

# Depth Map Quality Evaluation for Photographic Applications

Eloi Zalczer, François-Xavier Thomas, Laurent Chanas, Gabriele Facciolo and Frédéric Guichard  
DXOMARK; 24-26 Quai Alphonse Le Gallo, 92100 Boulogne-Billancourt, France

## Abstract

As depth imaging is integrated into more and more consumer devices, manufacturers have to tackle new challenges. Applications such as computational bokeh and augmented reality require dense and precisely segmented depth maps to achieve good results. Modern devices use a multitude of different technologies to estimate depth maps, such as time-of-flight sensors, stereoscopic cameras, structured light sensors, phase-detect pixels or a combination thereof. Therefore, there is a need to evaluate the quality of the depth maps, regardless of the technology used to produce them. The aim of our work is to propose an end-result evaluation method based on a single scene, using a specifically designed chart. We consider the depth maps embedded in the photographs, which are not visible to the user but are used by specialized software, in association with the RGB pictures. Some of the aspects considered are spatial alignment between RGB and depth, depth consistency, and robustness to texture variations. This work also provides a comparison of perceptual and automatic evaluations.

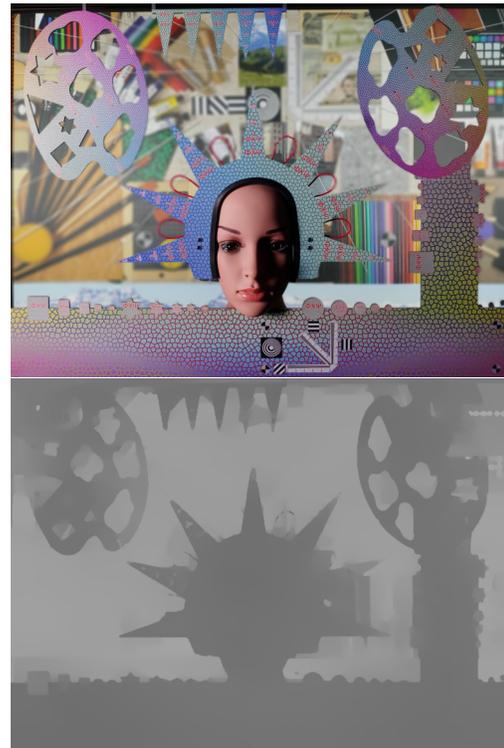
## Introduction

Computational Bokeh, and various other depth-based photographic features, have become major selling points for flagship smartphones in the last few years. These applications have different quality requirements than most depth imaging technologies. Bokeh simulation requires depth maps to be very precisely aligned with an RGB picture, and suffers largely from artifacts and segmentation errors. On the other hand, distance estimation is generally less important, and the relative depth levels of the elements of the scene is what matters. Additionally, those applications use various types of depth imaging technologies, ranging from single-image deep learning estimation to time-of-flight sensors.

In this article, we present a quality evaluation framework for depth maps as a means towards photographic applications. The different aspects of quality that we evaluate can be classified in two main categories: segmentation and depth consistency. Examples of segmentation problems are offset edges or incomplete detection of small elements. Under depth consistency, we regroup aspects such as sensitivity to texture variations, depth estimation and depth regularity.

Our approach is based on a single scene, designed to highlight those defects and make them measurable. The scene contains a 2-dimensional chart and a background. Our measurement algorithms output several metrics, mostly local. Rather than a single global evaluation metric, we propose feature-level measurements to enable manufacturers to evaluate precisely the strengths and weaknesses of their devices.

The rest of the article is organized as follows. The next section presents the chart that we designed as the main element of our test setup. The following section introduces the experimen-



**Figure 1.** Picture of the bokeh setup proposed in [7] and the associated depth map captured with a smartphone. Depth map defects are obviously visible in this example, mostly around the head and in the edges. Some background elements start to appear, and some holes of the palettes are blended in with the foreground.

tal setup and test protocol used for our analyses. Afterwards, we present the metrics we propose as well as the implemented algorithms, followed by the analysis of the results and a comparison with perceptual evaluation.

## Related Work

A number of papers on depth map quality evaluation have been published over the last few years. Several trends can be distinguished. Haddad [6] and Kim *et al.* [11] focus on depth maps for application such as Free Viewpoint Television or 3D Television, whereas Koch *et al.* [12] proposes a set of metrics to evaluate the output of CNN-based single-image depth estimation. Several papers [1, 19] propose evaluations based on well-known 2D quality metrics (PSNR, SSIM, VIF, etc.) applied to stereoscopic cases. Others, such as Benoit *et al.* [2], propose local quality metrics for disparity distortion. Finally, many papers [3, 14, 13, 18, 15] also propose neural network architectures to extract depth maps from

single images, using various quality metrics for training.

Those works have in common that they do not take advantage of a complete prior knowledge of the scene. They are meant to be usable for image-to-image comparisons with a ground truth image, which makes them easily testable against datasets such as KITTI [4], NYU v2 [17] or Middlebury [16]. The novelty of our work is to use a precisely known laboratory setup to compute local metrics, targeted at specific known defects.

## Chart design

The main part of our experimental setup is the chart. Our goal was to design the elements of the chart to mimic some real life situations, and highlight some specific defects. Therefore, its conception was very much based on observation and experimental prototyping. As starting point we used the Bokeh chart proposed in [7] shown in Fig. 1. This work was designed for perceptual evaluation, and was not fit for automated measurements because of the imperfect knowledge of the geometry of the chart. However, it enabled us to observe and classify many defects.

From those observations, we established a list of interesting elements to put in our new chart. Starting with textures, it was obvious that strong texture variations caused some devices to infer depth variations, even if there was none. This is likely caused by some kind of guided filtering [8], which causes background details to appear in the depth maps. Furthermore, highly repetitive patterns are tricky for devices that use stereoscopic vision, and low-reflectivity surfaces usually cause depth estimation issues for time-of-flight or structured light sensors.

Segmentation issues were most visible around elements that are very close together, such as fingers. Artifacts were also visible near the end of thin elements, where the resolution of the depth sensors may not be sufficient and filtering algorithms come into play. In low-contrast areas, the difference between foreground and background was sometimes not detected, highlighting the importance of the background.

Knowing that face detection is almost universally used in the process of depth estimation in smartphones, a simulated head is an essential element to put in our setup. It is likely that some devices will infer the shape of a body under the head, introducing some interesting defects. Furthermore, the face is the part of the body where human perception focuses the most and where any small defect can result in a very unrealistic portrait.

The decision was made to create a two-dimensional chart, shown in Fig. 2, because measurements can be carried out much more easily than on a three-dimensional one. It is easier to know the geometry with sufficient precision, and avoids occlusion and perspective issues. The legend for the regions of interest in Fig. 2 are given in Tab. 1. The bottom part of the chart is simply used as a support, so that it can be held without wasting any useful space. The chart was printed on DIBOND® aluminium composite material, which makes it rigid and reasonably flat.

## Experimental setup

Our experimental setup consists of the chart and a background, providing two layers of depth. As previously stated, the importance of the background is capital. The contrast and the patterns of the background have a big impact on the end result, especially parts that are connected or very close to the foreground elements. Therefore, the choice of the background to be used was

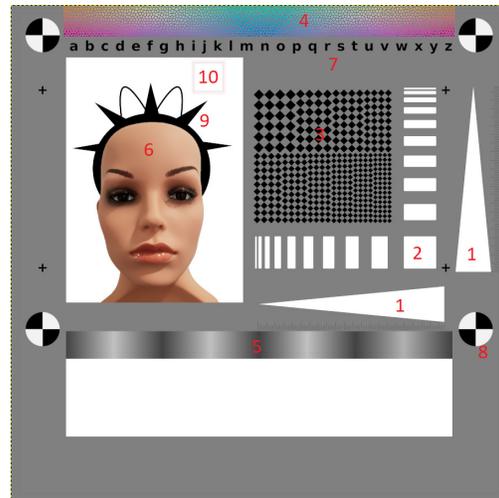


Figure 2. Annotated depth map chart.

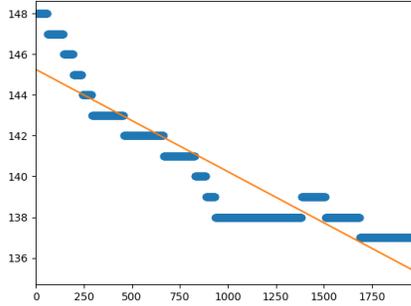
Element	Expected defects
1. Triangular holes	Filtering issues, wrong detection of thin parts
2. Gradual holes	No detection or mixed detection of thin holes
3. Checkerboard	Depth estimation issues due to repetitive pattern
4. Textured strip	Depth inconsistency due to texture variation
5. Black and white strip	Depth inconsistency due to reflectivity variation
6. Head	Depth inference caused by face detection
7. Letters	Filtering issues because of very contrasted elements
8. Markers	None, used for chart detection
9. Crown	Filtering issues, wrong detection of thin parts
10. Alignment marker	None, used to check the printer alignment

Table 1. Elements of the chart

non-trivial. Our first idea was to use a uniform white backdrop, but this was unfair to stereoscopic devices, and did not really resemble any real life situation. In the end, our choice was to use the same background that is used for the bokeh setup of [7].

This background is quite challenging because it has lots of small elements, however it is fair for every kind of technology, which is what we are mainly looking for. It is printed on a backdrop and is therefore flat. In order to ensure that results are comparable among devices, the same exact framing is used in every case. During a shooting session, both the distance from the chart to the background and the distance from the device to the chart need to be adjusted depending on the focal length of the device, using the following formula  $d_2 = \frac{f_2}{f_1} d_1$ , where  $d_2$  is the new distance between the device and the element,  $d_1$  the previous distance, and  $f_1$  and  $f_2$  are the focal lengths of each device.

We use two lighting conditions during our shooting sessions: D65 at 1000 lux, simulated by a fluorescent illuminant with a color temperature of 6500K, and A at 50 lux, a tungsten illuminant at 2856K. Those two cover the most common illuminants and offer widely different spectra. In both cases, our shooting protocol consists of five pictures taken on a tripod, forcing the autofocus



**Figure 3.** Evolution of the gray levels along the width of the chart. We can see that the levels are very noisy and it is difficult to extract a clear tendency. A linear regression is given for reference.

to run again between pictures. The chart is placed parallel to the sensor of the device or at low angles ( $\leq 10^\circ$ ).

## Metrics and algorithms

The goal of our work was to propose a number of local metrics rather than one global metric. In every case, we started from a known defect, and tried to find a metric to measure it. All of our metrics take advantage of a near-perfect knowledge of the scene to compute relevant values.

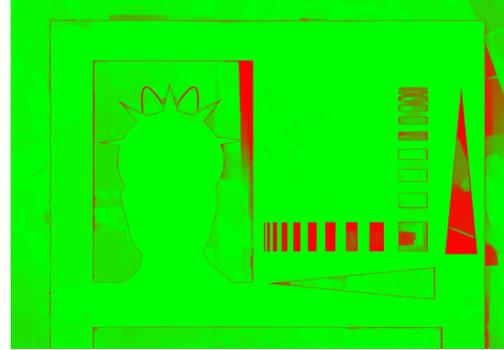
### The linearity issue

As a starting point, we need a precise estimate of the ground truth of the scene. By considering that the chart and the background are both flat, the depth becomes easy to model. However, the interpretation of the depth maps is more problematic. There is currently no standard for depth map encoding, and the formats used by the manufacturers are usually not documented. Since the absolute distance is not important in this use case, depth maps are not directly associated to real distances. The conversion from a distance to a gray level is done by an unknown function, which we assume strictly monotonic and continuous. In order to measure the accuracy, we need to project the ground truth in the depth map space, and therefore estimate this function.

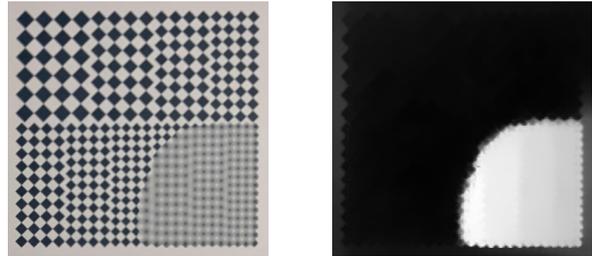
The four markers in the corners of the chart enable us to know its position in the RGB image with an error of less than two pixels, even in noisy conditions. It is therefore possible to extract the foreground pixels of the depth map and fit a parametric model to the surface. However, a trade-off needs to be found : an overly complicated model would risk over-fitting, and we would not be able to detect defects on the surface ; on the other hand, another extreme solution would be to consider that the chart is perfectly parallel to the sensor and use a single depth value for the chart, but this would leave too much room for human error during the shooting sessions.

A reasonable solution would have been to try several parametric models (e.g. linear, logarithmic, inverse...), but we observed that the variations were too small and noisy to extract any meaningful tendency since the chart was almost parallel to the sensor. As an illustration, Fig. 3 displays a plot of the gray levels along the width of the chart. Therefore, we chose a simple linear model for this fitting.

To reduce the impact of outliers, we use the robust Huber regression distance [9]. This parametric model enables us to infer



**Figure 4.** Error map displaying the error for each pixel, relative to the pixel dynamic.



**Figure 5.** Artifact due to a high frequency pattern.

an expected depth value for every foreground pixel. That way, we can compute a difference between the expected background and foreground values for every pixel, and use this difference to adapt the thresholds in our measures. We will refer to this difference as the *pixel dynamic*. When computing metrics on areas, we use the maximal pixel dynamic in the area and refer to it as *area dynamic*. Depending on the encoding, the value of the foreground pixels may be lower (darker) than the background, or the opposite : henceforth, we shall assume that the lowest depth values correspond to positions closer to the camera.

The only global metric of this work is called *error areas*, and corresponds to the proportion of the pixels where the difference between the expected value and the actual value is superior to 10% of the pixel dynamic. It is computed similarly for the background and the foreground, on a cropped area to avoid errors due to framing. The second output of this measurement is an *error map* (Fig. 4), which enables the user to see at a glance where the problematic areas are.

In addition to this metric, we introduce the *planarity error*. It is basically the same metric, applied to a specific part of the chart of Fig. 2. Those areas are the face, the checkerboard (seen in Fig. 5) and the two textured strips, enabling us to quickly see which part is most challenging for the device.

### Edge segmentation

We define an ideal edge as a sharp transition between two depth levels corresponding to the background and foreground. This transition should happen on a single pixel and be perfectly aligned with the ideal position of the edge. The first step of our process is to apply an homography on the depth map, using bilinear interpolation, to re-align the four markers of the target, which means that the edges of the chart are perfectly parallel to the edges



Figure 6. Regions of interest for the measurements on the edges (in red).

of the image. We chose to define twelve regions of interest along the edges of the chart (shown in Fig. 6), three on each side.

**Gradient-based measurements.** The first half of our metrics are gradient-based metrics, aimed at spatial characteristics of the edges. The preliminary step to these metrics is to convolve the depth map of the region of interest with a Sobel [10] filter. This will allow to precisely compute the position of the edges and the value of the edge gradient. Starting from there, we define the following three metrics. In all the following formulas,  $n$  stands for the length of the edge and  $x_{\text{edge}}$  is the position of the depth map edge along the perpendicular axis. This position should ideally be constant, because the ideal edges are aligned with the edges of the image.  $x_t$  is the ideal position of the edge, and is constant.

The *Pixel Shift* (PS) metric is the average distance between the detected edge and the ideal edge. It is expressed as  $\text{PS} = \frac{1}{n+1} \sum_{i=0}^n (x_{\text{edge}}(i) - x_t)$ .

This metric, being an average, does not tell us if the shift is global or if the average is biased by a few outliers. This is the purpose of our next metric, which consists of the *standard deviation of the edge position* ( $\sigma_{\text{edge}}$ ), defined as  $\sigma_{\text{edge}} = \sqrt{\frac{1}{n+1} \sum_{i=0}^n (x_{\text{edge}}(i) - \bar{x}_{\text{edge}})^2}$ .

Finally, our third spatial metric is the *Relative Gradient* (RG). It is a measurement of the edge gradient value, rather than its position. As explained before, the expected behavior is a sharp transition between the background value and the foreground value. The gradient at each point of the edge should then be equal to the pixel dynamic. The value computed is an average result over the region of interest, expressed as a proportion of the pixel dynamic. Fig. 7 gives a visual representation of some of the proposed metrics.

**Histogram-based measurements.** The second half of our metrics are based on the distribution of depth values in the region of interest of the edges, previously defined. They complement the first three ones with a fine analysis of the depth levels in the entire region of interest, rather than just along the main edge. Their purpose is to detect defects such as background details appearing in the depth map or depth steps effects.

The first metric we define is the *Pixel Repartition* (PR) metric. We first find the two main peaks of the histogram of depth

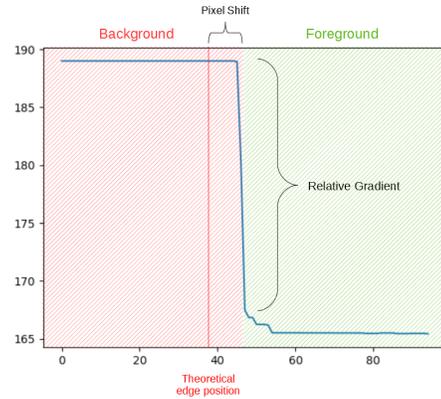


Figure 7. Graph of the gray levels along a line orthogonal to an edge of the chart, with visual representations of the relative gradient and pixel shift metrics.

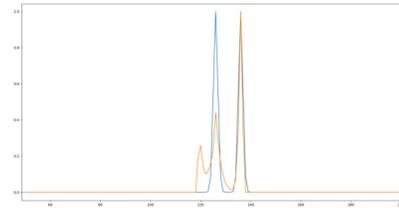


Figure 8. Histogram of a region of interest (in orange) and the Gaussian windows (in blue). We can see a small peak on the left, caused by some artifact.

values, corresponding to the background and foreground values, and then multiply the histogram with a bimodal gaussian distribution centered on the peaks. The PR metric is the sum of this weighted histogram, divided by the total number of pixels in the area. The gaussian windows have a height of 1, which means that the metric cannot exceed 1, and their standard deviations are normalized proportionally to the dynamic. A visual representation is given in Fig. 8.

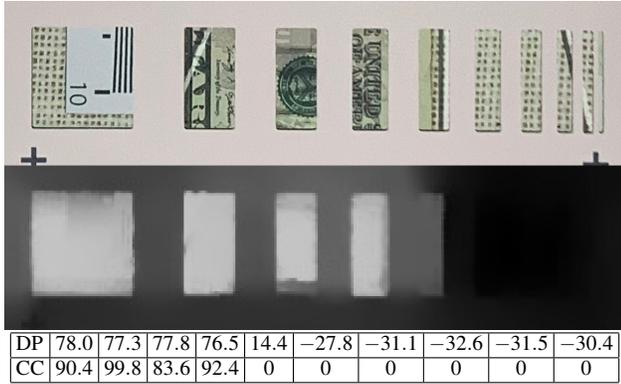
The use of a Gaussian window makes it so that only a perfect device can obtain a perfect PR of 1, but devices that show slight depth variations will not be too penalized. However, the larger the variations, the lower the PR will be.

This metric works well as a single indicative value, however it does not show which part of the depth map is most problematic. As a complement, we also compute the *background* and *foreground standard deviation* in the region of interest, using the average of the two main peaks as the separation threshold.

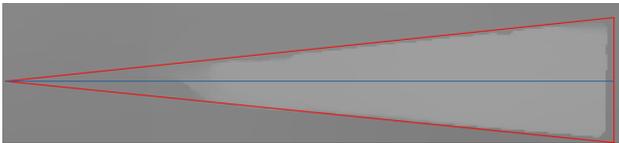
The last metric is the base-2 *entropy of the depth histogram*, defined as  $H = -\sum_k p_k \cdot \log_2(p_k)$ , where  $p_k$  is the probability corresponding to the  $k$ -th bin of the normalized histogram. This has the advantage of showing whether all pixels have the same or different depth values, thus enabling the user to differentiate a step effect and a gradient effect.

### Holes detection and dynamic

The main metric regarding the holes of the chart is the *Dynamic Proportion* (DP). Since the background of our experimental setup is flat, the expected behavior is that the value in every



**Figure 9.** Visualisation of the gradual holes of the chart, with visible depth estimation defects. The values for Dynamic Proportion (DP) and Contour Coverage (CC) are given in the table.



**Figure 10.** Example of incomplete triangular hole, with the ideal shape and its bisector.

hole should be equal to the background value around the chart. However, due to filtering issues and imprecision of depth sensors, this is often not the case in real depth map images. This metric is computed as follows on a given region:  $DP = \max_{i,j} \frac{D_{i,j} - \hat{D}_{fg}}{\hat{D}_{bg} - \hat{D}_{fg}}$ , with  $D_{i,j}$  the value of the depth map at coordinates  $i, j$ ,  $\hat{D}_{fg}$  the maximum estimated depth of the chart using the parametric model defined earlier, and  $\hat{D}_{bg}$  the depth of the background. Using the parametric model makes the metric robust to defects that could appear inside or around the hole. In some cases, holes are detected closer rather than farther by devices, in which case the metric may be negative.

For gradual holes (see Tab. 1), we implement the *Contour Coverage* (CC), which measures the detected area of the hole as a proportion of the ideal area. Depending on the size of the hole and the background elements behind it, it is frequent that devices only detect some part of the holes. The threshold used is the average between the expected background value and foreground values. The metric is the proportion of pixels in the hole area whose value is greater than the threshold. An example is given in Fig. 9.

For the two triangular holes, this metric would not be as interesting because their area is much larger. The most interesting aspect of those is that their tip is often wrongly filtered out, as shown in the example in Fig. 10. Hence, we define the *Visible Height* metric, which is computed as the proportion of visible pixels along the bisector of a triangle.

Tab. 2 summarizes the proposed metrics along with the targeted defects. All of the metrics are repeatable, with no stochastic aspect. They are also invariant to the resolution of the depth maps. The metrics that are based on the dynamic are normalized, making them invariant to the encoding of the depth.

Metric	Targeted defect
Error Areas	Global errors
Planarity Error	Depth estimation inconsistencies, on surfaces, artifacts due to high frequency patterns or texture variations
Pixel shift (PS)	Misalignment of depth map and image, wrong detection of edges
Edge standard deviation	Alteration of the shape of contours, due to background elements or filtering
Relative gradient (RG)	Wrong depth estimation around edges
Pixel repartition (PR)	Stepping effect around edges, generally due to background elements
Background / Foreground standard deviation	Inconsistencies of depth estimation around discontinuities
Entropy	Gradient effect around edges, sometimes added by devices to blur potential defects
Dynamic proportion (DP)	Wrong depth estimation in small discontinuities
Contour coverage (CC)	Incomplete or no detection of small discontinuities
Visible height	Filtering issues and hardware limitations

**Table 2. Implemented metrics and targeted defects.**

## Validation and results

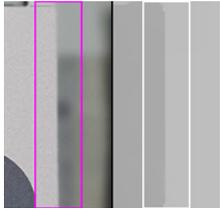
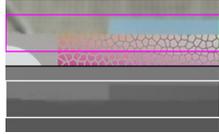
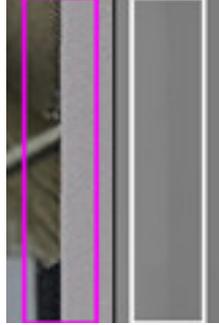
Computing the correlation coefficient between our metrics and a perceptual evaluation is difficult because our metrics are local and focus on very specific features of the depth maps. Our work provides building blocks, upon which aggregated metrics could be designed to match human perception. In its current state, this is more intended to be a tool for the tuning devices rather than a perceptual quality evaluation. Nevertheless, the obtained results are very promising. The strengths and weaknesses of the different technologies are easily visible, as in the example of Fig. 5, which was created by a stereoscopic device. Many of the artifacts caused by filtering algorithms are also easily measurable. Examples are given in Tab. 3.

Our measures also work when the chart is not perfectly parallel to the sensor. Our tests included pictures with a target tilted up to  $10^\circ$  around the vertical axis. On some devices, we observed a smoothing of depth levels on tilted surfaces, revealing a heavy filtering. We also see different segmentation issues and artifacts.

The only metric in our set which was easily correlated with perceptual measurements was the error areas, because it is global. We used the aggregated results of the perceptual analysis performed on the Bokeh chart as reference. Using seven uniformly distributed devices in our database, we find a 78% correlation. The monotonicity is overall preserved, and we can expect a top device to have a low value for the *error areas*. However, our metric is more sensitive to small depth variations than perceptual measurements, which explains some outliers. Moreover, the perceptual evaluation takes into account the Bokeh simulation output, which does not entirely rely on the depth map.

## Conclusion

We have presented a full measurement protocol and objective metrics to evaluate the quality of depth maps for photographic applications. Our method is based on a single scene, containing a specifically designed chart and a backdrop. It is designed to be challenging for all commonly used technologies and aims to reproduce and measure some common defects. Eleven metrics

Region of interest	Results	Remarks
	$\sigma_{\text{edge}} = 1.72$ px RG = 82.1 % PR = 85.0 %	Sharp transition. The $\sigma_{\text{edge}}$ is slightly higher because of shifts at the top and bottom ends of the edge.
	$\sigma_{\text{edge}} = 0.52$ px RG = 51.1 % PR = 87.2 % H = 4.33 bits	Linearity is very good, while RG is lower because the transition is unsharp.
	$\sigma_{\text{edge}} = 20.84$ px RG = 36.0 % PS = 18.43 px	Obvious stepping effect, causing a high $\sigma_{\text{edge}}$ and PS, and a low RG.
	$\sigma_{\text{edge}} = 3.45$ px RG = 19.8 % PS = 2.45	The depth transition is barely visible, causing a very low RG. However, linearity and alignment are fairly good.
	$\sigma_{\text{edge}} = 3.12$ px RG = 78.4 % PS = 1.66 px PR = 56.9 %	The defect here is poorly detected because the maximal gradient is linear even though the depth level varies. This is an edge case.

**Table 3. Examples of results obtained for multiple regions of interest of the edges. Only the most relevant metrics are displayed. The regions of interest considered are the areas enclosed in a white rectangle, the corresponding areas in the reference image are enclosed in magenta.**

are used (mostly local) to characterize precisely defined regions of interest. Those regions of interest are located on the edges of the chart, on flat surfaces or around holes. In total, 148 values are computed for each image, providing a very fine-grained analysis.

Considering that we work on the depth maps, which are not visible to the user, our work is oriented to provide a tool for tuning of devices and for selecting hardware and software components. In the case of computational bokeh, the background blur simulation rely on different parameters that affect the quality of the bokeh independently of the depth map. This explains the difficulty to correlate our metrics with perceptual evaluation. However, with respect to the depth maps, the results are relevant.

The proposed metrics are designed to be easily understandable, and could be used as building blocks in future works to extract higher level metrics. In the future, the possible adoption of an uniformized depth map format [5] would facilitate its comparison across devices.

## References

- [1] A. Banitalebi-Dehkordi et al. A study on the relationship between depth map quality and the overall 3d video quality of experience. In *IEEE 3DTV-CON*, 2013.
- [2] A. Benoit et al. Quality assessment of stereoscopic images. *EURASIP JIVP*, 2009.
- [3] D. Eigen et al. Depth map prediction from a single image using a multi-scale deep network. In *Advances in NIPS*, 2014.
- [4] A. Geiger et al. Vision meets robotics: The kitti dataset. *Int. J. of Robotics Research*, 2013.
- [5] Google Developers. Encoding depth and confidence. <https://developers.google.com/depthmap-metadata/encoding>. [Accessed 19-August-2019].
- [6] N. Haddad. *Non-reference depth map quality evaluation in immersive video applications*. PhD thesis, University of Surrey, 2016.
- [7] W. Hauser et al. Image quality benchmark of computational bokeh. *Electronic Imaging*, 2018.
- [8] A. Hosni et al. Fast cost-volume filtering for visual correspondence and beyond. *IEEE TPAMI*, 2012.
- [9] P. J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 1964.
- [10] N. o. Kanopoulos. Design of an image edge detection filter using the sobel operator. *IEEE JSSC*, 1988.
- [11] D. Kim et al. Depth map quality metric for three-dimensional video. In *Stereoscopic Displays and Applications*, 2009.
- [12] T. Koch et al. Evaluation of cnn-based single-image depth estimation methods. In *ECCV*, 2018.
- [13] B. Li et al. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *IEEE CVPR*, 2015.
- [14] F. Liu et al. Deep convolutional neural fields for depth estimation from a single image. In *IEEE CVPR*, 2015.
- [15] A. Roy and S. Todorovic. Monocular depth estimation using neural regression forest. In *IEEE CVPR*, 2016.
- [16] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. of Computer Vision*, 2002.
- [17] N. Silberman et al. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.
- [18] P. Wang et al. Towards unified depth and semantic prediction from a single image. In *IEEE CVPR*, 2015.
- [19] J. You et al. Perceptual quality assessment for stereoscopic images based on 2d image quality metrics and disparity analysis. In *Int. Workshop Video Process. Quality Metrics Consum. Electron*, 2010.

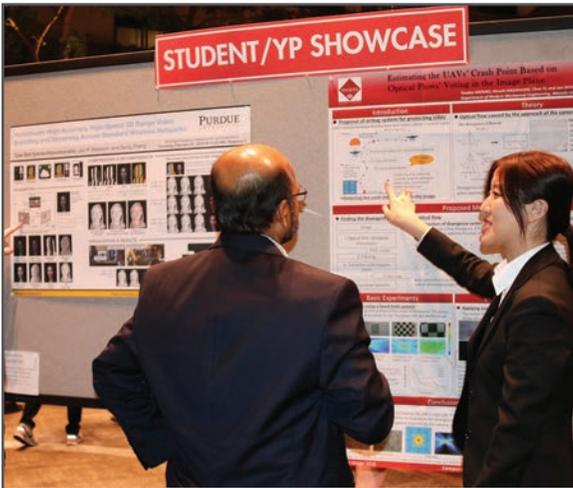
**JOIN US AT THE NEXT EI!**

IS&T International Symposium on

# Electronic Imaging

SCIENCE AND TECHNOLOGY

*Imaging across applications . . . Where industry and academia meet!*



- **SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •**
- **INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •**

[www.electronicimaging.org](http://www.electronicimaging.org)

