EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

Methods for depth-map filtering in view-plus-depth 3D video representation

Sergey Smirnov, Atanas Gotchev* and Karen Egiazarian

Abstract

View-plus-depth is a scene representation format where each pixel of a color image or video frame is augmented by per-pixel depth represented as gray-scale image (map). In the representation, the quality of the depth map plays a crucial role as it determines the quality of the rendered views. Among the artifacts in the received depth map, the compression artifacts are usually most pronounced and considered most annoying. In this article, we study the problem of post-processing of depth maps degraded by improper estimation or by block-transform-based compression. A number of post-filtering methods are studied, modified and compared for their applicability to the task of depth map restoration and post-filtering. The methods range from simple and trivial Gaussian smoothing, to in-loop deblocking filter standardized in H.264 video coding standard, to more comprehensive methods which utilize structural and color information from the accompanying color image frame. The latter group contains our modification of the powerful local polynomial approximation, the popular bilateral filter, and an extension of it, originally suggested for depth super-resolution. We further modify this latter approach by developing an efficient implementation of it. We present experimental results demonstrating high-quality filtered depth maps and offering practitioners options for highest-quality or better efficiency.

1 Introduction

View-plus-depth is a scene-representation format where each pixel of the video frame is augmented with depth value corresponding to the same viewpoint [1]. The depth is encoded as gray-scale image in a linear or logarithmic scale of eight or more bits of resolution. An example is given in Figure 1a,b. The presence of depth allows generating virtual views through so-called depth image based rendering (DIBR) [2] and thus offers flexibility in the selection of viewpoint as illustrated in Figure 1c. Since the depth is given explicitly, the scene representation can be rescaled and maintained as to address parallax issues of 3D displays of different sizes and pixel densities [3]. The representation also allows generating more than two virtual views which is demanded for auto-stereoscopic displays.

Another advantage of the representation is its backward compatibility with conventional single-view broadcasting formats. In particular, MPEG-2 transport stream standard used in DVB broadcasting allows transmitting auxiliary streams along with main video, which makes possible to enrich a conventional digital video transmission with depth information without hampering the compatibility with single-view receivers.

The major disadvantages of the format are the appearance of dis-occluded areas in rendered views and inability to properly represent most of the semi-transparent objects such as fog, smoke, glass-objects, thin fabrics, etc. The problems with occlusions are caused by the lack of information about what is behind a foreground object, when a new-perspective scene is synthesized. Such problems are tackled by occlusion filling [4] or by extending the format to *multi-view multi-depth*, or to *layered depth* [3].

Quality is an important factor for the successful utilization of depth information. Depth map degraded by strong blocky artifacts usually produces visually unacceptable rendered views. For successive 3D video transmission, efficient depth post-filtering technique should be considered.

Filtering of depth maps has been addressed mainly from the point of view of increasing the resolution [5-7]. In [6], a joint bilateral filtering has been suggested to upsample low-resolution depth maps. The approach has been further refined in [7] by suggesting proper anti-

* Correspondence: Atanas.Gotchev@tut.fi
Tampere University of Technology, Korkeakoulunkatu 10, FI-33720, Tampere, Finland

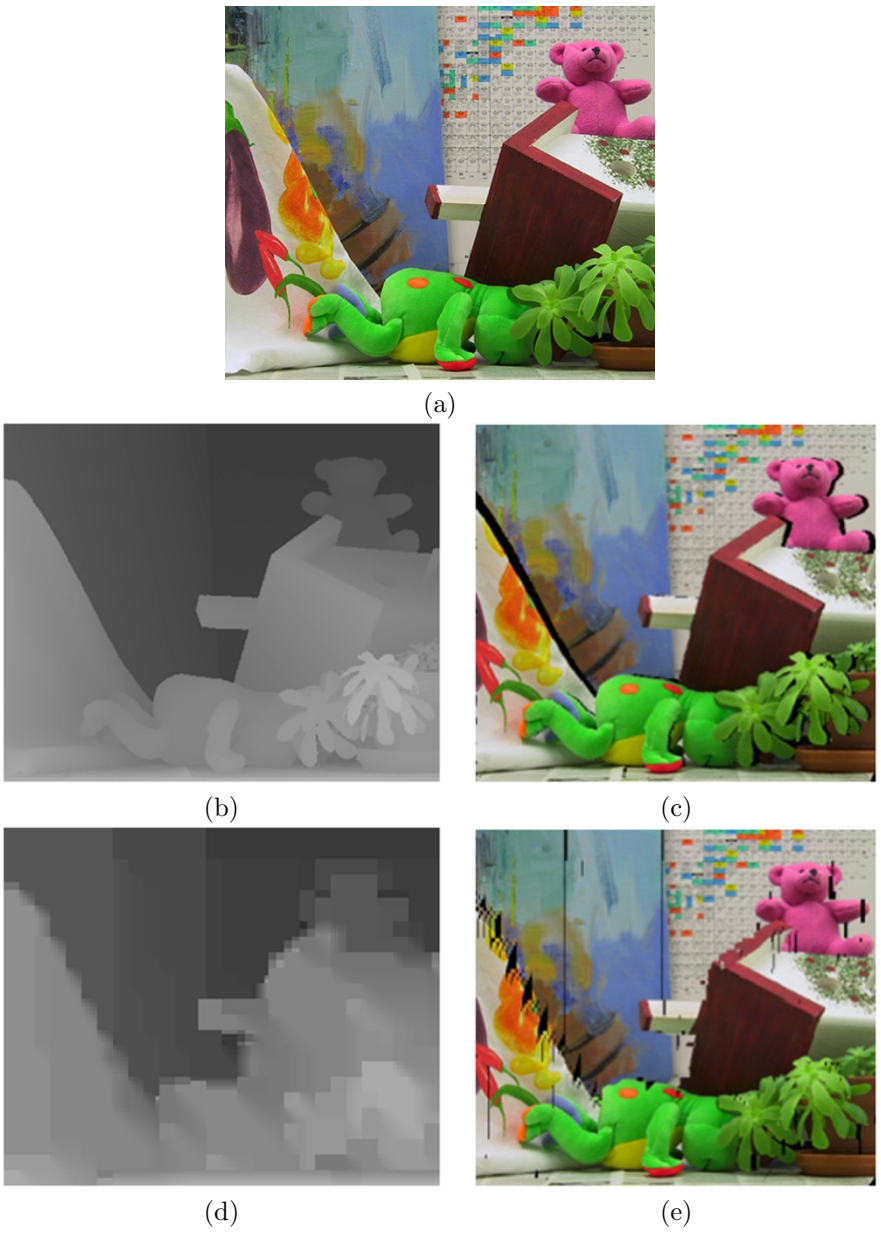


Figure 1 Example of view-plus-depth image format and virtual view rendering (no occlusion filling applied for rendered images). **(a)** True color channel; **(b)** true depth channel; **(c)** synthesized view using true depth; **(d)** highly compressed depth (H.264 I-frame with QP = 51); **(e)** synthesized view using compressed depth from **(d)**.

aliasing and complexity-efficient filters. In [5], a probabilistic framework has been suggested. For each pixel of the targeted high-resolution grid, several depth hypothesizes are built and the hypothesis with lowest cost is selected as a refined depth value. The procedure is run iteratively and bilateral filtering is employed at each iteration to refine the cost function used for comparing the depth hypotheses.

In this article, we study the problem of post-processing of depth maps degraded by improper estimation or by block-transform-based compression. A number of post-filtering methods are studied, modified, and compared for their applicability to the task of depth map restoration and post-filtering. We consider methods ranging from simple and trivial smoothing and deblocking methods to more comprehensive methods which utilize structural and color information from the accompanying color image frame. The present study is an extension of the study reported in [8]. Some of the methods included in the comparative analysis in [8] have been further

modified and for one of them, a more efficient implementation has been proposed. We present extended experimental results which allow evaluating the advantages and limitations of each method and give practitioners options for trading-off between highest quality and better efficiency.

2 Depth map characteristics

2.1 Properties of depth maps

Depth map is gray-scale image which encodes the distance to the given scene pixels for a certain perspective. The depth is usually aligned with and ac-companies the color view of the same scene [9].

Single view plus depth is usually a more efficient representation of a 3D scene than two-channel stereo. It directly encodes geometrical information contained otherwise in the disparity between the two views thus providing scalability and possibility to render multiple views for displays with different sizes [1]. Structure-wise, the depth image is piecewise smooth (as representing gradual change of depth within objects) with delineated, sharp discontinuities at object boundaries. Normally, it contains no textures. This structure should be taken into account when designing compression or filtering algorithms.

Having a depth map given explicitly along with color texture, a virtual view for a desired camera position can be synthesized using DIBR [2]. The given depth map is first inversely-transformed to provide the absolute distance and hence the world 3D coordinates of the scene points. These points are projected then onto a virtual camera plane to obtain a synthesized view. The technique can encounter problems with dis-occluded pixels, non-integer pixel shifts, and partly absent background textures, which problems have to be addressed in order to successfully apply it [1].

The quality of the depth image is a key factor for successful rendering of virtual views. Distortions in the depth channel may generate wrong objects contours or shapes in the rendered images (see, for example, Figure 1d,e) and consequently hamper the visual user experience manifested in headache and eye-strain, caused by wrong contours of familiar objects. At the capture stage, depth maps might be not well aligned with the corresponding objects. Holes and wrongly estimated depth points (outliers) might also exist. At the compression stage, depth maps might suffer from blocky artifacts if compressed by contemporary methods such as H.264 [10]. When accompanying video sequences, the consistency of successive depth maps in the sequence is an issue. Time-inconsistent depth sequences might cause flickering in the synthesized views as well as other 3D-specific artifacts [11].

At the capture stage, depth can be precisely estimated in floating-point hight resolution, however, for compression and transmission it is usually converted to integer values (e.g., in 256 gray-scale gradations). Therefore, the depth range and resolution have to be properly maintained by suitable scaling, shifting, and quantizing, where all these transformations have to be invertible.

Depth quantization is normally done in linear or logarithmic scale. The latter approach allows better preservation of geometry details for closer objects, while higher geometry degradation is tolerated for objects at longer distances. This effect corresponds to the parallax-based human stereo-vision, where the binocular depth cue losses its importance for more distanced objects and is more important and dominant for closer objects. The same property can be achieved if transmitting linearly quantized *inverse* depth maps. This type of depth representation basically corresponds to *binocular disparity* (also known as *horizontal parallax*), including again necessary modifications, such as scaling, shifting, and quantizing.

2.2 Depth map filtering problem formulation

This section formally formulates the problem of filtering of depth maps and specifies the notations used hereafter. Consider an individual color video frame in YUV (YCbCr) or RGB color space $\boldsymbol{y}(\boldsymbol{x}) = [y^Y(\boldsymbol{x}), y^U(\boldsymbol{x}), y^V(\boldsymbol{x})]$ or $\boldsymbol{y}(\boldsymbol{x}) = [y^R(\boldsymbol{x}), y^G(\boldsymbol{x}), y^B(\boldsymbol{x})]$, together with the associated per-pixel depth $z(\boldsymbol{x})$, where $\boldsymbol{x} = [x_1, x_2]$ is a spatial variable, $\boldsymbol{x} \in X$, X being the image domain.

A new, virtual view $\boldsymbol{\eta}(\boldsymbol{x}) = [\eta^Y(\boldsymbol{x}), \eta^U(\boldsymbol{x}), \eta^V(\boldsymbol{x})]$ can be synthesized out of the given (reference) color frame and depth by DIBR, applying projective geometry and knowledge about the reference view camera, as discussed in Section 2.1 [2]. The synthesized view is composed of two parts, $\boldsymbol{\eta} = \boldsymbol{\eta}_v + \boldsymbol{\eta}_o$, where $\boldsymbol{\eta}_v$ denotes the visible pixels from the position of the virtual view camera and $\boldsymbol{\eta}_o$ denotes the pixels of occluded areas. The corresponding domains are denoted by $\boldsymbol{X}_v$ and $\boldsymbol{X}_o$ correspondingly, $\boldsymbol{X}_v \subset \boldsymbol{X}$, $\boldsymbol{X}_o = \boldsymbol{X} \backslash \boldsymbol{X}_v$.

Both $\boldsymbol{y}(\boldsymbol{x})$ and $z(\boldsymbol{x})$ might be degraded. The degradations are modeled as additive noise contaminating the original signal

$$y_q^C = y^C + \varepsilon^C, \tag{1}$$

$$z_q = z + \in, \tag{2}$$

where $C = Y, U, V$ or R, G, B. Both degradations are modeled as independent white Gaussian processes: $\varepsilon^C(\cdot) \sim N(0, \sigma_C^2)$, $\varepsilon(\cdot) \sim N(0, \sigma^2)$. Note that the variance of color signal noise (σ_C^2) differs from the one of the depth signal noise (σ^2).

If degraded depth and reference view are used in DIBR, the result will be a lower-quality synthesized view $\breve{\eta}$. Unnatural discontinuities, e.g., blocking artifacts, in the degraded depth image cause geometrical distortions and distorted object boundaries in the rendered view. The goal of the filtering of degraded depth maps is to mitigate the degradation effects (caused by e.g., quantization or imperfect depth estimation) in the depth image domain, i.e., to obtain a refined depth image estimate $\hat{z}$, which would be closer to the original, error-free depth, and would improve the quality of the rendered view.

2.3 Depth map quality measures

Measuring the quality of depth maps has to take into account that depth maps are type of imagery which are not visualized per-se, but through rendered views.

In our study, we consider two types of measures:

- measures based on comparison between processed and *ground truth* (reference) depth;
- measures based on comparison between virtual views rendered from processed depth and from ground truth one.

Measures for the first group have the advantage of being simple, while measures from the second group are closer to subjective perception of depth. For both of these groups we suggest and test new measures.

PSNR of Restored Depth

Peak signal-to-noise ratio (PSNR) measures the ratio between the maximum possible power of a signal (within its range) and the power of corrupting noise. *PSNR* is commonly used as a measure of fidelity of image reconstruction. *PSNR* is calculated via the mean squared error (*MSE*):

$$MSE = \frac{1}{N}\sum_{x}(z(x) - \hat{z}(x))^2, \tag{3}$$

$$PSNR = 10\log_{10}\left(\frac{MAX_z^2}{MSE}\right) \tag{4}$$

where $z(\boldsymbol{x})$ and $\hat{z}(x)$ are the reference and processed signals; N is number of samples (pixels) and MAX_z is the maximal possible pixel value, assuming the minimal one is zero. In this metric higher value means better quality. Applying *PSNR* to depth images must be done with care and with proper rescaling, as most of depth maps have a sub-range of the usual 8-bit range of 0 to 255 and *PSNR* might turn to be unexpectedly high.

PSNR of rendered view

PSNR is calculated to compare the quality of rendered view using processed depth versus that of using original depth [10]. It essentially measures how close the rendered view is to the 'ideal' one. In our calculations, pixels, dis-occluded during the rendering process, are excluded so to make the comparison independent on the particular hole fitting approach. For color images, we calculate *PSNR* independently for each color channel and then calculate the mean between three channels.

Percentage of bad pixels

Bad pixels percentage metric is defined in [12] to measure directly the performance of stereo-matching algorithms.

$$BAD = \frac{100}{M}\sum_{x}\left(\left|z(x) - \hat{z}(x)\right| > \Delta_d\right),$$

where $\hat{z}$ is the computed depth, z is the true depth and Δ_d is a threshold value, (usually equal to 1). Figure 2 shows thresholding results for some highly compressed depth maps. We include this metric to our experiments in an attempt to check its applicability for comparing post-filtering methods. For this metric, lower value means better quality.

Depth consistency

Analysing the *BAD* metric, one can notice that the thresholding imposed there, does not emphasize the importance of small or big differences. It is equally important, when the error is just a quantum above the threshold and when it is quite high.

In a case of depth degraded by compression artifacts, almost all pixels are quantized thus changing their original values and therefore causing the *BAD* metric to show very low quality while the quality of the rendered views will not be that bad.

Starting from the idea that the perceptual quality of rendered view will depend more on the amount of geometrical distortions than on the number of bad depth pixels, we suggest to give preference to areas where the change between ground truth depth and compressed depth is *more abrupt*. Such changes are expected to cause perceptually high geometrical distortions.

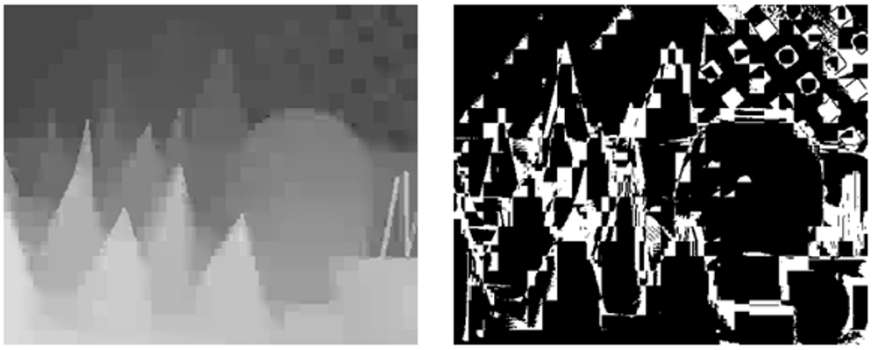
Consider the gradient of the difference between true depth and approximated depth $\nabla\xi = \nabla(z - \hat{z})$. By *depth consistency* we denote the percentage of pixels, having magnitude of that gradient higher than a pre-specified threshold.

$$CONSIST = \frac{100}{N}\sum\left(\|\nabla\xi\|_2 > \sigma_{consist}\right). \tag{5}$$

The measure favors non-smooth areas in the restored depth considered as main source of geometrical distortion, as illustrated in Figure 3.

Gradient-Normalized RMSE

As suggested in [13], the performance of optical flow estimation algorithms can be evaluated using gradient-



(a) (b)

Figure 2 BAD pixels mask for "Cones" dataset $\left(\left|z(x)-\hat{z}(x)\right|>\Delta_d\right)$ caused by H.264 Intra compression with QP = 51.

normalized *RMSE* metric. Such measure decreases the over-penalization of errors caused by fine textures.

In our implementation, we calculate this metric for the luminance channels of reference and rendered views and exclude occluded areas determined by the DIBR on the ground truth data.

$$NRMSE_{\eta}=\left[\sum_{x\in X_v}\frac{\left(\eta^{Y}(x)-\hat{\eta}^{Y}(x)\right)^{2}}{\left\|\nabla\eta^{Y}(x)\right\|^{2}+1}\right]^{1/2}, \tag{6}$$

where $\boldsymbol{\eta}^{Y}(\boldsymbol{x})$ is the luminance of the virtual image generated by ground truth depth and $\hat{\boldsymbol{\eta}}^{Y}(\boldsymbol{x})$ is the luminance of virtual image generated by processed depth. For better quality, the metric shows low values.

Discontinuity Falses

We propose using a measure based on counting of wrong occlusions in the view rendered out of processed depth. If all occlusions between true and processed virtual images coincide, then depth discontinuities are preserved correctly.

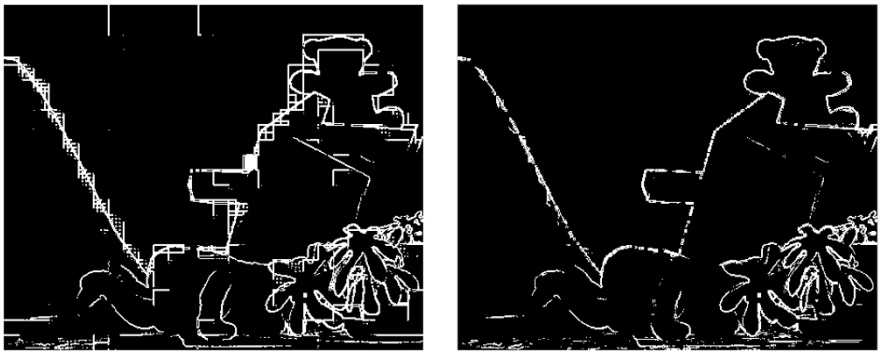
$$DISC=\frac{100}{N}\#\left(\left(X_o\cup\hat{X}_o\right)\setminus\left(X_o\cap\hat{X}_o\right)\right), \tag{7}$$

where $\#(X)$ is cardinality (number of elements) of a domain X. The measure decreases with improving the quality of the processed depth.

3 Depth filtering approaches

A number of post-processing approaches for restoration of natural images exist [14]. However, they are not directly applicable to range images due to differences in image structure.

In this section, we consider several existing filtering approaches and modify them for our need. First group of approaches works on the depth map images with using no structural information from the available color channel. Gaussian smoothing and H.264 in-loop deblocking filter [15] are the filtering approaches included in this group. The approaches of the second group actively use available color frame to improve depth map quality. While there is an apparent



(a) (b)

Figure 3 Distortions in depth "Teddy" dataset ($||\nabla\xi||_2>\sigma_{\text{consist}}$) caused by H.264 Intra compression with (a) QP = 51 and the same after de-blocking with super-resolution filter (b).

correlation between the color channel and the accompanying depth map, it is important to characterize which color and structure information can help for depth processing.

More specifically, we optimize state-of-the-art filtering approaches, such as *local polynomial approximation* (LPA) [16] and bilateral filtering [17] to utilize edge-preserving structural information from the color channel for refining the blocky depth maps. We suggest a new version of the LPA approach which, according to our experiments, is most appropriate for depth map filtering. In addition, we suggest an accelerated implementation of the method based on hypothesis filtering as in [5], which shows superior results for the price of high computational cost.

3.1 LPA approach

The anisotropic LPA is a pixel-wise method for adaptive signal estimation in noisy conditions [16]. For each pixel of the image, local sectorial neighborhood is constructed. Sectors are fitted for different directions. In the simplest case, instead of sectors, 1D directional estimates of four (by 90 degrees) or eight (by 45 degrees) different directions can be used. The length of each sector, denoted as *scale*, is adjusted to meet the compromise between the exact polynomial model (low bias) and sufficient smoothing (low variance). A statistical criterion, denoted as *intersection of confidence intervals* (ICI) rule is used to find this compromise [18,19], i.e., the *optimal scale* for each direction. These optimal scales in each direction determine an anisotropic star-shape neighborhood for every point of the image well adapted to the structure of the image. This neighborhood has been successfully utilized for shape-adaptive transform-based color image de-noising and de-blurring [14].

In the spirit of [14], we use the quantized luminance channel $y_q^Y = y^Y + \varepsilon^Y$ as source of structural information. The image is convolved with a set of 1D directional polynomial kernels $\{g_{h_j,\theta_k}\}$, where $\{h\}_{j=1}^{J}$ is the set of different lengths (scales) and $\theta_k = k\frac{\pi}{4}, k = 1, 2, \ldots, 8$ are the directions, thus obtaining the estimates $y_{h_j,\theta_k}(x) = \left(y_q^Y * g_{h_j,\theta_k}\right)(x)$. The ICI rule helps to find the optimal scale $h_+(x)$ for each direction (the notation of direction is omitted). This is the largest scale (in number of pixels), which ensures a non-empty *ICI* [18,19] $J_i = \cap_{i=1}^{j} I_i$ where

$$I_i = \left[y_{h_i}(x) - \Gamma\sigma^Y \left\|g_{h_i}\right\|, y_{h_i}(x) + \Gamma\sigma^Y \left\|g_{h_i}\right\|\right]. \tag{8}$$

After finding optimal scales $h_+(x)$ for each direction at pixel $\boldsymbol{x}_0$ a star shape neighborhood Ω_{x_0} is formed, as illustrated in Figure 4a. There is a clear evidence that there is a relation between the adaptive neighborhoods and the (distorted) depth, as examplified in Figure 4b. Adaptive neighborhoods are formed for every pixel in the image domain $\boldsymbol{X}$. Once adaptive neighborhoods are found, one must find some modeling for depth channel before utilizing this structural information.

Constant depth model

For our initial implementation of LPA-ICI depth filtering scheme, the depth model is rather simple. The depth map is assumed to be constant in the neighborhood of the filtering pixel $\boldsymbol{x}_0$, where neighborhood is found by the LPA-ICI procedure. This modeling is based on the assumption that the luminance channel is nearly planar at areas where the depth is smooth (close to constant). Whenever the depth has a discontinuity, the luminance is most likely to have a discontinuity too. The constant-modeling results in simple weighted average over the region of optimal neighborhood

$$\forall x_0, \exists\Omega_{x_0}, z_q(x) \approx const, x \in \Omega_{x_0}, \tag{9}$$

$$\hat{z}(x_0) = \frac{1}{N}\sum_{x\in\Omega_{x_0}} z_q(x), \tag{10}$$

where N is the number of pixels inside adaptive support Ω_{x_0}. Note, that the scheme depends on two parameters: the noise variance of the luminance channel σ^Y and the positive threshold parameter Γ. The latter can be adjusted so to control the smoothing in restored depth map.

Linear regression depth model

In a more sophisticated approach we apply pixelwise-planar depth assumption, stating of planarity of depth inside some neighborhood of processing pixel. This is a higher order extension of the previous assumption.

$$\forall x_0, \exists\Omega_{x_0}, z_q(x) \approx A\tilde{x}, \tilde{x} = [x, 1], x \in \Omega_{x_0}, \tag{11}$$

where $\hat{x}$ is homogeneous coordinate.

Based on this assumption, instead of simple averaging in depth domain we apply *plane fitting* (linear regression). $A = \boldsymbol{d}B^{-1}$, where $\boldsymbol{d}$ is a row-vector of depth values $z(x), x \in \Omega_{x_0}$, B is a 3-by-N matrix of their homogeneous coordinates in image space and B^{-1} is Moore-Penrose pseudoinverse of rectangular matrix. Estimated depth values are found with a simple linear equation:

$$\hat{z}(x) = A\tilde{x}, \tilde{x} = [x, 1], x \in \Omega_{x_0}. \tag{12}$$

Aggregation procedure

Since for each processed pixel we may have multiple estimates due to overlapping neighborhoods, we aggregate them as follows:

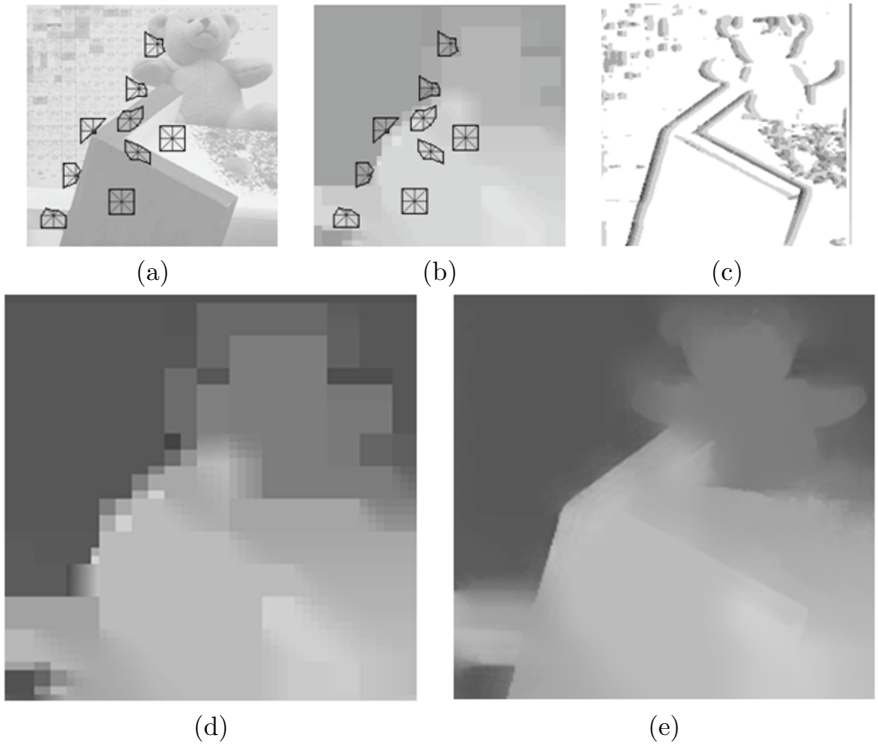


Figure 4 Example of adaptive neighborhoods: **(a)** luminance channel with some of found optimal neighborhoods; **(b)** compressed depth with the same neighborhoods overlaid; **(c)** optimal scales for one of the direction (black for small scale and white for big scale); **(d)** example of highly compressed depth map; **(e)** the same depth map filtered with LPA-ICI (constant depth model with aggregation).

$$\hat{z}^{agg}(x_0) = \frac{1}{M} \sum_{j=1..M} \hat{z}^j(x_0), \tag{13}$$

where M is number of estimates coming from overlapping regions in particular coordinate $\boldsymbol{x}_0$. A result of depth, filtered by LPA-ICI is given in Figure 4e.

Color-driven LPA-ICI

Luminance channel of the color image is usually considered as the most informative channel for processing and also as the most distinguishable by the human visual system. That is why many image filtering mechanisms use color transformation to extract luminance and then process it in different way to compare with chrominance channels. This also may be explained by the fact that luminance is usually the less noisy component and thus it is most reliable. Nevertheless, for some color processing tasks pixel differentiation based only on luminance channel is not appropriate due to some colors may have the same luminance whereas they have different visual appearance.

Our hypothesis is that a color difference signal will better differentiate color pixels, as illustrated in Figure 5. L_2 norm is used to form a color difference map around pixel $\boldsymbol{x}_0$:

$$C_x^{x_0} = \sqrt{(Y_{x_0} - Y_x)^2 + (U_{x_0} - U_x)^2 + (V_{x_0} - V_x)^2}, \tag{14}$$

where $\boldsymbol{x}_0$ is the currently processing pixel, and $x \in \Omega_{x_0}$.

The color difference map is used as a source of structural information, i.e., the LPA-ICI procedure is run over this map instead over the luminance channel. Differences are illustrated in Figure 5. In our implementation, we calculate color-difference only for those pixels of the neighborhood which participate in 1D directional convolutions. Additional computational cost of such implementation is about 10% of the overall LPA-ICI procedure.

For all mentioned LPA-ICI based strategies the main adjusting parameter, capable to set proper smoothing for varying depth degradation parameter (e.g., varying QP in coding) is the parameter Γ.

3.1.1 Comparison of LPA-ICI approaches The performance of different versions of the LPA-ICI approach are compared in Figure 6. The 'normalized RMSE' (equation 6) and 'depth consistency' (equation 5) metrics have been computed and averaged over a set of test images. The parameters of the filters were empirically optimized with 'depth consistency' (equation 5) as a cost measure. As it can be seen, the color-driven LPA-ICI approach

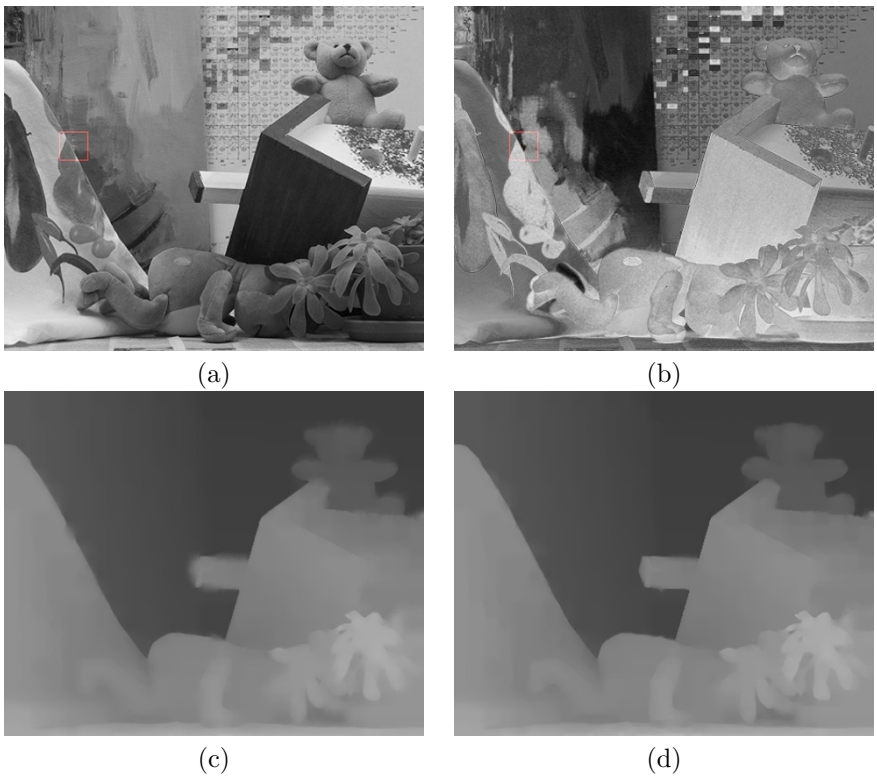


Figure 5 Example of different LPA-ICI implementations: **(a)** luminance channel; **(b)** color-difference channel for central pixel of a red square; **(c)** LPA-ICI filtering result, optimal scales were found in luminance channel; **(d)** LPA-ICI filtering result, optimal scales were found in color-difference channel.

with plane fitting and encapsulated aggregation is the best performing approach, while also having the most stable and consistent results. Because of the superior performance of color-driven LPA-ICI, we use it in the experiments from now on. All experiments and comparisons involving LPA-ICI presented in the following sections refer to the optimized color-driven LPA-ICI implementation.

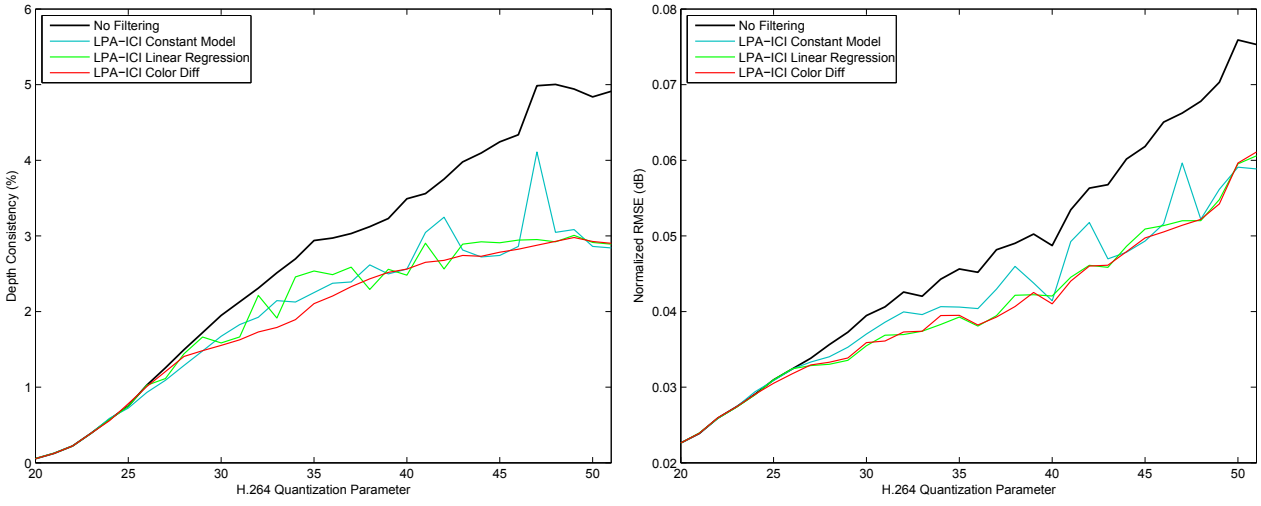


Figure 6 Comparison of different LPA-ICI approaches.

3.2 Bilateral filter

The bilateral filter is a non-linear filter which smooths the image while preserves strong edges [17]. Filtered pixel value is obtained by weighted averaging of its neighborhood combined with color weighting. For gray-scale images, filter weights are calculated based on both spatial distance and photometric similarity, favoring near values to distant values in both spatial domain and range. For color images, bilateral filtering uses color distance to distinguish photometric similarity between pixels, which affects in reducing phantom colors in the resulting image. In our approach, we calculate filter weights using information from color frame in RGB, while applying filtering on depth map. Our design of bilateral filter has been inspired by [5], as follows:

$$\hat{z}(x) = \frac{\sum_u \omega_s(\|x-u\|)\,\omega_c\left(\|\gamma(x)-\gamma(u)\|\right) z_q(u)}{\sum_u \omega_s(\|x-u\|)\,\omega_c\left(|\gamma(x)-\gamma(u)|\right)}, \quad (15)$$

where $\omega_a(t) = e^{-\frac{t}{\gamma_a}}, (a = s, c)$ and $\boldsymbol{u} \in \Omega_{\boldsymbol{x}}$ are neighborhood pixels of point $\boldsymbol{x}$. This design allows for relatively fast implementation by storing all possible color and distance weights as look-up tables. Parameters γ_s, γ_c and processing window size $\Omega_{\boldsymbol{x}}$ are adjustable parameters of the filter. Figure 7 illustrates the filtering. The color channel (Figure 7a) provides the color difference information, with respect to the processed pixel position (Figure 7b). It is further weighted by spatial Gaussian filter to determine the weights of pixels from the depth map taking part in estimating the current (central) pixel (Figure 7f).

3.3 Spatial-depth super resolution approach

A post-processing approach was suggested aimed at increasing the resolution of low-resolution depth images, given high-resolution color image as a reference [5]. In our study, we study the applicability of this filter for suppression of compression artifacts and restoration of true discontinuities in the depth map. The main idea of the filter is to process depth in probabilistic manner, constructing 3D cost volume from several depths hypothesizes. After bilateral filtering of each slice of the volume, the hypothesis with the lowest cost is selected as a new depth value. The procedure is applied iteratively, calculating cost volume using the depth estimated in previous step. The cost volume on ith iteration is constructed to be quadratic function of the current depth estimate:

$$C_{(i)}(x,d) = \min\left\{L_\eta, \left(d - z_{(i)}(x)\right)^2\right\}, \quad (16)$$

where L_η denotes tunable search range.

The bilateral filtering, defined as in Equation 15 enforces an assumption of piecewise smoothness. The procedure is illustrated in Figures 8 and 9. The approach resembles the local depth estimation idea, where a volumetric cost volume is further aggregated with bilateral filter.

Since cost function is discrete on d, the depth obtained by winner-takes-all approach will be discrete as well. To tackle this effect, the final depth estimate is taken as the minimum point of quadratic polynomial which approximates the cost function between three discrete depth candidates: d, d - 1 and d + 1

$$f(d) = ad^2 + bd + c, \quad (17)$$

$$d_{\min} = -\frac{b}{2a}. \quad (18)$$

$f(d_{min})$ is the minimum of quadratic function $f(d)$, thus given d, $f(d)$, $f(d$ -1) and $f(d$ + 1), value d_{min} can be calculated:

$$d_{min} = d - \frac{f(d+1) - f(d-1)}{2(f(d+1) - f(d-1) - 2f(d))}. \quad (19)$$

After the bilateral filtering is applied to the cost volume, the depth is refined and true depth discontinuities might be completely recovered.

In our implementation of the filter, we have suggested two simplifications:

- we use only one iteration of the filter;
- before processing we scale the depth range by factor of 20, thus reducing the number of slices, and subsequently reducing the processing time.

The main tunable parameters of the filter are the parameters of the bilateral filter γ_d and γ_c. As long as the processing time of the filter still remains extremely high, we do not perform optimization of this filter directly, but assume that the optimal parameters $\gamma_d = f_d(QP)$ and $\gamma_c = f_c(QP)$ found for the direct bilateral filter are optimal or nearly optimal for this filter as well.

3.4 Practical implementation of the super resolution filtering

In this section, we suggest several modifications to the original approach to make it more memory-efficient and to improve its speed. It is straightforward to figure out that there is no need to form cost volume in order to obtain the depth estimate for a given coordinate $\boldsymbol{x}$ at the ith iteration. Instead, the cost function is formed for the required neighborhood only and then filtering applies, i.e.,

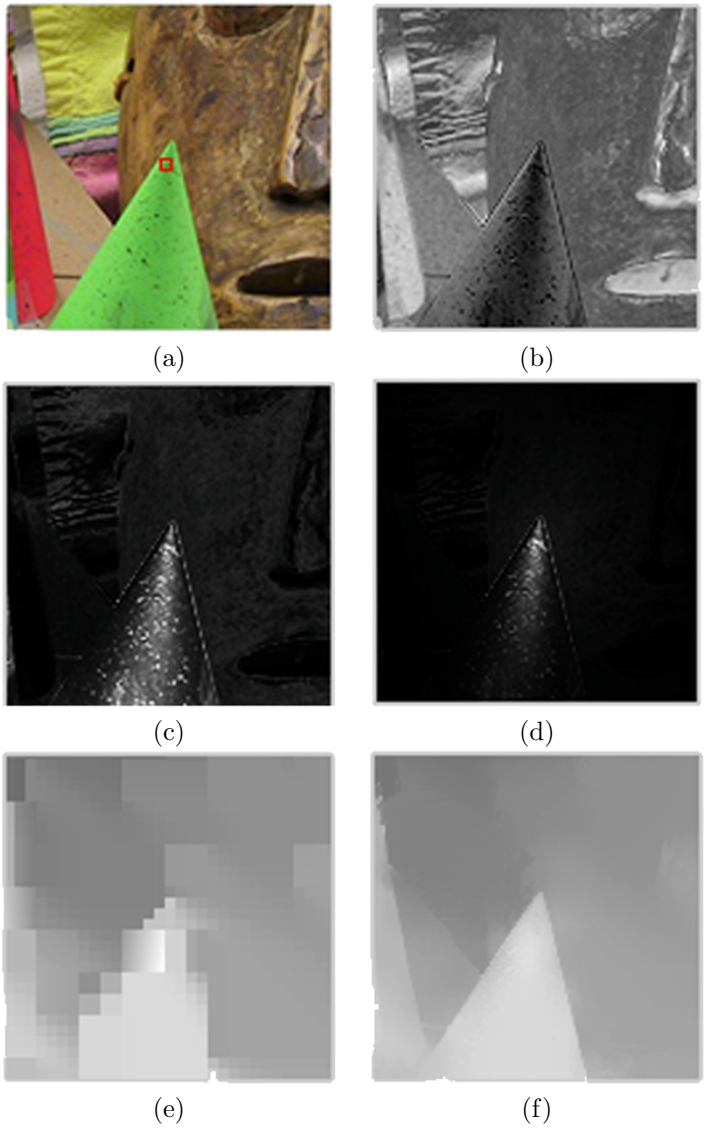


Figure 7 Example of bilateral filtering: **(a)** color channel; **(b)** color-difference for central pixel (marked red); **(c)** color-weighed component of bilateral product; **(d)** complete weights for selected window ((c) multiplied with spatial Gaussian component); **(e)** example of blocky depth map for the same scene; **(f)** the same block filtered with bilateral filter.

$$\begin{aligned}\hat{z}_{(i+1)}(x) &= \arg\min_{d}\left[\frac{\sum_{u\in\Omega_x} W(x,u)G(u,d)}{\sum_{u\in\Omega_x} W(x,u)}\right],\\ W(x,u) &= \omega_s\left(\|x-u\|\right)\omega_c\left(\left\|\gamma(x)-\gamma(u)\right\|\right),\\ G(u,d) &= \min\left\{\eta * L, \left(d-z_{(i)}(u)\right)^2\right\}.\end{aligned} \tag{20}$$

Furthermore, the computation cost is reduced by assuming that not all depth hypotheses are applicable for the current pixel. A safe assumption is that only depths within the range $d \in [d_{\min}, d_{\max}]$ where $d_{\min} = \min(z(\boldsymbol{u}))$, $d_{\max} = \max(z(\boldsymbol{u}))$, $u \in \Omega_{\boldsymbol{x}}$ have to be checked.

Additionally, depth range is scaled with the purpose to further reduce the number of hypothesizes. This step is especially efficient for certain types of distortions such as compression (blocky) artifacts. For compressed depth maps, the depth range appears to be sparse due to the quantization effect.

Figure 10 illustrates histograms of depth values before and after compression so to confirm the use of rescaled search range of depth hypotheses. This modification speeds up the procedure and relies on the subsequent quadratic interpolation to find the true minimum. A pseudo-code of the suggested procedure in Equation 20 is given in following listing.

Require: C, the color image; D, the depth image; X, a spatial image domain

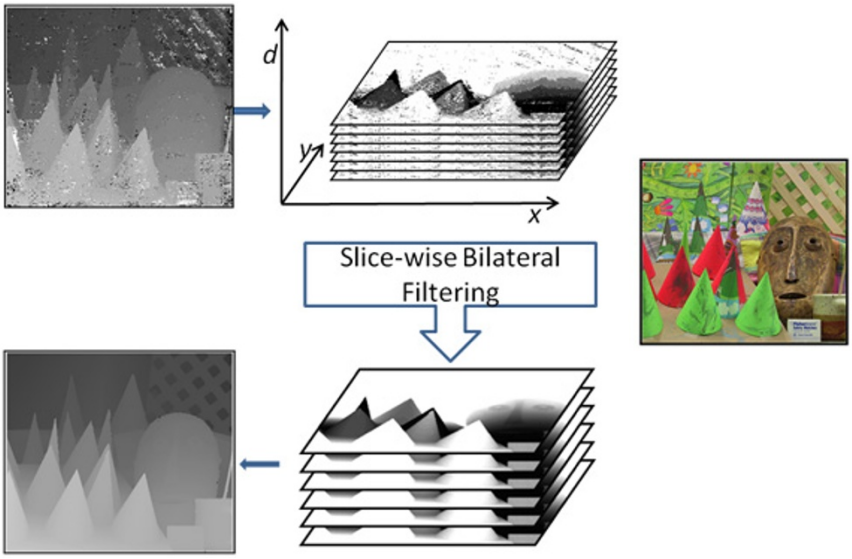


Figure 8 Principle of super-resolution filtering.

for all $x \in X$ **do**
  $d_{\min} = \min_{u \in \Omega_x} D_u$, $d_{\max} = \max_{u \in \Omega_x} D_u$
  if $d_{\max}$ - $\boldsymbol{d}_{\min} < \gamma_{\text{thr}}$ **then**
    $\hat{D}_x = D_x$
  else
    $F(x,u) = \frac{\|C_u - C_x\| \, \|u - x\|}{\sum_u \|C_u - C_x\| \, \|u - x\|}$ {bilateral weights}
    $S_{\text{best}} \leftarrow S_{\max}$ {$S_{\max}$ is maximum reachable value for S}

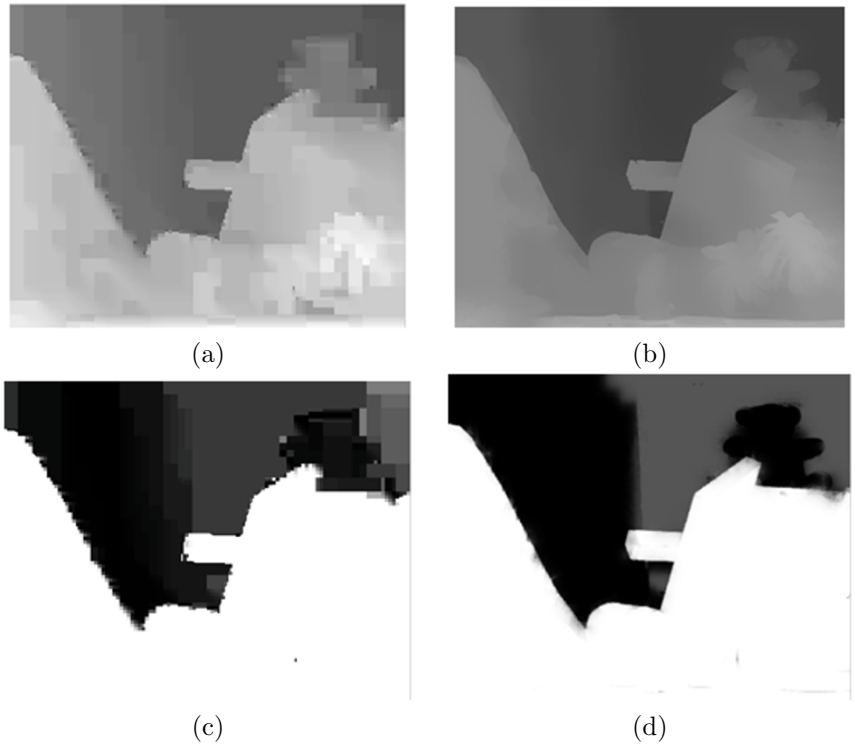


Figure 9 Super-resolution filtering example. **(a)** Compressed depth map; **(b)** same depth, processed by super-resolution filter; **(c)** single cost volume slice, constructed from **(a)**; **(d)** result of bilateral filtering for same slice.

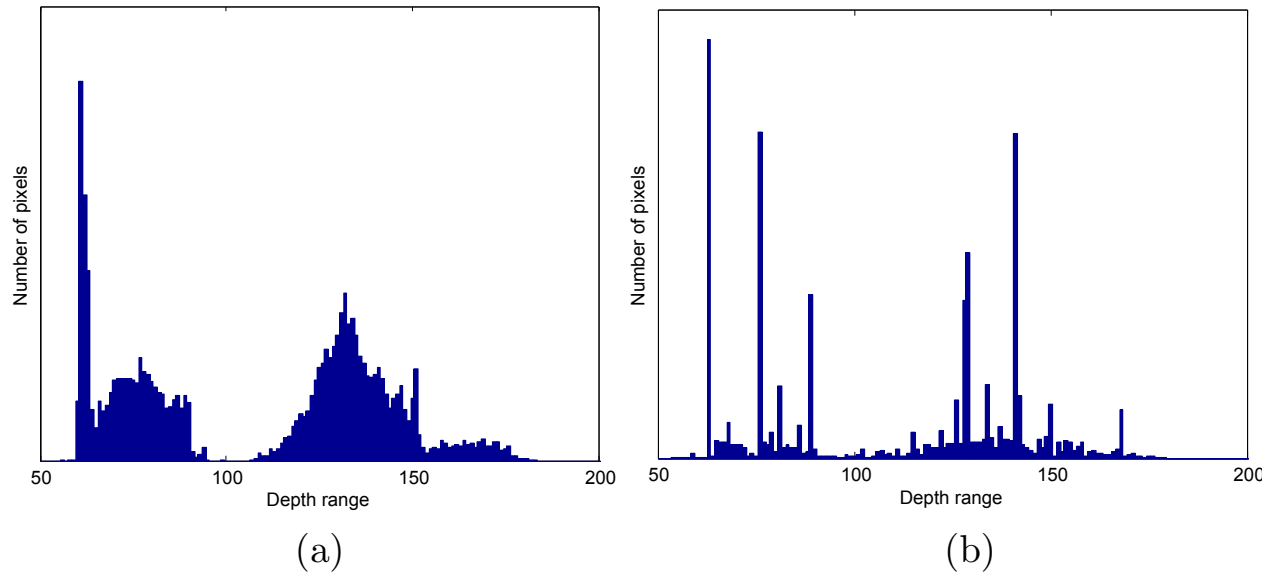


Figure 10 Histograms of non-compressed (a) and compressed (b) depth maps.

```
for d = ⌊d_min⌋ to [d_max] do
  S ← 0
  for all u ∈ Ω_x do
    E ← min{(d - D_u)^2, ηL}
    S ← S + F(x, u) * E
  end for
  if S < S_best then
    S_best ← S
    d_best ← d
  end if
end for
D̂_x = d_best
end if
end for
```

The memory foot-print required by our implementation is significantly lower than the one imposed by a direct implementation. A straightforward implementation would require a large memory buffer to store the complete cost volume in order to process it pixel-by-pixel and avoid computing (the same) color weights across different slices. In the proposed implementation, two memory buffers with relatively low sizes are required: a memory buffer which is equal to the processing window size to store current color weights, and a buffer to store the cost values for the current pixel along the '*d*' dimension. In case of multi-thread (parallelized) implementation, these memory buffers are multiplied by the number of processing threads. More information about platform-specific optimization of the proposed algorithm is given in [20].

Figure 11 illustrates the performance in terms of speed. The figure shows experiments with different implementations of the filtering procedure. The 'Teddy' dataset has been processed (see also Figure 12 for reference). All filter versions have been implemented in C and then compiled into MEX files to be run within Matlab environment. The experiments have been run on a 1.3 GHz Pentium Dual-Core processor with 1 Gb of RAM under MS Windows XP operating system. In the figure, the vertical axis shows the execution time in seconds and the horizontal line shows the number of slices processed (i.e., the depth dynamic range assumed). The dotted curve shows single-pass bilateral filtering. It does not depend on the dynamic range, but on the window size, thus it is a constant in the figure. The red line shows the computational time for the original approach implemented as a three step procedure for the full dynamic range. Naturally, it is a linear function with respect to the slices to be filtered. Our implementation (blue curve) applying a reduced dynamic range is also linearly depending on the number of slices, but with dramatically reduced steepness.

4 Experimental results

4.1 Experimental setting

In our experiments, we consider depth maps degraded by compression. Thus degradation is characterized by the quantization parameter (QP). For better comparison of selected approaches, we present two types of experiments. In the first set of experiments, we compare the performance of all depth filtering algorithms assuming the true color channel is given (it has been also used in the optimization of the tunable parameters). This shows ideal filtering performance, while in practice it cannot

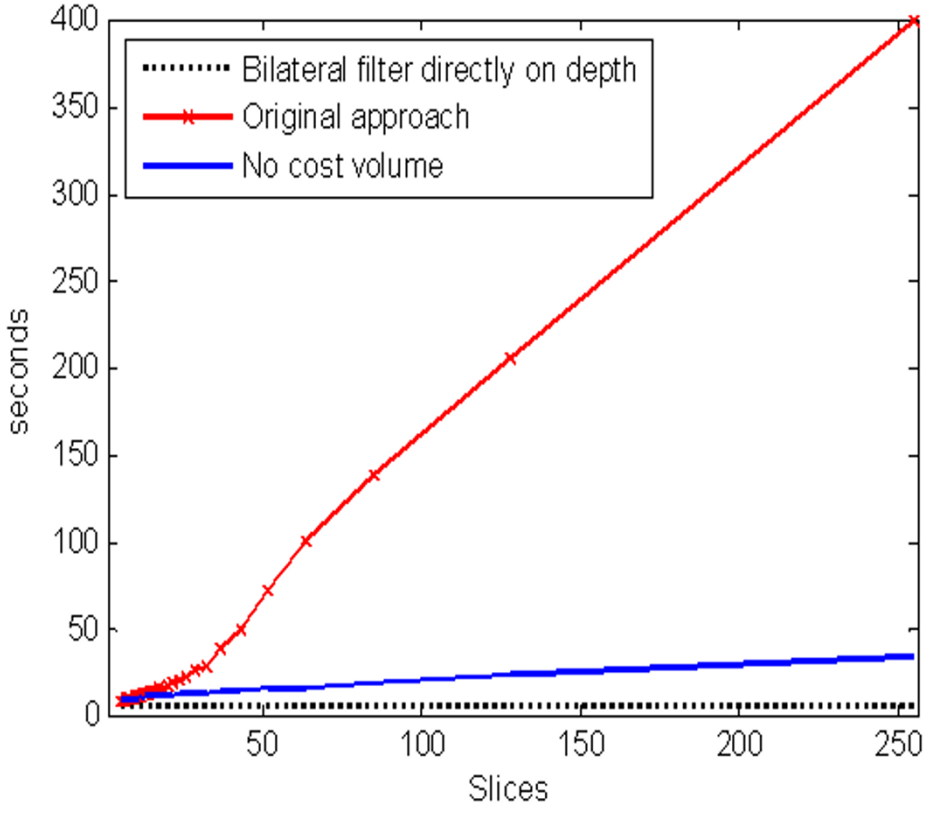


Figure 11 Execution time of different implementations of filtering approach.

be achieved due to the fact that the color data is also degraded by e.g., compression.

In the second set of experiments, we compare the effect of depth filtering in the case of mild quantization of the color channel. General assumption is that color data is transmitted with backward compatibility in mind, and hence most of the bandwidth is occupied by the color channel. Depth maps in this scenario are heavily compressed, to consume not more than 10-20% of the total bit budget [21,22].

We consider the case where both $\boldsymbol{y}$ and z are to be coded as H.264 intra frames with some QPs, which leads to their quantized versions $\boldsymbol{y}_q$ and z_q. The effect of quantization of DCT coefficients has been studied thoroughly in the literature and corresponding models have been suggested [23]. Following the degradation model in Section 2.2, we assume quantization noise terms added to the color channels and the depth channel considered as independent white Gaussian processes: $\varepsilon^C(\cdot) \sim N(0, \sigma_C^2), \varepsilon(\cdot) \sim N(0, \sigma^2)$. While this modeling is simple, it has proven quite effective for mitigating the blocking artifacts arising from quantization of transform coefficients [14]. In particular, it allows for establishing a direct link between the QP and the quantization noise variance to be used for tuning deblocking filtering algorithms [14].

Training and test datasets for our experiments (see Figure 12) were taken from Middlebury Evaluation Testbench [12,24,25]. In our case, we cannot tolerate holes and unknown areas in the depth datasets, since they produce fake discontinuities and unnatural artifacts after compression. We semi-manually processed 6 images to fill holes and to make their width and height be multiples of 16.

4.1.1 Parameters optimization

Each tested algorithm has a few tunable parameters which could be modified according particular filtering strategy related with a quality metric. So, to make comparison as fair as possible, we need to tune each algorithm to its best, according such a strategy and within certain range of training data.

Our test approach is to find empirically optimal parameters for each algorithm over a set of training images. It is done separately for each quality metric. Then, for each particular metric we evaluate it once more on the set of test images and then average. Then comparison between algorithms is done for each metric independently.

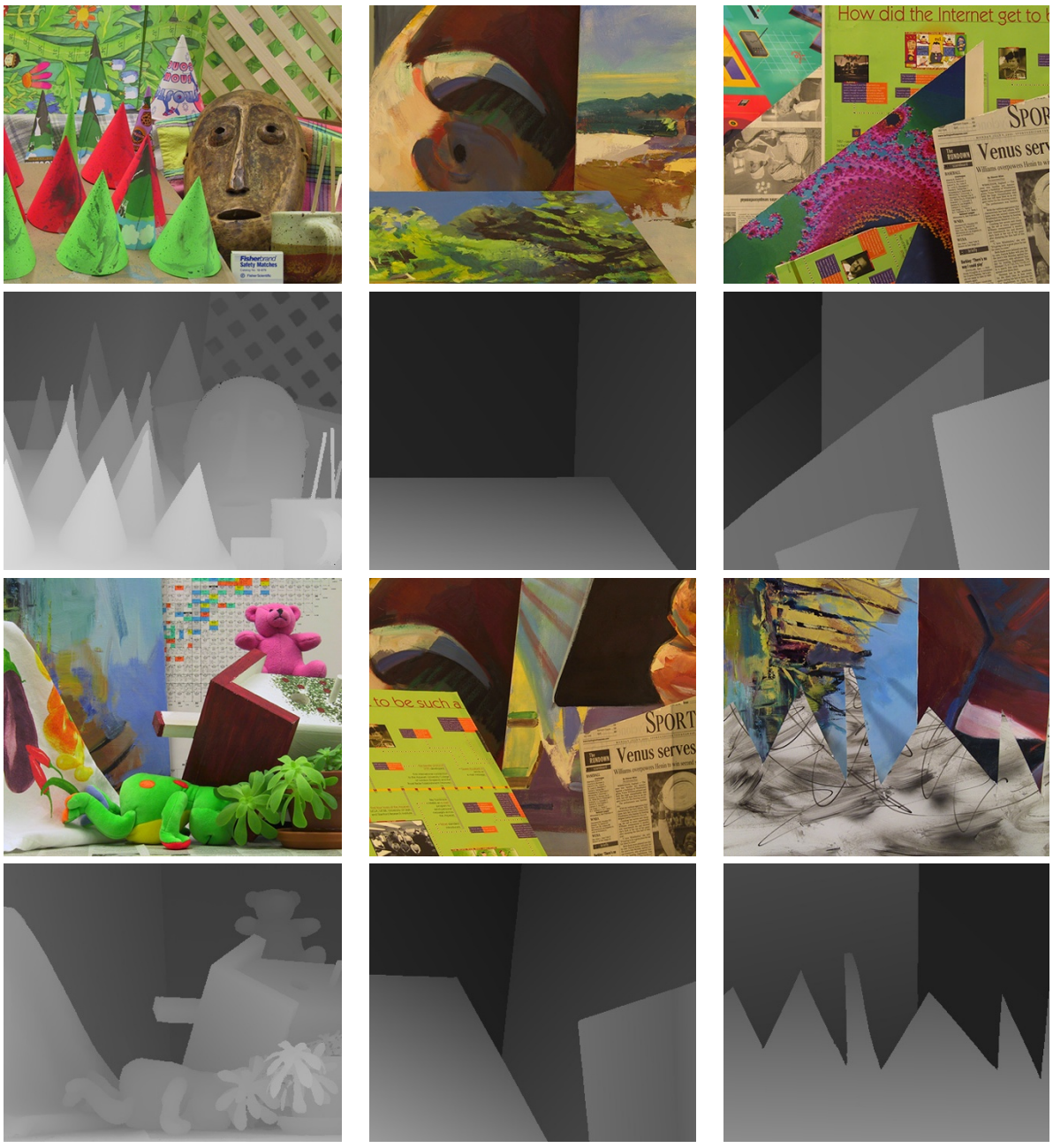


Figure 12 Training (upper row) and testing (lower row) datasets used in study.

Particularly, for bilateral filtering and hypothesis (super-resolution) filtering we are optimizing the following parameters: processing window size, γ_s and γ_c. For the Gaussian Blurring we are optimizing parameters σ and processing window size. For LPA-ICI based approach we are optimizing the Γ parameter.

4.2 Visual comparison results

Figures 13 and 14 present depth images paired with consecutive rendered frames (no occlusion filling is applied). This approach helps to illustrate artifacts in the depth channel as well as their effect on the rendered images.

As it is seen in the top row (a), rendering with true depth, results in sharp and straight object contours, as well as in continuous shapes of occlusion holes. For such holes, a suitable occlusion filling approach will produce good estimate.

Row (b) shows unprocessed depth after strong compression (H.264 with QP = 51) frame and its rendering

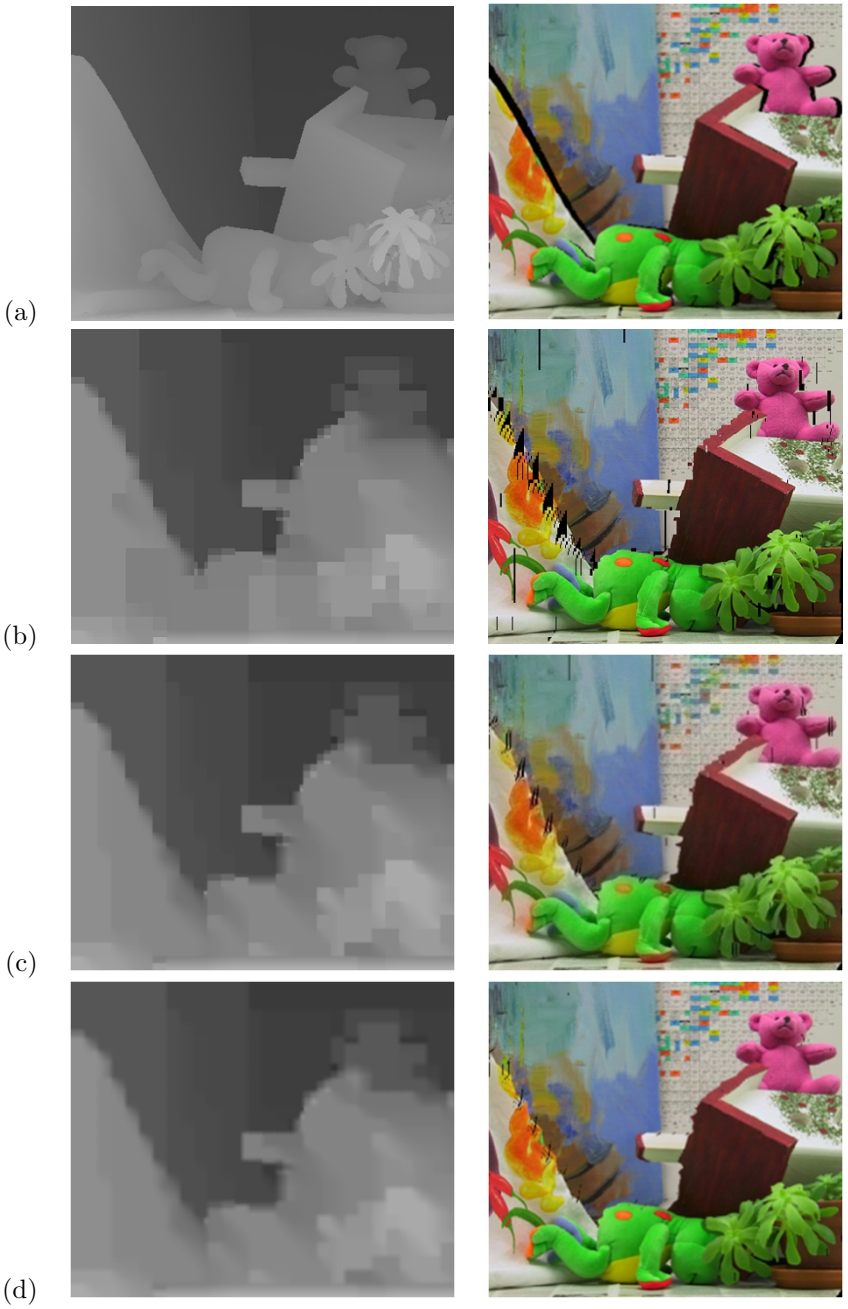


Figure 13 Visual results for "Teddy" dataset. Left column: depth, right column: respective rendered result (no occlusion filling applied). **(a)** Ground truth depth; **(b)** depth compressed H.264 Intra with QP = 51; **(c)** loop-filtered depth; **(d)** Gaussian-filtered depth.

capability. Objects edges are particularly affected by block distortions.

With respect to occlusion filling, the methods behave as follows.

- Gaussian smoothing of depth images is able to reduce number of occluded pixels, making occlusion filling simpler. Nevertheless, this type of filtering does not recover geometrical properties of depth, which results in incorrect contours of the rendered images.
- Internal H.264 in-loop deblocking filtering was performed similarly to the Gaussian smoothing, with no improvement of geometrical properties.
- LPA-ICI based filtering technique performs significantly better both is sense of depth frame and

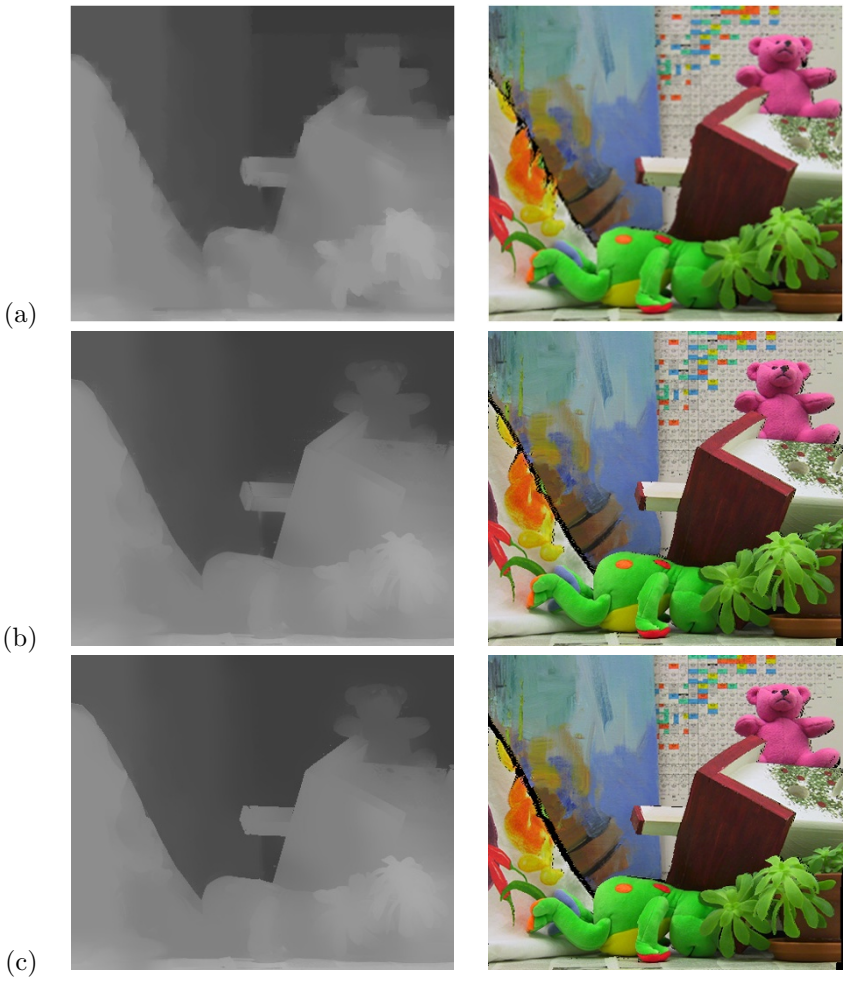


Figure 14 Visual results for "Teddy" dataset (continued). **(a)** LPA-ICI filtered depth; **(b)** bilateral filtered depth; **(c)** super-resolution (modified implementation) filtered.

rendered frame visual quality. Geometrical distortions are less pronounced, however, still visible in rendered channel.

- Bilateral filter almost recovers the sharp edges in depth image, while has minor artifacts (for instance, see chimney of house).
- Super-resolution depth filter recovers discontinuities as good as bilateral or even better. Resulted depth image does not have artifacts as in the previous methods. Geometrical distortions in rendered image are not pronounced.

Among all filtering results, the latter one contains occlusions which are most similar to the occlusions of the original depth rendering result. Visually, super-resolution depth approach is considered to be the best. The numerically estimated results for all presented approaches are presented in following section.

4.3 Numerical results for ideal color channel

Figure 15 summarizes averaged results over the three test datasets: 'venus', 'sawtooth', and 'teddy'.

X-axis on all the plots represents varying QP parameters of the H.264 Intra coding, while each Y -axis shows a particular metric. On the most of the metric plots it is visible that there is no need to apply any kind of filtering before QP reaches some critical value. Before that value, the quality of the compressed depth is high enough, so no filtering could improve it.

The group of structurally-constrained methods clearly outperforms the simple methods working on the depth image only. The two PSNR-based metrics and the BAD metric seem to be less reliable in characterizing the performance of the methods. The three remained measures, namely depth consistency, discontinuity falses and gradient-normalized RMSE perform in a consistent manner. While Normalized RMSE is perhaps the measure closest

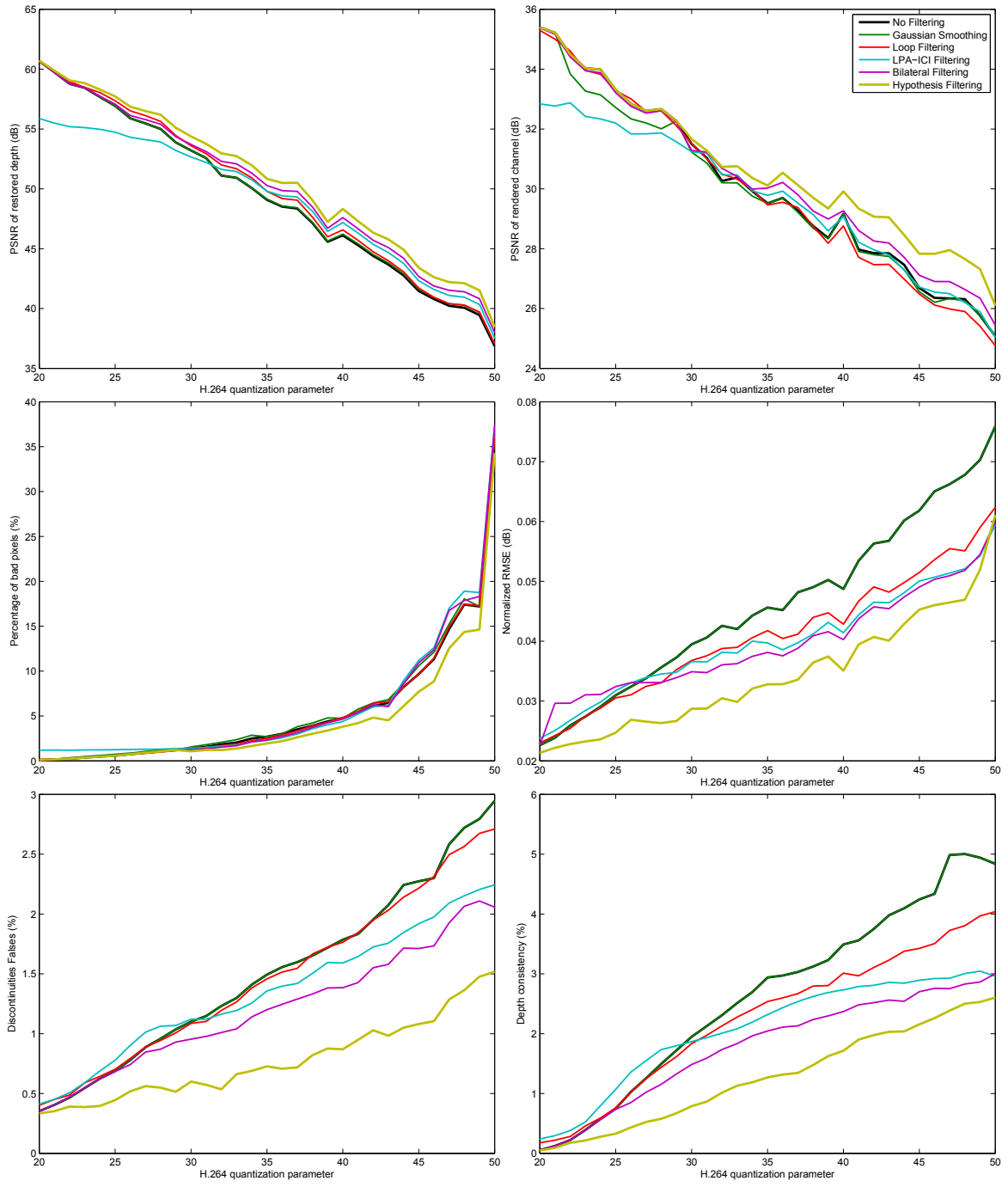


Figure 15 Performance results averaged between test datasets. All metrics are optimized for best performance.

to the subjective perception, we favor also the other two measures of this group as they are relatively simple and do not require calculation of the warped (rendered) image.

4.4 Numerical results for compressed color channel

So far, we have been working with uncompressed color channel. It has been involved in the optimizations and comparisons. Our aim was to characterize the pure influence of the depth restoration only.

In practice, when 'color-plus-depth' frame is compressed and then transmitted over a channel, the color frame is also compressed with a pre-specified bit-rate, aiming at maximizing visual quality of the video. Transmission of 'color plus depth' stream has also to be

Figure 16 Visual results of experiments with compressed color channel (testing dataset). First row: compressed color channels, second row: depth filtered with hypothesis filter using compressed color channel, third row: depth filtered with hypothesis filter using true color channel.

constrained within a given bit-budget. Thus, receiver-side device has to cope with compressed color and compressed depth.

In the second experiment, we assume mild quantization of the color image, e.g., by QP = 30. For our test imagery, the first depth QP corresponds to about 10% of the total bit-rate. 'Depth consistency', 'Discontinuity falses' and 'PSNR of rendered channel' are calculated for different depth maps: compressed, post-filtered with LPA-ICI filtering approach, post-processed with the bilateral filter and post-filtered with our implementation of the super-resolution approach. The resulting numbers are averaged over three dataset images. Visual results of hypothesis filtering are presented in Figure 16 which shows comparison between higly-compressed depth filtered with compressed color (second row) and same, filtered with ideal color (last row). The numerical results are given in Figures 17, 18, and 19. Cases with post-processed depth are marked with color. One can see that the depth postprocessing clearly makes a difference allowing to use stronger quantization of the depth channel and still to achieve good quality.

5 Conclusions

In this article, the problem of filtering of depth maps was addressed and the case of processing of depth map images impaired by compression artifacts was emphasized.

Before proceeding with the actual depth processing task, the characteristics of the representation *view-plus-depth* were overviewed, including methods of depth image based rendering for virtual view generation, and formulation of the depth map filtering problem. In addition, number of quality measures for evaluating the

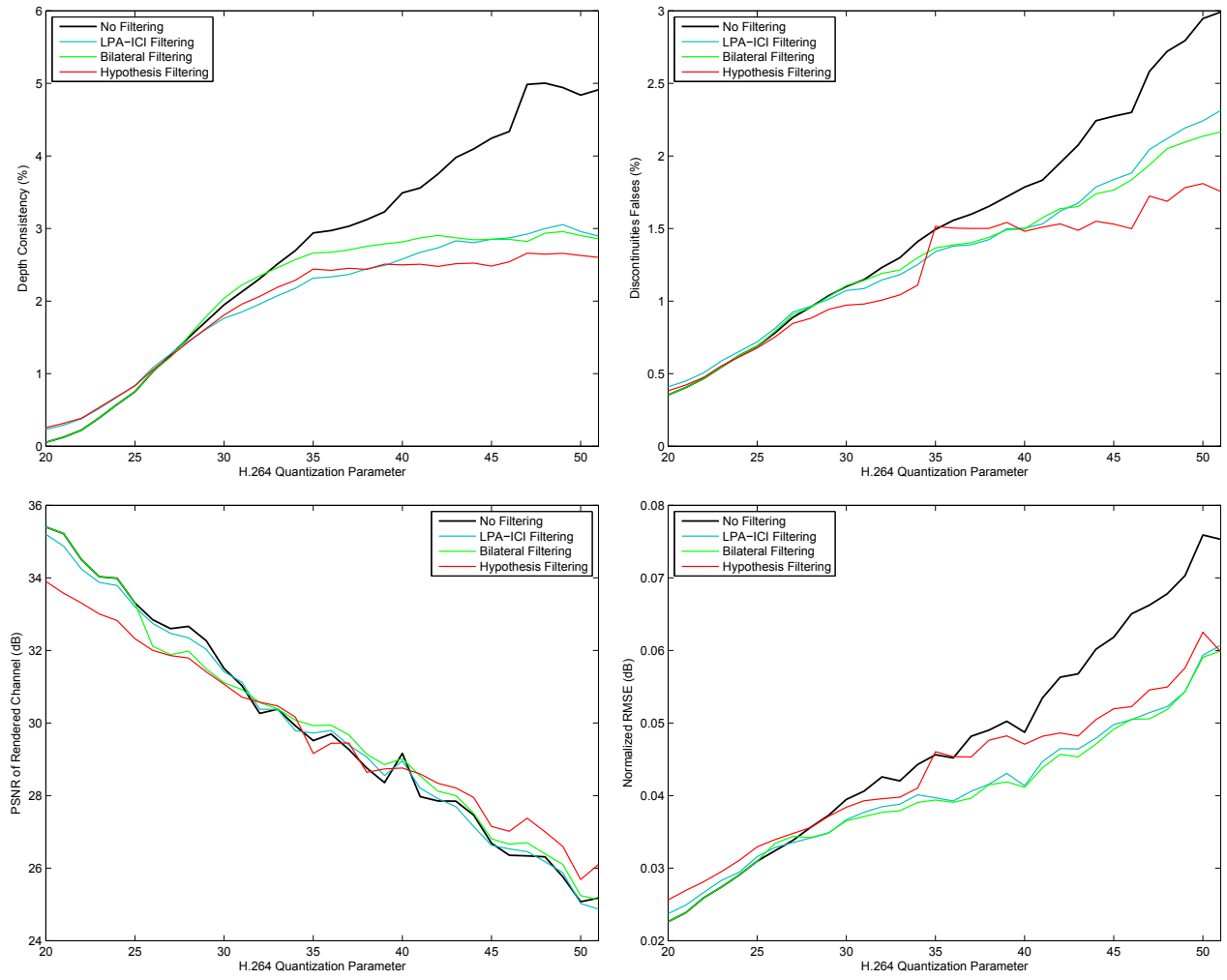


Figure 17 Results of filtering with compressed color. Optimization done by "Depth consistency" metric.

depth quality were studied and new ones were suggested.

For the case of post-filtering of depth maps impaired by compression artifacts, a number of filtering approaches were studied, modified, optimized, and compared. Two groups of approaches were underlined. In the first group, techniques working directly on the depth map and not taking into account the accompanying color frame were studied. In the second group, filtering techniques utilizing structural or color information from the accompanying frame were considered. This included the popular bilateral filter as well as its extension based on probabilistic assumptions and originally suggested for super-resolution of depth maps. Furthermore, the LPA-ICI approach was specifically modified for the task of depth filtering and a few versions of this approach were proposed. The techniques from the second group have shown better performance over all measures used. More specifically, the method based on probabilistic assumptions showed superior results for the price of very high computational cost. To tackle this problem, we have suggested practical modifications leading to faster and higher memory-efficient version which adapts to the true depth range and its structure and is suitable for implementation on a mobile platform. The competitive methods, i.e., LPA-ICI and bilateral filtering, should not be, however, discarded as fast implementations of those do exist as well. They demonstrated competitive performance and thus form a scalable set of algorithms. Practitioners can choose between the algorithms in the second group of methods depending on the requirements of their applications and available computational resources. The de-blocking tests demonstrated that it is possible to tune the filtering parameters depending on the QP of the compression engine. It is also feasible to allocate really small fraction of the total bit budget for compressing the depth, thus allowing for high-quality backward compatibility and

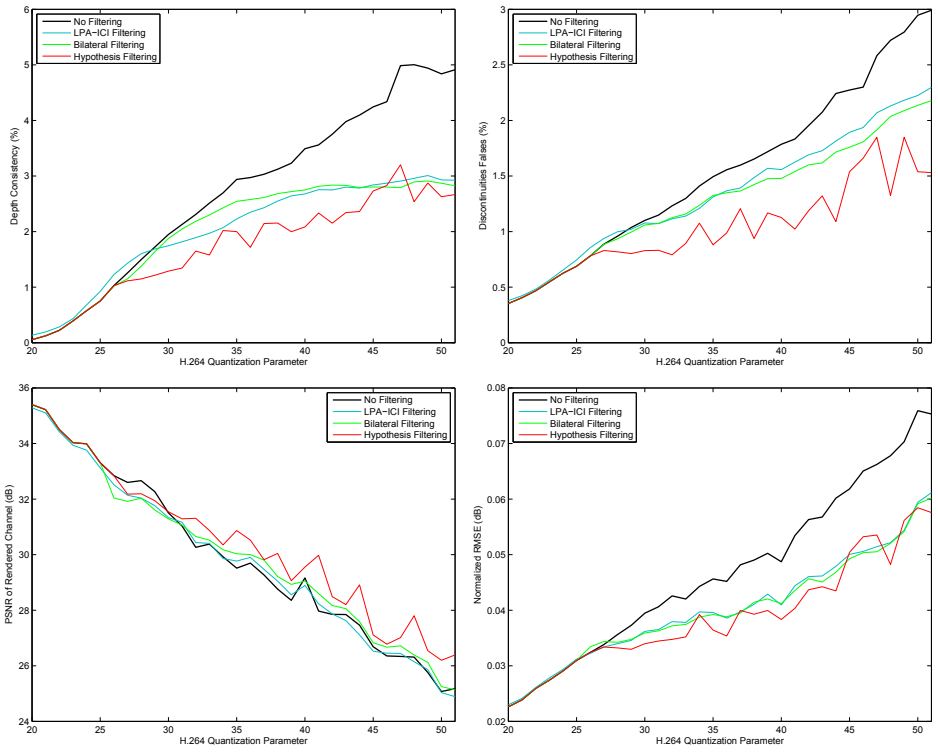


Figure 18 Results of filtering with compressed color. Optimization done by "Normalized RMSE" metric.

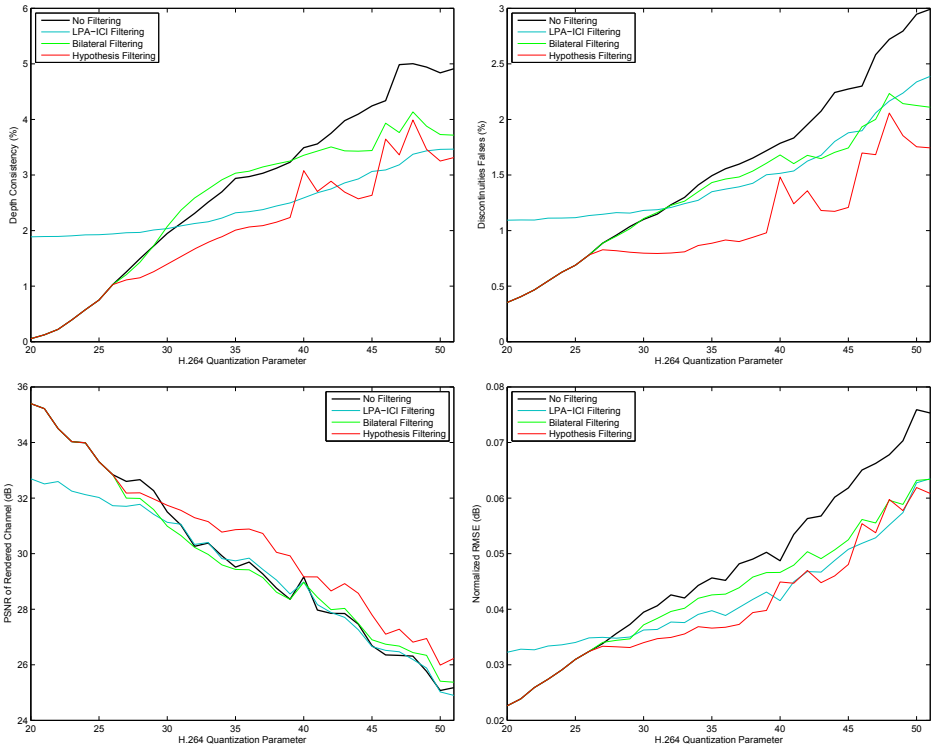


Figure 19 Results of filtering with compressed color. Optimization done by "PSNR of rendered channel" metric.

channel fidelity. The price for this would be some additional post-processing at the receiver side.

Competing interests

The authors declare that they have no competing interests.

Received: 6 June 2011 Accepted: 14 February 2012
Published: 14 February 2012

References

1. K Mueller, P Merkle, T Wiegand, 3-D video representation using depth maps. Proc IEEE. **99**(4), 643–656 (2011)
2. C Fehn, Depth-image-based rendering (DIBR), compression and transmission for a new approach on 3D-TV, Proceedings of SPIE Stereoscopic Displays and Virtual Reality Systems XI, SPIE, **5291**, San Jose, CA, USA, 93–104 (2004)
3. A Vetro, S Yea, A Smolic, Towards a 3D video format for auto-stereoscopic displays, SPIE Conference on Applications of Digital Image Processing XXXI SPIE, **7073**, San Diego, CA, USA, 70730F (2008)
4. S Kang, R Szeliski, J Chai, Handling occlusions in dense multi-view stereo, IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2001), **1**, IEEE Computer Society, Kauai, HI, USA, 103–110 (2001)
5. Y Qingxiong, Y Ruigang, J Davis, D Nister, Spatial-depth super resolution for range images, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, IEEE Computer Society, Minneapolis, MN, 1–8 (2007)
6. J Kopf, M Cohen, D Lischiski, M Uyttendaele, Joint bilateral upsam-pling, ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007), **26**(3), ACM New York, NY, USA, 96.1–96.5 (2007)
7. AK Riemens, OP Gangwal, B Barenbrug, R-PM Berretty, Multistep joint bilateral depth upsampling. Proc SPIE Visual Commun Image Process. **7257**, 72570M (2009)
8. S Smirnov, A Gotchev, K Egiazarian, Methods for restoration of compressed depth maps: a comparative study, in *Proceedings of the Fourth International Workshop on Video Processing and Quality Metrics Consumer Electronics, VPQM 2009*, Scottsdale, Arizona, USA, 6 (14–19 January 2009)
9. A Alatan, Y Yemez, U Gudukbay, X Zabulis, K Muller, CE Erdem, C Weigel, A Smolic, Scene representation technologies for 3DTV-A survey. IEEE Trans Circuits Syst Video Technol. **17**(11), 1587–1605 (2007)
10. P Merkle, Y Morvan, A Smolic, D Farin, K Muller, PHN de With, T Wiegand, The effect of depth compression on multiview rendering quality, in *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video*, Istanbul, 245–248 (2008)
11. A Boev, D Hollosi, A Gotchev, K Egiazarian, Classification and simulation of stereoscopic artifacts in mobile 3DTV content, in *Stereoscopic Displays and Applications XX*, SPIE, **7237**, San Jose, CA, USA, 72371F (2009)
12. D Scharstein, R Szeliski, A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int J Comput Vision. **47**, 7–42. doi:10.1023/A:1014573219977
13. S Baker, D Scharstein, JP Lewis, A database and evaluation methodology for optical flow, in *Proc IEEE Int'l Conf on Computer Vision*, Crete, Greece, 243–246 (2007)
14. A Foi, V Katkovnik, K Egiazarian, Pointwise shape-adaptive dct for high-quality denoising and deblocking of grayscale and color images. IEEE Trans Image Process. **16**(5), 1395–1411
15. P List, A Joch, J Lainema, G Bjntegaard, M Karczewicz, Adaptive deblocking filter. IEEE Trans Circuits Syst Video Technol. **13**(7), 614–619
16. V Katkovnik, K Egiazarian, J Astola, Local Approximation Techniques in Signal and Image Processing, SPIE Press, Monograph **PM157**, ISBN 0-8194-6092-3
17. C Tomasi, R Manduchi, Bilateral Filtering for Gray and Color Images, in *IEEE International Conference on Computer Vision*, Bombay, 839–846 (1998)
18. A Goldenshluger, A Nemirovski, On spatial adaptive estimation of non-parametric regression. Math Meth Statistics. **6**, 135–170
19. V Katkovnik, A new method for varying adaptive bandwidht selection. IEEE Trans Signal Process. **47**(9), 2567–2571. doi:10.1109/78.782208
20. O Suominen, S Sen, S Smirnov, A Gotchev, Implementation of depth map filtering algorithms on mobile-specific platforms, accepted, in *The International Conference on Consumer Electronics (ICCE)*, Las Vegas, USA, IEE, 319–322 (January 13-16, 2012)
21. Y Morvan, D Farin, PHN de With, Depth-imagecompression based on an R-D optimized quadtree decomposition for the transmission of multiview images, IEEE International Conference on Image Processing, **5**, San Antonio, TX, USA , 105–108 (2007)
22. A Tikanmaki, A Smolic, K Mueller, A Gotchev, Quality assessment of 3D Video in rate allocation experiments, in *IEEE International Symposium on Consumer Electronics ISCE 2008*, Algarve, Portugal, 1–4 (2008)
23. M Robertson, R Stevenson, DCT quantization noise in compressed images. IEEE Trans Circuits Syst Video Technol. **15**(1), 25–38
24. D Szeliski, R Scharstein, High-accuracy stereo depth maps using structured light, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), **1**, , Madison, WI, 195–202 (2003)
25. D Scharstein, R Szeliski, Middlebury stereo vision page.

doi:10.1186/1687-6180-2012-25
Cite this article as: Smirnov *et al.*: **Methods for depth-map filtering in view-plus-depth 3D video representation.** *EURASIP Journal on Advances in Signal Processing* 2012 **2012**:25.