

Therein, $p_{m,n}$ denotes the step probability of the random walk, which is obtained by the edge weights $w_{m,n}$ of G. The weights themselves are computed employing the L1 color distance of the respective pixels m and n. The regularization term allows for the control of the superpixel size by favoring superpixels with small variance in size. The problem is solved using a Greedy algorithm.

Superpixels Extracted via Energy-Driven Sampling

Superpixels Extracted via Energy-Driven Sampling (SEEDS) proposed by van den Bergh et al. optimizes an initial segmentation iteratively [37]. The authors propose to use an hill-climbing algorithm to minimize the energy

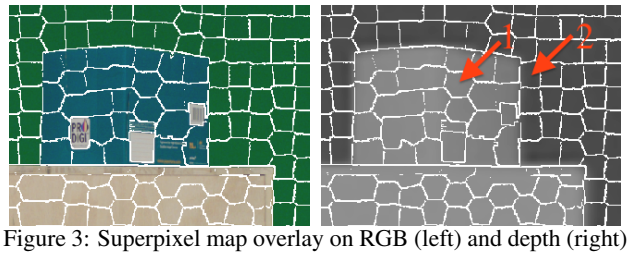

Figure 3: Superpixel map overlay on RGB (left) and depth (right) images; (1) and (2) mark a superpixel on a smooth surface and a superpixel at an object edge, respectively.

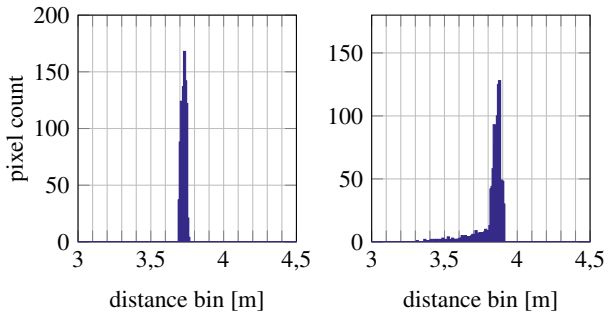

Figure 4: Depth histogram of superpixel 1 in Figure 3 containing only valid pixels (left) and superpixel 2 in Figure 3 containing numerous flying pixels (right)

$$\begin{aligned} E(s) =& H(s) + \gamma G(s),\ where \\ H(s) =& \sum_k \sum_{H_j} (c_{A_k}(j))^2, \\ G(s) =& \sum_i \sum_k (b_{N_i}(k))^2. \end{aligned} \tag{4}$$

Therein, the $H(s)$ term optimizes for color similarity within each superpixel using the color histogram c_{A_k}. $G(s)$ controls the shape of the superpixel using the histogram of superpixel labels b_{N_i} in a neighborhood N_i.

To improve the results, SEEDS is applied hierarchically by forming n levels, where level 0 consists of single pixels and each level $n+1$ builds a new block of a 2x2 block neighborhood in level n. The algorithm starts at the coarse level n. The similarity to superpixels is measured by the intersection of the respective superpixel histograms. Hence, in this block update process, blocks are assigned to superpixels with higher similarity. In level 0, this process is referred to as pixel update, since the amount of identical color values is evaluated in neighboring superpixels. Finally, a contour smoothing process is conducted to obtain more compact superpixels.

Simple Linear Iterative Clustering

Simple Linear Iterative Clustering (SLIC) proposed by Achanta et al. is based on typical k-means clustering. Hence, a 5-dimensional feature space that comprises pixel color distances in the CIE La*b* color space and the pixel coordinates for positional information. Thereby, the distance $D_{i,k}$ of a pixel i to

DOI: 10.2352/ISSN.2470-1173.2016.21.3DIPM-406

a superpixel center k is formulated by

$$\begin{aligned} D_{i,k} &= D_{lab} + \frac{m}{S} D_{xy}, \text{ with} \\ D_{lab} &= \sqrt{(l_i - l_k)^2 + (a_i - a_k)^2 + (b_i - b_k)^2}, \\ D_{xy} &= \sqrt{(x_i - x_k)^2 + (yi - y_k)^2}, \end{aligned} \tag{5}$$

where m is a weighting parameter between the color and the normalization distance $S = \sqrt{\frac{N}{K}}$, with the number of pixels N and number of superpixels K, denotes the superpixel grid interval. The SLIC algorithm alters between the assignment of pixels in accordance to minimal $D_{i,k}$ and subsequent updating of the superpixel centers. SLIC ends with a contour evolution step that assigns incoherent pixel clusters, which are unconnected but assigned to one superpixel, to neighboring superpixel.

Approach

Our approach accepts high-resolution color images and dense occlusion-free low-resolution depth maps. Thus, we use data generated from a monocular RGBD camera system, as described by our previous papers [9, 12]. However, we want to point out that our approach is not limited to this kind of camera systems. There are multiple other systems, which generate similar data like the system described by Bamji et al. [1].

We start at a common base for RGBD data, which comprises a matching naively upscaled depth map using bi-cubic interpolation. This upscaling process includes the registration of the depth map from the raw depth map resolution of 160x90 in our case to the color image geometry and resolution, which is 1920x1080. In particular, the registration routine is a typical step in the realm of sensor fusion and is commonly done separately before higher level filtering, as outlined in our paper considering depth denoising [10]. Considering a monocular system, registration is applied using a full perspective transformation and a appropriate lens model. In contrast, systems using mirrors in front of the lens and binocular RGBD systems require 3D registration techniques and the resolution of occlusions. We next apply one of the specified superpixel algorithms depth-independently on the RGB image. Figure 3 shows an RGB image with corresponding depth map overlaid by the according superpixel segmentation.

As the superpixel segmentation is available, we swap the segmentation information from the RGB image to the depth map. Using these superpixel clusters, it becomes highly beneficial to statistically consider the depth value ensemble instead of single pixel values. Therefore, we directly deploy properties of the depth ensemble histograms, depicted in Figure 4. Therein, histograms of a first superpixel region (1) containing, by definition, only valid depth values and a second superpixel region (2) contaminated by numerous native and interpolated flying pixels, are shown. Obviously, the contaminated region shows a very flat but long tail on the left, compared to the spiky uncontaminated histogram. Deploying this observation, we estimate the maximum likelihood (ML) depth values of each superpixel ensemble individually. This process provides a robust estimate of the most likely depth value that represents a superpixel region.

Regarding the scope of applications outlined in the introductory section, the algorithm is intended to deliver primarily high-fidelity depth edges rather than perfectly smooth surfaces within

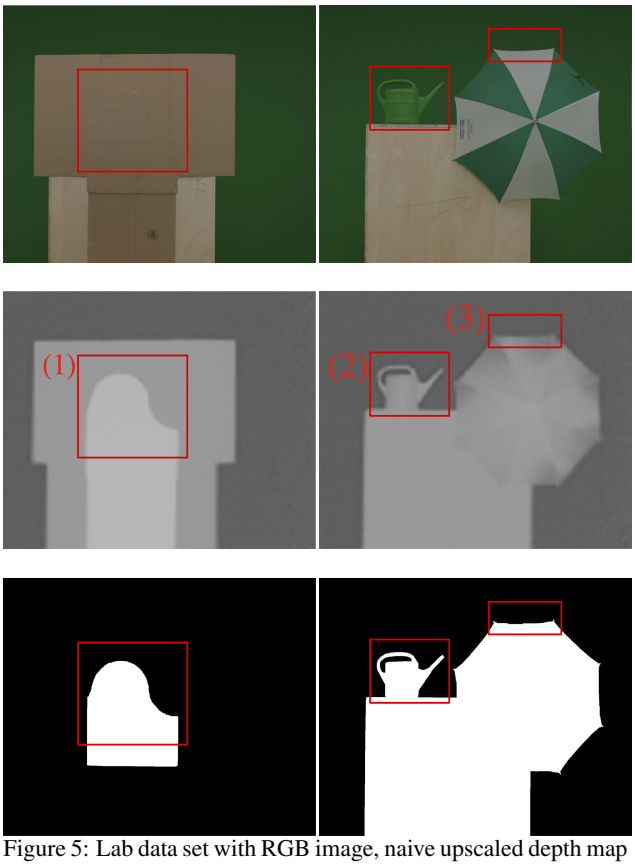

Figure 5: Lab data set with RGB image, naive upscaled depth map and segmentation mask. Test patches for evaluation are indicated by the red box: (1) *cardboard*, (2) *watering can*, (3) *umbrella*

objects. Thus, we assign the ML estimate, which is most likely not a native or a interpolated flying pixel, to the entire superpixel region. Thereby, all object boundaries, which are indicated by superpixel boundaries, are discontinued in the depth. Although being the desired effect, minor depth jumps are also introduced into smooth surfaces of the depth map. Soh et al. discovered a similar problem while fitting planes into the depth ensemble. Therefore, they encountered by using an edge-sensitive maximum a posteriori Markov-Random-Field approach for subsequent smoothing. Unfortunately, this procedure again introduces flying pixels in accordance to our own experiments. Hence, we decided to provide the depth map without any subsequent continuity restriction, yielding a maximum of depth segmentation capabilities. An example of the resulting depth map is shown in Figure 6.

Experimental Evaluation

In this work, we provide an evaluation on real RGBD data. In real TOF depth data, we observe principle-based differences in depth data characteristics compared to synthetic data sets like Middlebury [34, 33]. In detail, we encounter noise types like range ambiguity or Poisson noise as well as measurement errors [7, 11, 16]. Focusing on flying pixels, we observe a lot of noise influence, which leads to temporally highly fluctuating flying pixels, which is not known to be available in TOF simulations yet.

For the evaluation of the depth upscaling performance on real RGBD data, a novel method for ground truth creation was

DOI: 10.2352/ISSN.2470-1173.2016.21.3DIPM-406

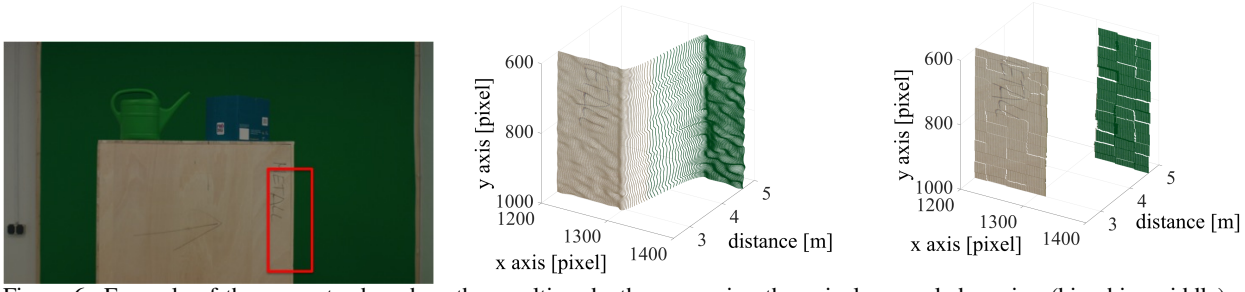

Figure 6: Example of the geometry based on the resulting depth maps using the naively upscaled version (bi-cubic, middle) and our method (right). Please notice the high selectivity of our method yielding to a minimum amount of flying pixels between both surfaces. To avoid confusion: the green screen was not used for segmentation purposes in any of our examples.

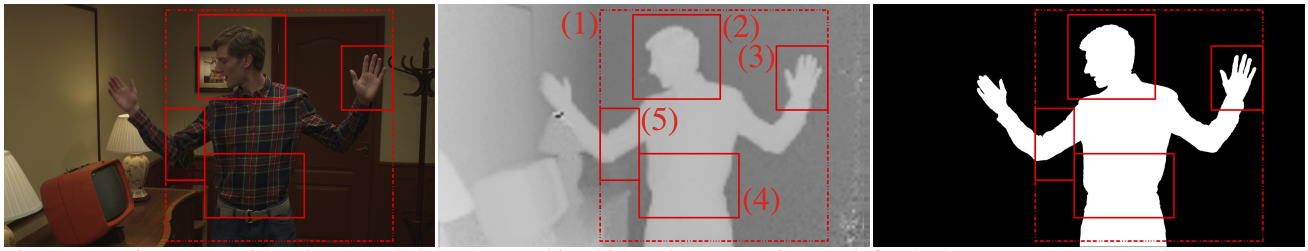

Figure 7: Acting scene with naive upscaled depth map (bi-cubic) and segmentation mask for the human actor. Test patches: (1, dashed line) *body*, (2) *head*, (3) *hand*, (4) *waist*, (5) elbow

Figure 8: Results using our method based on different superpixel algorithms. Rows from top to bottom: bi-cubic upscaling, ERS, SEEDS, SLIC, CRS

developed for phase or round-trip time measuring TOF cameras since there is no ground-truth-free method available yet. In particular, typical depth map evaluation when using the Middlebury data set, deploys metrics like the root-mean-square error or mean absolute deviation. These, give an idea of the similarity between ground truth and processed depth map. However, both metrics do not provide a specific statement on the reproduction of depth discontinuities. This fact becomes obvious when considering typical depth maps, which consist of smooth surfaces and jumps. Jump areas are extremely narrow and seldom as objects consist of much more surface area than depth discontinuities that only describe the boundaries of the objects. Thus, using aforementioned metrics mainly gives an average statement of the distance between ground truth and processed depth surfaces. Contrary, we aim for a metric that specifically provides quantitative information about the reconstruction of depth discontinuities.

The following section provides details on the developed metric as well as the generated data set.

Ground Truth and Metrics for Real Data

A single RGB frame, like it is available in the discussed systems, can be used as input for quick manual rotoscoping. Optionally partitioning of the rotoscoped objects provides a series of ground truth patches, which can be obtained from one single image. Furthermore, one opaque pixel must correspond to one depth value. Semi-transparent RGB pixels can't match sensible depth values. Hence, for ground truth selection, all image parts with in-focus depth jumps are suitable. Figures 5 and 7 depict such rotoscoped binary masks. Therein, object boundaries are correct up to minimal blur which arouses from remaining low pass properties of physical lenses, even when they are fully focused, and low pass properties of filter glasses in front of the image sensor. Additionally, also high-resolution color sensors are limited to sampling.

DOI: 10.2352/ISSN.2470-1173.2016.21.3DIPM-406

Hence, semi-transparency is also induced by mixture of foreground and background colors, similar to the occurrence of native flying pixels. We account for this minimal blur at boundaries, which does not correspond to semi-transparencies, by drawing the mask boundary in the visual center of the blurred RGB region.

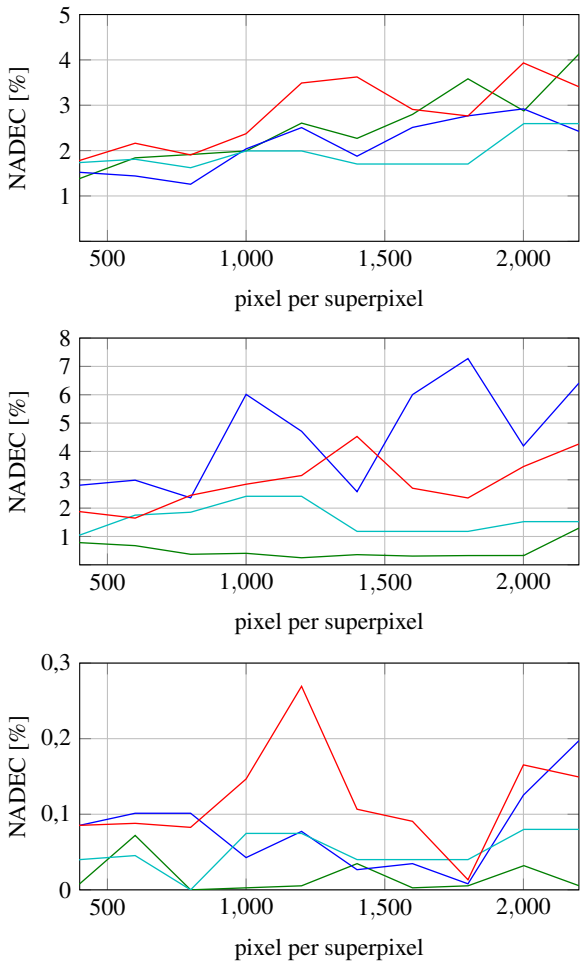

Figure 9: NADEC value when sweeping the amount of superpixels per image. CRS, ERS, SEEDS and SLIC for the *cardboard* patch (top, *watering can* patch (middle), *umbrella* patch (bottom)
— CRS: $\beta = 0.001$; 256 bins — ERS: $\lambda = 1.1$; 64 bins
— SEEDS: $\gamma = 0.25$; 1024 bins — SLIC: $m = 6$; 64 bins

Next, we define a criterion that reveals the depth segmentation performance of depth boundaries. Therefore, we use the bi-cubicly upscaled raw depth map and the mask from rotoscoping the depth patch to segment into the object and remainder. The mean value and the standard deviation are calculated for each of the two regions separately. As a numerical measure, our metric utilizes typical classification metrics. In this case, it's the number of false positives f_p and the number of false negatives f_n.

Considering depth maps, false negative pixels are depth values of the background located in areas of the foreground whereas false positive pixels consist of depth values of the foreground object bleeding into the background area. Subsequent classification is done by masking and cutting the processed depth map. Depth values outside a range of $\mu \pm 3\sigma$ of the respective area are declared as false. Thereby, we are able to correctly classify more than 99% of the inliers while respecting signal noise.

Our metric called noise-aware depth edge classification (NADEC) is mathematically built by

$$NADEC = \frac{f_n + f_p}{N}. \tag{6}$$

where N normalizes to the total number of pixels in the binary mask. Hence, we obtain 0, as best possible value, if and only if all depth pixels share perfect segmentation and thus best upscaling with respect to the ground truth mask.

For a more detailed evaluation, we subdivided the *Body* segment, seen in Figure 7, in 4 patches, *head*, *hand*, *waist*, *elbow*, which emphasize different challenging situations for RGB-guided upscaling filters.

Head is difficult for three reasons. Around the nose part, the background provides a drawing which is very arbitrary in structure but similar in color compared to the skin tone. A proper segmentation is hard to achieve. Hair typically provides only poor depth signal and the color is similar to the background which gives an uncertain key, too. The opened mouth is difficult to distinguish from the background in color but provides a detailed and distinct depth structure.

Hand only provides a glove-like structure in the raw depth map because of the low native resolution which can't give a proper depth map of the actor's fingers. Furthermore, due to the thin structure of the fingers, the native depth map is likely to consist of flying pixels instead of valid depth values since the projection of the finger onto the sensor is thinner than the pixel's diameter.

Waist is prone to texture copying due to the significant structure of the actors woven shirt. The red patterns of the shirt are quite similar to the background color and hence color-guided upscaling filters require strong color sensitivity settings to distinguish between foreground and background. However, this is said to lead to texture copying within smooth surfaces.

Elbow is challenging due to the green leaves in the background and the green patterns on the shirt. Thus, edge bleeding is likely to occur.

Numerical Results

Our numerical evaluation consists of two parts. First, we provide a detailed comparison of the upscaling performance while altering the employed superpixel algorithm among the 4 selected algorithms CRS, ERS, SEEDS and SLIC. Figure 9 shows the NADEC value when sweeping the amount of superpixels per image. The amount of superpixels per image corresponds to the cluster size and thus, to the granularity of object patches that can be described using superpixels. Figure 8 shows the depth maps using the minimum NADEC value from Figure 9.

Represented by the NADEC value, Table 1 provides a numerical evaluation of the upscaling performance of our approach compared to a selection of state-of-the-art algorithms in color-guided depth map upscaling. Therein, we tested on 5 different image patches to obtain a small statistical set. On average, our method is superior to the selected state-of-the art methods with respect to our experiment. The typical improvement is more than 50% of the NADEC value. Solely our implementation of WMF by Min et al. is close to our approach. However, due to the enormous execution time of WMF while computing the parameter sweep forced us to reduce the problem to an 8-bit

DOI: 10.2352/ISSN.2470-1173.2016.21.3DIPM-406

(a) nearest (b) bi-linear (c) bi-cubic

(d) JBU [19] (e) GF 1 stage [15] (f) GF 2 stage [18]

(g) NAFDU [4] (h) SDIS [35] (i) TGV [8]

(j) WMF [[27] (k) Our method (ERS) (l) Our method (SEEDS)

(m) Our method (SLIC) (n) Our method (CRS)

Figure 10: Resulting depth map of sate-of-the-art filters and our approach with each of the 4 possible superpixel algorithms

DOI: 10.2352/ISSN.2470-1173.2016.21.3DIPM-406

depth map in this case. The quantization of the depth leads to less distributed flying pixels, meaning that an algorithm yields improved results when using the NADEC value.

Figure 10 visually compares the results from Table 1 and the 4 possible results using our framework. Considering all numerical and visual results when employing different superpixel algorithms for our approach, we recommend using SLIC or CRS as ERS and SEEDS show obvious drawbacks when segmenting the color image. This is not necessarily bad segmentation. To obtain a good depth assignment, it is also of the same importance to yield very compact superpixel. This avoids tubular superpixels along object boundaries, which only contain flying pixels for subsequent ML estimation.

Table 1: State-of-the-art algorithms in comparison to our approach using the acting scene. Benchmark was generated employing the proposed metric NADEC [%]. Best values of each column are marked in bold face; second best value in italic face

Method	Body	Head	Hand	Elbow	Waist	Avg.
Nearest	1.57	0.30	0.17	0.11	0.17	0.46
Bi-linear	1.73	0.32	0.14	0.12	0.18	0.50
Bi-cubic	1.59	0.30	0.13	0.10	0.16	0.47
JBU [19]	1.31	0.29	0.12	0.064	0.17	0.39
NAFDU [4]	1.34	0.28	0.12	0.067	0.16	0.39
GF_1s [15]	1.37	0.29	0.10	0.082	0.16	0.40
GF_2s [18]	1.38	0.28	0.11	0.075	0.13	0.40
SDIS [35]	1.13	0.24	0.17	0.085	0.26	0.38
TGV [8]	1.36	0.26	0.13	0.11	0.26	0.42
$WMF_{(8\text{-bit})}$ [27]	*0.60*	0.26	0.13	**0.028**	**0.061**	0.23
Our method (CRS)	**0.59**	*0.19*	*0.10*	0.044	*0.084*	**0.20**
Our method (ERS)	0.71	*0.19*	**0.087**	0.042	0.13	0.23
Our method (SEEDS)	0.72	**0.18**	0.13	0.038	0.077	0.23
Our method (SLIC)	0.70	0.21	0.12	*0.032*	0.095	0.23

Conclusion

In this paper, we introduced a novel application-demanded depth map upscaling algorithm. Our approach strongly focuses on depth jump discontinuation, which is necessary due to the outlined depth map artifact called flying pixels. Therefore, we employ state-of-the-art superpixel segmentation on an accompanied RGB image to obtain clusters at object boundaries. Those clusters are processed by our algorithm, yielding superior depth edge reconstruction results on a real RGBD data set compared to state-of-the-art approaches. For detailed evaluation purposes, we designed a new metric, which gives a measure for the quality of depth jump reconstruction in upscaled TOF depth maps. Furthermore, as our approach relies on superpixel algorithms, we provide a numerical comparison and criterions to select suited ones.

Our improved depth edge representation is especially important for applications that make use of a clean depth key to separate objects like, for instance, news speakers in green screen TV studios when substituting the green screen by an RGBD camera in the future or depth-image-based rendering, which requires convincing results at object boundaries, where the human observer is particularly attentive.

References

[1] C. Bamji and P. Zhao. Single chip red, green, blue, distance (RGB-Z) sensor. US Patent Office, 8139141, 2012.

[2] A. Bevilacqua, L. Stefano, and P. Azzari. People Tracking Using a Time-of-Flight Depth Sensor. In *IEEE International Conference on Video and Signal Based Surveillance*, page 89. IEEE, 2006.

[3] M. Camplani and L. Salgado. Adaptive spatio-temporal filter for low-cost camera depth maps. *IEEE International Conference on Emerging Signal Processing Applications, 2012*, pages 33–36, 2012.

[4] D. Chan, H. Buisman, C. Theobalt, and S. Thrun. A noise-aware filter for real-time depth upsampling. In *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications-M2SFA2*, 2008.

[5] C. Conrad, M. Mertz, and R. Mester. Contour-Relaxed Superpixels. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 280–293. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[6] J. Diebel and S. Thrun. An application of markov random fields to range sensing. In *IEEE International Conference on Computer Vision*, pages 291–298, 2005.

[7] T. Edeler. *Bildverbesserung von Time-Of-Flight Tiefenkarten*, pages 12–15. Shaker, Aachen, 2012.

[8] D. Ferstl, C. Reinbacher, and R. Ranftl. Image guided depth upsampling using anisotropic total generalized variation. *IEEE International Conference on Computer Vision*, pages 993–1000, 2013.

[9] T. Hach, C. Bosch, P. Arias, J. Montesa, and P. Gasco. Seamless 3D Interaction of Virtual and Real Objects in Professional Virtual Studios. In *SMPTE Annual Technical Conference Exhibition*, pages 1–20, 2015.

[10] T. Hach and T. Seybold. Joint Video and Sparse 3D Transform-Domain Collaborative Filtering for Time-of-Flight Depth Maps. In *IEEE International Symposium on Multimedia*, pages 1–6, 2015.

[11] T. Hach, T. Seybold, and H. Böttcher. Phase-aware candidate selection for time-of-flight depth map denoising. In *IS&T/SPIE Electronic Imaging*, pages 93930E–93930E–9, 2015.

[12] T. Hach and J. Steurer. A novel RGB-Z camera for high-quality motion picture applications. In *Proceedings of the 10th European Conference on Visual Media Production*, pages 1–10, 2013.

[13] T. Hach, J. Steurer, A. Amruth, and A. Pappenheim. Cinematic Bokeh rendering for real scenes. In *Proceedings of the 12th European Conference on Visual Media Production*, pages 1–10, 2015.

[14] K. He, J. Sun, and X. Tang. Guided Image Filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409, 2015.

[15] K. He, J. Sun, and X. Tang. Guided Image Filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409, 2015.

[16] H. Hirschmuller and D. Scharstein. Evaluation of Cost Functions for Stereo Matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[17] B. Huhle, P. Jenke, and W. Straßer. On-the-fly scene acquisition with a handy multi-sensor system. *International Journal of Intelligent Systems Technologies and Applications*, 5(3/4):255–263, 2008.

[18] T.-W. Hui and K. N. Ngan. Depth enhancement using RGB-D guided filtering. In *IEEE International Conference on Image Processing*, pages 3832–3836, 2014.

[19] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele. Joint bi-

DOI: 10.2352/ISSN.2470-1173.2016.21.3DIPM-406

lateral upsampling. *ACM Transactions on Graphics*, 26(3):96, 2007.

[20] Y. Li, T. Xue, L. Sun, and J. Liu. Joint Example-Based Depth Map Super-Resolution. In *2012 IEEE International Conference on Multimedia and Expo*, pages 152–157, 2012.

[21] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa. Entropy rate superpixel segmentation. In *Conference on Computer Vision and Pattern Recognition*, pages 2097–2104, 2011.

[22] T. Lokovic and E. Veach. Deep shadow maps. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 385–392. ACM Press, 2000.

[23] Z. Ma, K. He, Y. Wei, J. Sun, and E. Wu. Constant Time Weighted Median Filtering for Stereo Matching and Beyond. In *2013 IEEE International Conference on Computer Vision*, pages 49–56, 2015.

[24] K. Matsuo and Y. Aoki. Depth Interpolation via Smooth Surface Segmentation Using Tangent Planes Based on the Superpixels of a Color Image. In *2013 IEEE International Conference on Computer Vision Workshops*, pages 29–36, 2014.

[25] K. Matsuo and Y. Aoki. Depth image enhancement using local tangent plane approximations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3574–3583, 2015.

[26] S. May, D. Droeschel, D. Holz, and C. Wiesen. 3D pose estimation and mapping with time-of-flight cameras. *International Conference on Intelligent Robots and Systems, 3D Mapping workshop*, 2008.

[27] D. Min, J. Lu, and M. N. Do. Depth Video Enhancement Based on Weighted Mode Filtering. *IEEE Transactions on Image Processing*, 21(3):1176–1190, 2015.

[28] M. Mueller, F. Zilly, and P. Kauff. Adaptive cross-trilateral depth map filtering. In *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video, 2010*, pages 1–4, 2010.

[29] J. Park, H. Kim, Y.-W. Tai, M. S. Brown, and I. Kweon. High quality depth map upsampling for 3D-TOF cameras. In *IEEE International Conference on Computer Vision*, pages 1623–1630, 2011.

[30] C. Richardt, D. Orr, I. Davies, and A. Criminisi. Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid. *Computer Vision–ECCV 2010*, 6313(37):510–523, 2010.

[31] C. Richardt, C. Stoll, N. A. Dodgson, H.-P. Seidel, and C. Theobalt. Coherent Spatiotemporal Filtering, Upsampling and Rendering of RGBZ Videos. *Computer Graphics Forum*, 31(2pt1):247–256, 2012.

[32] A. Sabov and J. Krüger. Identification and correction of flying pixels in range camera data. In *Proceedings of the 24th Spring Conference on Computer Graphics*, pages 135–142, 2008.

[33] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nesic, X. Wang, and P. Westling. *High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth*, pages 31–42. 2014.

[34] D. Scharstein and C. Pal. Learning Conditional Random Fields for Stereo. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[35] Y. Soh, J.-Y. Sim, C.-S. Kim, and S.-U. Lee. Superpixel-based depth image super-resolution. In *IS&T/SPIE Electronic Imaging*, pages 82900D–10, 2012.

[36] A. Swadzba, B. Liu, and J. Penne. A comprehensive system for 3D modeling from range images acquired from a 3D ToF sensor. In *Proceedings of the 5th International Conference on Computer Vision Systems*, 2007.

[37] M. Van den Bergh, X. Boix, G. Roig, B. de Capitani, and L. Van Gool. SEEDS: Superpixels Extracted via Energy-Driven Sampling. In *Computer Vision – ECCV 2012*, pages 13–26. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[38] M. Van den Bergh, D. Carton, and L. J. Van Gool. Depth SEEDS: Recovering incomplete depth data using superpixels. *2013 IEEE Workshop on Applications of Computer Vision*, pages 363–368, 2013.

[39] L. Wang, C. Zhang, and R. Yang. Tofcut: Towards robust real-time foreground extraction using a time-of-flight camera. In *International Symposium on 3D Data Processing, Visualization and Transmission*, 2010.

[40] J. Yang, X. Ye, K. Li, C. Hou, and Y. Wang. Color-Guided Depth Recovery From RGB-D Data Using an Adaptive Autoregressive Model. *IEEE Transactions on Image Processing*, 23(8):3443–3458, 2015.

[41] Q. Yang, R. Yang, J. Davis, and D. Nister. Spatial-Depth Super Resolution for Range Images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[42] X. Yang, J. Liu, J. Sun, X. Li, W. Liu, and Y. Gao. DIBR based view synthesis for free-viewpoint television. In *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, 2011*, pages 1–4, 2011.

Author Biography

Thomas Hach received his B.Sc. and his Dipl.-Ing. degree from the Technical University Munich (TUM) in 2011 and 2013, respectively. Since then he is working as part of the R&D department of Arnold & Richter Cine Technik. His current focus lies on depth sensing and sensor fusion signal processing, which is part of his doctorate supervised by the chair for data processing at TUM.

Johannes Steurer received his Dr.-Ing. degree in electrical engineering from Technical University Munich in 1992. Today he is Principal Engineer R&D at ARRI Cine Technik and is responsible for research and technical innovations for motion picture cameras such as depth-sensing, self-localization, lens focusing. Johannes represents ARRI in international collaborative research projects. He is an expert reviewer for the European Commission, a member of SMPTE, FKTG, and VDI and received several awards including an OSCAR® Statuette.

Sascha Knob received his B.E. in media engineering from the University of Applied Science Wiesbaden Rüsselsheim, Germany (2015). He developed his bachelor thesis in the research and development division at ARRI in Munich and worked in the field of 3D imaging. His thesis includes the application of superpixel algorithmes. Presently, he is a master student at the University of Applied Science Wiesbaden Rüsselsheim.