Dataset and metrics for predicting local visible differences

KRZYSZTOF WOLSKI, MPI Informatik
DANIELE GIUNCHI, University College London
NANYANG YE, University of Cambridge
PIOTR DIDYK, Saarland University, MMCI, MPI Informatik, Università della Svizzera italiana
KAROL MYSZKOWSKI, MPI Informatik
RADOSŁAW MANTIUK, West Pomeranian University of Technology, Szczecin
HANS-PETER SEIDEL, MPI Informatik
ANTHONY STEED, University College London
RAFAŁ K. MANTIUK, University of Cambridge

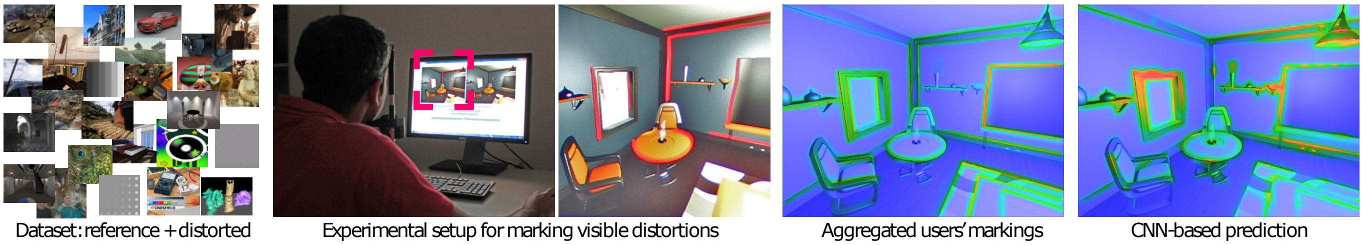

Dataset: reference + distorted
Experimental setup for marking visible distortions
Aggregated users' markings
CNN-based prediction

Fig. 1. We collect an extensive dataset of reference and distorted images together with user markings that indicate which local and possibly non-homogeneous distortions are visible. The dataset lets us train existing visibility metrics and develop a new one based on a custom CNN architecture. We demonstrate that the metric performance can be improved significantly when such localized training data is used.

A large number of imaging and computer graphics applications require localized information on the visibility of image distortions. Existing image quality metrics are not suitable for this task as they provide a single quality value per image. Existing visibility metrics produce visual difference maps, and are specifically designed for detecting just noticeable distortions but their predictions are often inaccurate. In this work, we argue that the key reason for this problem is the lack of large image collections with a good coverage of possible distortions that occur in different applications. To address the problem, we collect an extensive dataset of reference and distorted image pairs together with user markings indicating whether distortions are visible or not. We propose a statistical model that is designed for the meaningful interpretation of such data, which is affected by visual search and imprecision of manual marking. We use our dataset for training existing metrics and we demonstrate that their performance significantly improves. We show that our dataset with the proposed statistical model can be used to train a new CNN-based metric, which outperforms the existing solutions. We demonstrate the utility of such a metric in visually lossless JPEG compression, super-resolution and watermarking.

CCS Concepts: • **Computing methodologies** → **Perception**; *Image manipulation*; *Image processing*;

Additional Key Words and Phrases: visual perception, visual difference predictor, visual metric, distortion visibility, image quality, data-driven metric, dataset, convolutional neural network

ACM Reference format:
Krzysztof Wolski, Daniele Giunchi, Nanyang Ye, Piotr Didyk, Karol Myszkowski, Radosław Mantiuk, Hans-Peter Seidel, Anthony Steed, and Rafał K. Mantiuk. 2017. Dataset and metrics for predicting local visible differences. *ACM Trans. Graph.* 36, 4, Article 1 (July 2017), 14 pages.
DOI: http://dx.doi.org/10.1145/8888888.7777777

1 INTRODUCTION

A large number of applications in graphics and imaging can benefit from knowing whether introduced changes in images are visible to the human eye or not. Existing visibility metrics provide such predictions [Alakuijala et al. 2017; Mantiuk et al. 2011], but achieve only moderate success. They work well for simple stimuli, but their performance is worse for complex images. They can predict low-level visual phenomena, such as luminance and contrast masking, but they do not account for higher-level effects due to image content. We show that such higher-level effects have a significant role in detecting visible distortions.

To create a robust predictor of visible distortions, we collect a large dataset with locally marked distortions. The dataset includes 557 image pairs, 296 of which were manually marked by 15 to 20 observers and the remaining 261 pairs that were generated from existing TID2013 datasets. In comparison, the next largest dataset contains just 37 marked images [Čadík et al. 2012]. Moreover, our

0730-0301/2017/7-ART1 $15.00
DOI: http://dx.doi.org/10.1145/8888888.7777777

dataset contains more extensive variations in both type and magnitude of image artifacts. We use our dataset to calibrate popular visible difference metrics and show improvements in their performance. Then, we use the dataset to train a novel metric based on a CNN architecture, which improves the prediction accuracy over existing metrics by a substantial factor.

We demonstrate that the new CNN metric can not only predict subjective data, but enables many relevant applications. In the first example, we show that the CNN metric can reduce the size of JPEG images by about 60%, while achieving visually lossless compression. In the second application example we demonstrate that the metric can determine maximum downsampling factor for which a single-image super-resolution algorithm can reconstruct a visually equivalent image. Finally, in the third application we show how the CNN-based metric can be used to introduce to an image an invisible watermark.

The main contributions of the paper are as follows:

- The largest publicly available dataset[1] of manually marked and generated visible distortions (Section 3).
- A statistical inference model allowing robust fits to noisy subjective data (Section 4).
- Retraining of a number of popular metrics, which has significantly improved their predictions (Section 5).
- A CNN-based visibility metric, which outperforms all existing metrics in cross-validation (Section 6).
- The utility of the visibility metrics is demonstrated in three practical applications: visually lossless JPEG compression, superresolution, and watermarking (Section 8).

The supplementary materials, code and the dataset can be found at http://visibility-metrics.mpi-inf.mpg.de/.

2 RELATED WORK

Image metrics can be divided into *quality* and *visibility* metrics, both addressing different applications. Image quality metrics (IQMs) predict a single global quality score for the entire image. These metrics usually are trained and evaluated on mean opinion scores (MOS) [Ponomarenko et al. 2015; Sheikh et al. 2006] that are obtained in user experiments for each distorted image. In contrast, visibility metrics [Aydin et al. 2008; Daly 1992; Mantiuk et al. 2011] predict the probability that a human observer will detect differences between a pair of images. They provide localized information in the form of visibility maps, in which each value represents a probability of detection. Visibility metrics tend to be more accurate for small and barely noticeable distortions but are unable to assess the severity of distortion. Visibility metrics are often more relevant for graphics applications whose goal is to maximize performance without introducing any visible artifacts.

This work focuses on visibility metrics which are suitable for computer graphics applications. In this section, we provide an overview of previous quality and visibility metrics with a focus on the latter ones. Since we use machine learning techniques to derive our metric, we also discuss relevant literature in this area.

2.1 Quality metrics

The vast majority of IQMs are *full reference* (FR) techniques that take as input both reference and distorted images and compute local differences, which are then pooled across the entire image into a single, global quality score. The simplest approach to compute such local differences is pixel-wise absolute difference (ABS), or the Euclidean distance (ΔE) between RGB components. The latter is employed in the popular Root Mean Square Error (RMSE) and Peak Signal-to-Noise Ratio (PSNR) metrics. A better approximation to the perceived differences is achieved when pixel RGB values are converted to a perceptually uniform color space and a color difference formula, such as CIE ΔE 2000 (CIEDE2000), is used.

More advanced quality measures such as the Structural Similarity Index Metric (SSIM) [Wang and Bovik 2006, Ch. 3.2] account for spatial information by computing differences in the local mean intensity and contrast, as well as cross-correlation between pixel values. The Visual Saliency-Induced Index (VSI) employs a similar framework, but contribute to the local difference map four components: the visual saliency, the gradient magnitude, and two chrominance channels [Zhang et al. 2014]. The Feature Similarity Index (FSIM) also employs the gradient magnitude, this time complemented by the phase congruency, to derive the local difference map [Zhang et al. 2011]. VSI employs a saliency map, and FSIM the phase congruency map, as a weighting function used for pooling the final score. While VSI and FSIM produce local difference maps at intermediate processing stages, their utility as local visibility predictors has not been tested. Although global IQMs, such as SSIM, VSI, and FSIM were not intended to predict visibility, in this paper we demonstrate that they can be trained as such.

For a complete overview of IQMs we refer the reader to numerous surveys [Chandler 2013; Lin and Kuo 2011], which also provide information on over 20 image quality databases with MOS data, including the popular LIVE [Sheikh et al. 2006] and TID2013 [Ponomarenko et al. 2015] datasets. The distortion types covered in those datasets correspond to most prominent applications of IQMs and include various image compression and transmission artifacts, as well as different types of noise, blur, and ringing.

2.2 Visibility metrics

Visibility metrics address a challenging problem of predicting the visibility of distortions for each pixel location. Since it is more difficult to collect sufficient data for training such visibility metrics, they often rely on the low-level models of the visual system. The models help to constrain a space of possible solutions and reduce the number of parameters that need to be trained.

Early works on visibility predictors often focused on modeling spatial contrast sensitivity. For example, the sCIELab [Zhang and Wandell 1997] metric prefiltered CIELab encoded pixels with a spatio-chromatic contrast sensitivity function prior to computing the visibility map. This simple approach has limited success in predicting distortions in complex images, as it ignores important aspects of supra-threshold perception, such as contrast constancy or contrast masking. Those issues are addressed by a family of more complex metrics, inspired by the models of low-level vision. Some examples of those are the Visual Discrimination Model

[1] The dataset is available at: https://doi.org/10.17863/CAM.21484

(VDM) [Lubin 1995], the Visible Differences Predictor (VDP) [Daly 1993], and HDR-VDP [Mantiuk et al. 2011]. Those metrics consider luminance adaptation, contrast sensitivity, contrast masking, and frequency-selective visual channels [Chandler 2013]. However, their predictions are less accurate for complex images [Čadík et al. 2012].

A different approach was taken by metrics based on feature maps, such as Butteraugli, which is the core part of Google's perceptually guided JPEG encoder project "Guetzli" [Alakuijala et al. 2017]. Butteraugli transforms an input image pair into a set of feature maps, such as an edge detection map, and a low-frequency map, which are then combined to form the final difference map prediction. The maximum value in the difference map is taken as the score of JPEG compression quality. As the metric was intended for JPEG artifacts, it is not clear how this metric performs for computer graphics applications that are notoriously difficult for other visibility metrics [Čadík et al. 2012].

Here we support the observation that the key problem that hinders the development of better quality metrics is limited training data, which must be locally annotated and represent a great variety of distortions with sub-threshold, near-threshold, and supra-threshold magnitudes [Chandler 2013]. In an attempt to fill this gap, Alam et al. [2014] measured local discrimination thresholds for over 3000 patches but in only 30 images. The measurement of discrimination thresholds is a very tedious task, which is not well suited for collecting large datasets. Therefore, in this work we use a more efficient experimental procedure of marking visible distortions [Čadík et al. 2013; Čadík et al. 2012; Herzog et al. 2012].

We extend the range of collected data not only by considering a much larger set of images and distortions, but also by measuring different levels of these distortions, which we found to be essential for successful training. We propose an efficient experimental setup to reduce the within-subject variance in such systematic distortion marking, which further improves the consistency of our training data. We capitalize on the availability of such massive training data to improve the performance of existing visibility metrics such as sCIELab, HDR-VDP, and Butteraugli. Also, we attempt to further improve the accuracy of local visibility predictions with a new CNN-based metric. We expect that through extensive training, relevant characteristics of the human visual system (HVS) can be learned, including higher-level effects related to the image content, its complexity, and distortion type.

2.3 CNN-based quality metrics

Although the utility of CNN-based methods has not been demonstrated in the context of visibility metrics so far, several solutions for IQMs have been proposed. We discuss here selected IQMs that show some similarities to our method regarding the architecture or training methodology. Since in all cases huge volumes of training data are required, those metrics are trained on small patches extracted from image pairs [Bianco et al. 2016; Bosse et al. 2016a,b; Kang et al. 2014]. The key problem is that all patches derived from an image share the same MOS value. Although this seems to be a reasonable assumption for homogeneous distortions, such as compression or noise, even in such cases the artifact visibility varies across the image due to complex HVS effects such as contrast masking [Chandler 2013; Mantiuk et al. 2011]. Furthermore, when dealing with spatially-varying distortions, which are common for computer graphics applications, using a single value per image is not an option. In our training we avoid this problem by using dense visibility maps from the marking experiment.

Recently, it has been demonstrated that machine learning techniques can significantly improve the performance of IQMs. While early approaches used predefined features such as SIFT and HOG [Moorthy and Bovik 2010; Narwaria and Lin 2010; Saad et al. 2012; Tang et al. 2011], the best performance has been reported when learning techniques are applied to both feature design and regression at the same time. In such a case, one can train a metric without any domain-specific knowledge. The dominant trend here is to derive *no-reference* (NR) IQMs that do not require any reference image as an input. Inspired by previous observations that low-level features should be able to capture natural scene statistics [Wang and Bovik 2006], Kang et al. [2014] designed a shallow CNN architecture consisting of five layers and including only one convolutional layer with 50 filter kernels. Although the complexity of the architecture is similar to the traditional quality metrics, the key difference is that in the case of CNN-based metrics, the kernels are learned. Despite the good performance of shallow architectures, it has been demonstrated that a much deeper 12-layer CNN can achieve further improvements [Amirshahi et al. 2016; Bosse et al. 2018]. This suggests that higher level features encoding locally invariant information important for object recognition may contribute to the image quality evaluation. A similar architecture was applied in the context of FR-IQMs [Bosse et al. 2018]. The key idea was to duplicate the previous architecture for both the reference and distorted images and then submit the resulting feature vectors to a 2-layer fully connected network to learn a regression function that estimates the aggregated MOS rating. To address the spatial variance of relative image quality, a second branch is integrated into the regression module to learn the importance of patches, i.e., their relative weight in the aggregated MOS rating. This is an attempt to correct for the assumption of the same MOS rating for all patches in a distorted image, which was made for the training data.

In this work, we propose a CNN-based visibility metric that greatly benefits from our adequate training data, and features a novel CNN architecture that significantly departs from architectures considered so far in image quality evaluation. In our loss function, we explicitly account for the stochastic nature of training data due to differences between subjects as well as the combined effect of distortion search and actual detection in complex images.

3 DATASET OF VISIBLE DISTORTIONS

The aim of the experiment is to collect data on distortion visibility in each image location. This serves to distinguish between distortions that are below the visibility threshold and cannot be detected, and those that are well visible. Such visibility thresholds are typically collected in threshold experiments, using constant stimuli, adjustment or adaptive methods, which can measure a single image location at a time, making such procedures highly inefficient. For example, the largest dataset collected using such methods [Alam et al. 2014] contains just 30 images, 216×216 pixels each, and it required tens of experiment hours to collect it. Instead, we refine the procedure from

Subset name	**Scenes**	**Images**	**Distortion levels**	**Level generation method**	**Res. [px]**	**Source**
MIXED	20	59	2-3	blending	800×600	custom software, photographs [Čadík et al. 2012]
PERCEPTIONPATTERNS	12	34	1,3	blending	800×800	MATLAB [Čadík et al. 2013]
ALIASING	14	22	1-3	varying sample number	800×600	Unity, CryEngine [Piórkowski et al. 2017]
PETERPANNING	10	10	1	n/a	800×600	Unity, CryEngine [Piórkowski et al. 2017]
SHADOWACNE	9	9	1	n/a	800×600	Unity, CryEngine [Piórkowski et al. 2017]
DOWNSAMPLING	9	27	3	varying shadow map resolution	800×600	Custom OpenGL app [Piórkowski et al. 2017]
ZFIGHTING	10	10	1	n/a	800×600	Unity, CryEngine [Piórkowski et al. 2017]
COMPRESSION	25	71	2-3	varying bit-rates	512×512	crops from photographs
DEGHOSTING	12	12	1	n/a	900×900	photographs [Karađuzović-Hadžiabdić et al. 2017]
IBR	18	36	1,3	varying distance between key frames	960×720	custom software [Adhikarla et al. 2017]
CGIBR	6	6	1	n/a	960×720	custom software [Adhikarla et al. 2017]
TID2013	25	261	n/a	n/a	512×384	Kodak image dataset [Ponomarenko et al. 2015]

Table 1. Dataset details.

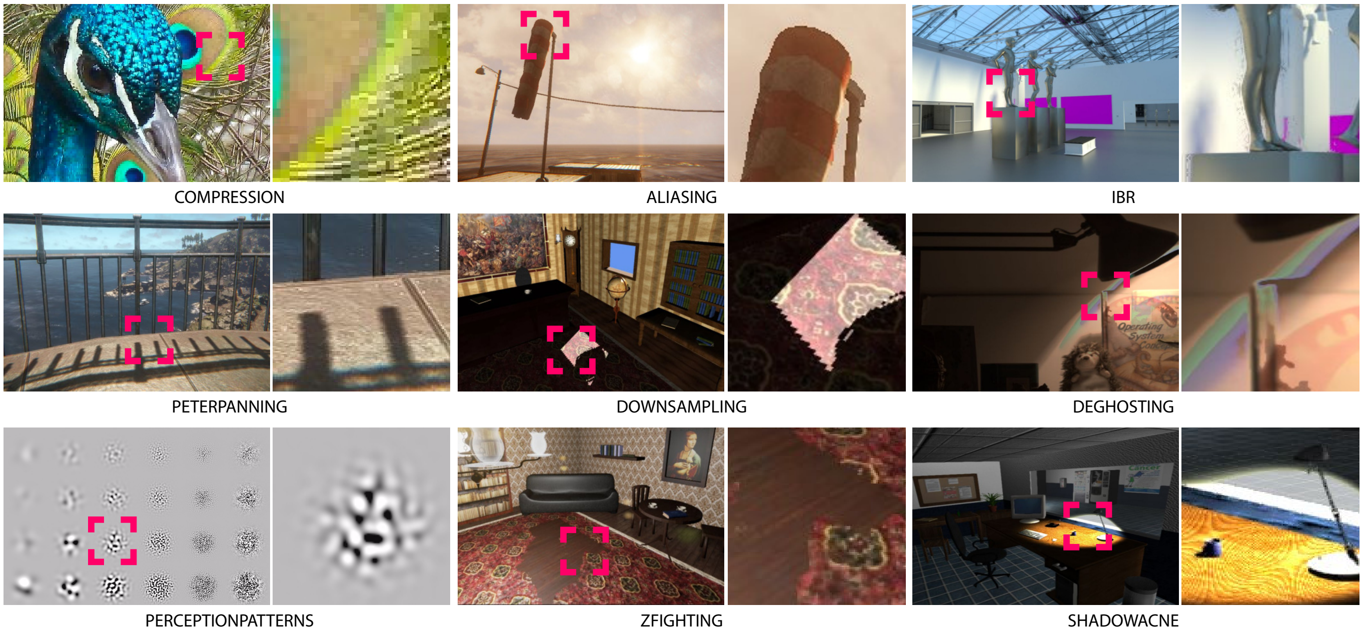


Fig. 2. The figure presents examples of stimuli from our dataset. The insets show the closeup of the artifacts. For the full preview of the image collection please refer to the supplemental materials.

[Čadík et al. 2012] to obtain the largest dataset of local visible distortions. In addition, we also included images from the TID2013 quality datasets with automatically generated markings, as described in the supplemental material.

3.1 Stimuli

The dataset consists of 557 images with 170 unique scenes. Many of them are generated for up to 3 distortion levels, for example, different quality settings of image compression. The scenes were selected to cover many common and specialized computer graphic artifacts such as noise, image compression, shadow acne, peter-panning, warping artifacts from image-based rendering methods and deghosting due to HDR merging. This variety makes our data challenging for existing visibility metrics.

The images used in our dataset come from many previous studies. We organize them into the following subsets. MIXED (59 images) is an extended LOCCG dataset from [Čadík et al. 2012] where we generate images at several distortion levels by blending or extrapolating difference between the distorted and the reference images. The distortions include high-frequency and structured noise, virtual point light (VPL) clamping, light leaking artifacts, local changes of brightness, aliasing and tone mapping artifacts. PERCEPTIONPATTERNS (34 images) from [Čadík et al. 2013] are artificial patterns designed to expose well known perceptual phenomena, such as luminance masking, contrast masking and contrast sensitivity. Datasets ALIASING (22 images), PETERPANNING (10 images), SHADOWACNE (9 images), DOWNSAMPLING (27 images) and ZFIGHTING (10 images) are derived from [Piórkowski et al. 2017] and contain real-time rendering artifacts. Those images were created using popular game engines (i.e. Unreal Engine 4, Unity) and they contain both near-threshold (e.g. aliasing) and supra-threshold distortions (e.g. z-fighting, peter-panning). COMPRESSION (71 images) contains distortions due to experimental low-complexity image compression, operating at several bit-rates. This set is an important source of near-threshold distortions. DEGHOSTING (12 images) contains artifacts due to HDR merging, which exposes shortcomings of popular deghosting methods [Karađuzović-Hadžiabdić et al. 2017]. IBR (36 images), and CGIBR (6 images) contain artifacts produced by view-interpolation and image-based rendering methods, which come from [Adhikarla et al. 2017]. TID2013 (261 images) contains a subset of images from TID2013 image quality dataset [Ponomarenko et al. 2015] in which images were selected so that the distortions are visible in the entire image (the entire marking map set to 1), or are invisible (the entire marking map set to 0). More details about all dataset categories may be found in the supplemental material. A terse summary of our dataset is provided in Table 1 and examples of selected images are shown in Figure 2.

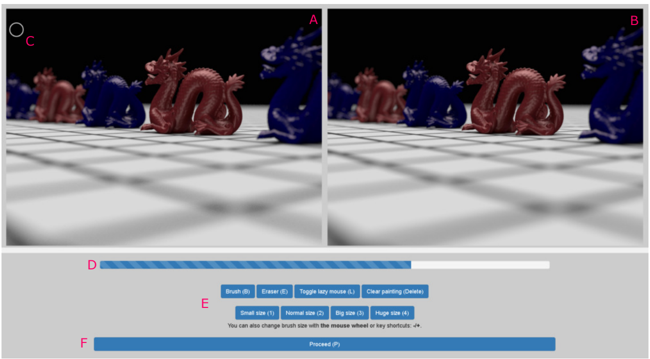
3.2 Experimental procedure and apparatus

In this section we present our experimental procedure for marking visible distortions.

Comparison method. The visibility of image differences can be measured using different presentation methods, such as flickering between distorted and reference images, the same with a short blank screen in between, a side-by-side presentation, or no-reference presentation [Čadík et al. 2012]. Different presentation methods will result in different sensitivity to distortions. Observers are extremely sensitive to differences in flicker presentation, resulting in overly conservative estimates of visible differences for most applications, in which a reference image is rarely presented or available. For that reason we opted for side-by-side presentation, which is also more relevant for many graphics applications.

Experiment software. For the purpose of collecting training data, we designed a web application for marking visible distortions. To increase comfort and accuracy of marking, we provided ability to change brush size, erase, clear all marking, and a "lazy mouse" modifier. The "lazy mouse" function makes the brush follow slowly the cursor and let the observer to paint smooth strokes without using advanced devices while significantly increasing marking precision. Figure 3 depicts the application layout.

Multiple levels of distortion magnitude. To increase the efficiency of collecting data and consistency, the images were presented with gradually increasing distortion magnitude. Up to three distortion levels were generated, depending on the distortion type. The marking map was copied from the previous distortion level so that only newly visible distortions had to be marked. Moving back to the previous distortion level was not allowed. Figure 4 shows a sample scene with three distortion levels and the corresponding observer markings.



You can also change brush size with **the mouse wheel** or key shortcuts: -/+.

Fig. 3. Layout of the custom application for marking visible distortions: A) distorted and B) reference images, C) brush cursor, D) progress bar, E) setting buttons, and F) proceed button.

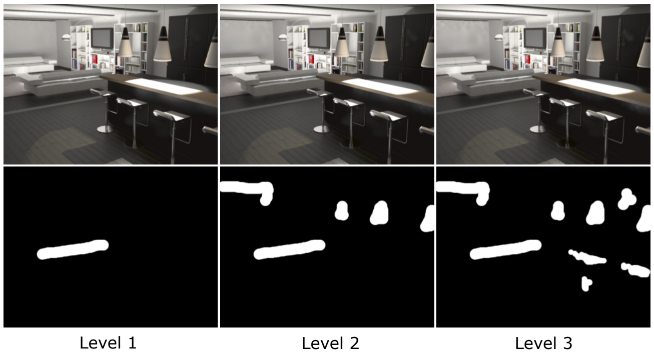


Fig. 4. An example scene with three levels of distortion magnitude (top row), and the corresponding distortion markings (bottom row). The distortion level increases from left to right, which results in adding newly marked regions.

Viewing conditions. The experimental room had dimmed lights, and the monitor was positioned to minimize screen reflections. The observers sat 60 cm from a 23″, 1920×1200 resolution Acer GD235HZ display, resulting in the angular resolution of 40 pixels per visual degree. The measured peak luminance of the display was 110 cd/m^2 and the black level was 0.35 cd/m^2. For COMPRESSION and DEGHOSTING sets, the distance was changed to the one corresponding to 60 pixels per visual degree to reduce the visibility of distortions.

Observers. Different groups of observers were asked to complete each subset of the dataset. At least 15 and at most 20 observers completed each subset. In total, 46 observers (age 23 to 29) were recruited from among computer science students and researchers. The observers were paid for their participation. All observers had normal or corrected-to-normal vision and were also naive to the

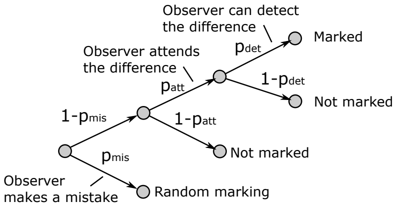


Fig. 5. The statistical process modeling observed data, given the probability of observer making a mistake (p_{mis}), the probability of attending (p_{att}) and detecting (p_{det}) differences in images.

purpose of the experiment. To reduce the effect of fatigue, the experiment was split into several sessions, where each session lasted less than one hour. The post-experiment interviews indicated that the session length was acceptable and did not cause excessive fatigue.

4 MODELING EXPERIMENTAL DATA

When fitting a visibility prediction model to the collected data, it is important to note that the data is the result of a stochastic process that is affected by noise and cannot be considered as the ground truth. The visibility threshold is not constant as the detection can vary substantially between observers or even for the same observer when the measurements are repeated. Moreover, we collect binary data (visible or not) in a marking experiment, which is affected by more factors than the detection performance. If we had tried to fit the metrics directly to the data, we would have trained them to predict the data with all noise and irrelevant effects collected in the experiment. Therefore, we found it necessary to model the stochastic process and introduce such a model into the loss function used for calibrating the metrics.

But let us first consider what kind of process we want to model. Most of the existing visibility metrics, including VDP and HDR-VDP, attempt to predict a detection threshold for an average observer. This is done by measuring detection thresholds for each observer and then predicting the average of that data. The problem with this approach is that the visual performance of most observers is different from that of an average observer. For many applications, it is arguably more important to predict the performance of a population of observers rather than an average observer. Therefore, our goal is to predict the proportion of the entire population that is going to perceive a difference. For that reason, we involve more observers (15–20) than typically found in discrimination experiments (2–5), but we expect less precise measurements for each observer.

Let us model the likelihood of observing collected marking data given that we know the true probability of detection (p_{det}). We will later use such likelihood as a loss function. Firstly, we need to account for observers making a mistake and marking or not marking pixels regardless whether they see a difference or not. We denote the probability of a mistake as p_{mis} and assume that markings are due to a mistake in 1% of the cases ($p_{mis} = 0.01$), which are mostly caused by inaccurate marking of a selected region. Such a probability of a mistake is a common element of many statistical models of psychophysical procedures, which prevent strongly penalizing the outcome of on experiment if an observer did not act as expected. Secondly, we observed that many well visible distortions were not marked because the observers were not attending a particular part of an image. Finding localized differences, in particular for computer graphics distortions, is a challenging task and not every detectable difference is going to be attended and spotted every time. Therefore, for a pixel to be marked, an observer needs to both attend a particular image location (the observer must look at the spot) and must be able to detect the difference (the difference must be visible). Given that the probability of attending an image location is p_{att} and the probability of detecting a difference is p_{det}, the entire process of marking for a single observer can be modeled statistically as shown in Figure 5. If we have multiple observers marking an image location, the probability of observing an outcome of an experiment in which k out of N observers mark a patch is described by the Bernoulli process with an adjustment for the mistakes:

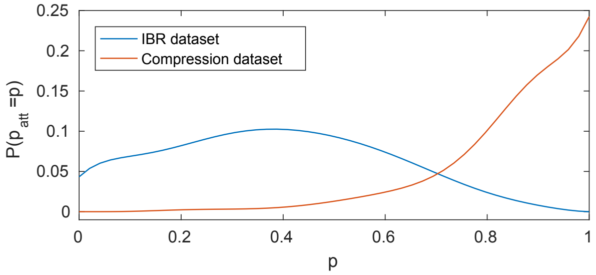


Fig. 6. The probability that the probability of attending a difference is equal to p, plotted separately for two datasets.

$$\begin{aligned} P(data) &= p_{mis} + (1 - p_{mis})\binom{N}{k}(p_{att}{\cdot}p_{det})^k\,(1 - p_{att}{\cdot}p_{det})^{n-k} \\ &= p_{mis} + (1 - p_{mis})\,Binomial(k, N, p_{att}{\cdot}p_{det})\,. \end{aligned} \tag{1}$$

In most practical applications we are mostly interested in predicting p_{det} – the probability that the difference is visible to an observer assuming that he is paying attention to every part of an image. However, we cannot infer p_{det} without knowing p_{att}. Even worse, p_{att} is likely to be different between observers, distortion types, and images. Therefore, it is a random variable rather than a constant.

However, we can estimate the distribution of p_{att}. We found that the distribution of p_{att} depends mostly on the type of the artifact. Therefore, we split our data into datasets with similar distortion types and estimate p_{att} for each one. In every dataset, there are some image parts with very large differences in pixel values between distorted and reference images. If the difference is large enough (20/255 for our datasets), we can assume that the difference is going to be always detectable when observed. Since this corresponds to $p_{det} = 1$, p_{att} is distributed as:

$$P(p_{att} = p) = p_{att}(p) = \frac{1}{|\Omega|}\sum_{(x,y)\in\Omega} Binomial(k(x,y), N, p)\,, \tag{2}$$

where Ω is a set of all pixels (x,y) with large pixel value differences and $|\Omega|$ is the cardinality of that set. For simplicity, we ignore p_{mis} in the above estimate. An example distribution of p_{att} for two datasets is plotted in Figure 6. Now we can incorporate the random variable $p_{att}(p)$ into our statistical model of the marking process and compute the log-likelihood that a set of p_{det} values predicted by a metric describes the marking data collected in the experiment:

$$L = \sum_{(x,y)\in\Theta} \log[p_{mis} + (1-p_{mis}) \cdot \int_0^1 p_{att}(p)\cdot Binomial\,(k(x,y), N, p_{att}(p)\cdot p_{det}(x,y))\; dp]\,, \tag{3}$$

where Θ is the set of all pixels with coordinates x, y. The second line of the equation is the expected value of observing the outcome given the distribution of p_{att}. Equation 3 gives a probabilistic loss function, which we use when fitting the visibility models.

To better understand the importance of modeling probability of attending (p_{att}), let us consider the expected likelihood value (the integral in Equation 3) for two different datasets and for a single pixel instead of the entire image. The plots in Figure 7 show the probability that p_{det} is equal to a certain value if exactly k observers out of 20 have marked the pixel. It can be noted that when $k = 10$ observers mark the pixel in the IBR dataset (upper plot), the probability of detection can range from about 0.5 to 1. This means that if only half of the observers mark a pixel, the difference could still be perfectly detectable ($p_{det} = 1$) with high likelihood, and that the lower number of markings could be attributed to the difficulty in spotting the differences ($p_{att}{\approx}0.5$). In the compression dataset (lower plot), the probability for $k = 10$ is concentrated around $p_{det} = 0.55$, suggesting that almost half of the observers could not detect the difference even if they attended the corresponding spot in the image.

Instead of fitting quality metrics directly to the noisy raw data, the likelihood function from Equation 3 lets us fit the probabilistic estimate of what the true detection thresholds are most likely to be. Furthermore, we do not assume a single estimate of the true detection probability (e.g. mean, mode or expected value), but the distribution of such probabilities, accounting for uncertainty in the data, for example, due to a limited number of observers.

5 VISIBILITY METRICS

We adapt several popular image quality metrics to predict visibility and then use our dataset to train them. To train the metrics with a large number of images, we used a customized version of OpenTuner[2] optimization software. We extended the software so that it could be run on a compute cluster and computing metric predictions was distributed to a large number of nodes. Such parallelization was necessary for more complex metrics, such as HDR-VDP or SSIM. We calibrated the following metrics (refer to Section 2 for a short description of each metric):

[ABS]. The absolute differences (D) between pixel values were divided by the threshold value t and then put into an expression for a psychometric function:

$$p_{det}(x,y) = 1 - \exp\left(\log(0.5)\cdot\left(\frac{D(x,y)}{t}\right)^{\beta}\right)\,, \tag{4}$$

where x, y are pixel coordinates. The two optimized parameters were t and β. The absolute difference D was computed between luma values of distorted and reference images. The predicted $p_{det}(x,y)$ values can be used with Equation 3 to compute the probabilistic loss function.

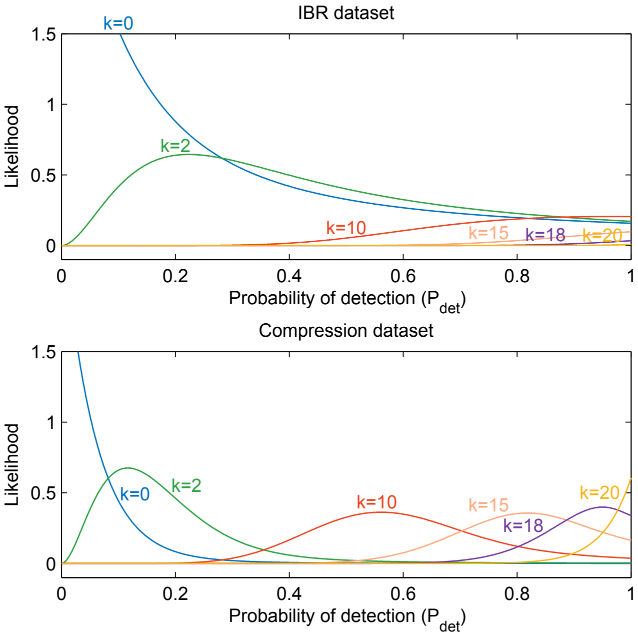


Fig. 7. The probability of detecting the difference for two datasets.

[SSIM]. We found that the values predicted by the SSIM metric are non-linearly related to the magnitude of visibility and benefit from the transformation:

$$D_{SSIM}(x,y) = \frac{1}{\epsilon}\left(\log\left(1 - M_{SSIM}(x,y) + \exp(-\epsilon)\right) + \epsilon\right)\,, \tag{5}$$

where $\epsilon = 10$ and M_{SSIM} is the original SSIM difference map. The transformation makes the D_{SSIM} values positive, in the range 0–1 and increasing with higher image differences. The $D_{SSIM}(x,y)$ values are then processed by the psychometric function from Equation 4. The optimized parameters were t, β and two parameters of the SSIM metric, C_1 and C_2 (refer to Equation 3.13 in [Wang and Bovik 2006]).

[VSI, FSIM]. After transforming the difference maps D_{VSI} and D_{FSIM} into increasing values in the range 0–1, Equation 4 can directly be applied. The fitted parameters were t, β, and three parameters of the VSI metric, C_1, C_2, and C_3 (refer to Equations 4–6 in [Zhang et al. 2014]), or respectively two parameters of the FSIM metric, T_1 and T_2 (refer to Equations 4-5 in [Zhang et al. 2011]).

[CIEDE2000]. The distorted and reference images were transformed into linear XYZ space assuming Rec. 709 color primaries and using a gain-gamma-offset display model simulating our experimental display. The predicted ΔE were transformed into probabilities using the psychometric function (Equation 4). The calibrated parameters were t and β.

[2] http://opentuner.org

[sCIELab]. Our adaptation of sCIELab was identical to the one we used for CIEDE2000, except that the metric was also supplied with the image angular resolution in pixels per visual degree.

[Butteraugli]. In the original Butteraugli implementation the threshold for visible distortions is determined by a constant "good_quality". However, we found that this constant does not correlate well with human experiment results, and better results can be achieved if the map is transformed by the psychophysical function from Equation 4.

[HDR-VDP]. We modified HDR-VDP (v2.2) to better predict our datasets. Firstly, we observed that including orientation-selective bands did not improve predictions for any of our datasets; therefore, we simplified the multi-scale decomposition to all-orientations spatial-frequency bands. Secondly, we improved spatial probability pooling. The original HDR-VDP was calibrated to detection datasets in which one distortion was visible at a time. This enabled using a simplified spatial pooling, in which all differences in an image were added together. However, this resulted in inaccurate results for our datasets, in which distortions vary in their magnitude across an image. We replaced the original pooling with spatial probability summation

$$P_{sp}(x,y) = \mathbf{1} - \exp\left(\log(1 - P(x,y) + \epsilon) * g_\sigma\right)(x,y) , \qquad (6)$$

where $P(x,y)$ is the original probability of detection map (Equation 20 in [Mantiuk et al. 2011]), ϵ is a small constant, and g_σ is the Gaussian kernel. The fitted parameters were the peak sensitivity, a self masking factor (`mask_self`), a cross-band masking factor (`mask_xn`), the p-exponent of the band difference (`mask_p`), and the standard deviation of the spatial pooling kernel (`si_sigma`) in visual degrees.

In the following sections we add the prefix "T-", e. g., T-Butteraugli, to distinguish between the metrics trained by us and their original counterparts.

6 CNN-BASED METRIC

Inspired by the success of convolutional neural networks (CNNs) in many applications, we designed a fully convolutional architecture as a visibility metric and trained it on our dataset.

6.1 Two-branch fully convoluted architecture

Siamese networks gained popularity among tasks that involve difference comparison or a relationship between two images. A Siamese CNN consists of two identical branches that share weights but have distinct inputs and outputs. For example, Bosse et al. [2016] use Siamese architecture to encode distorted and reference patches in the features space, take the difference in that space and input it to the fully-connected layers to predict per-patch quality. After having experimented with such an architecture, we found that better performance can be achieved when a) the difference between distorted and reference patches is computed in image space rather than feature space, and b) fully-connected layers are replaced with convolutional layers.

Figure 8 illustrates our modified architecture, in which one branch is responsible for encoding the difference between distorted and reference patches and the other for encoding a reference patch. Unlike the Siamese architecture, our approach uses separate weights for each branch, as each branch encodes different information. Both branches are concatenated to preserve all features. The visibility map is reconstructed from the feature-space representation using 3 deconvolution layers. Using convolutional layers instead of fully connected layers reduced the number of parameters and improved the ability of the metric to generalize to different datasets.

Formally we denote R as the reference patch, D as the distorted patch. We also define two mapping functions $F_{w^d_{conv}}$ and $F_{w^r_{conv}}$, where w^d_{conv} and w^r_{conv} represent the weights for convolutional layers of the difference branch and the respective weights for the reference branch. In addition, we use a mapping function $F_{w_{dec}}$ where w_{dec} represents the weights for deconvolution layers with the skip connection mechanism. Our metric $M_w(D,R)$ is formulated as:

$$M_w(D,R) = F_{w_{dec}}(Concatenate(F_{w^d_{conv}}(D-R), F_{w^r_{conv}}(R))) \quad (7)$$

Convolutional layers. Since our training dataset does not contain the necessary amount of images to perform a full training, we perform a fine-tuning of our network by initializing the weights via transfer learning from AlexNet implementation [Krizhevsky et al. 2012]. We found that the feature maps generated with two convolutional layers achieve similar results as with 5 layers, and consequently, we remove the last 3 convolutional layers. The two convolution layers alternate with pooling layers. We use the rectified linear unit (ReLU) as the activation function. In addition, to avoid overfitting, we set the dropout value to 0.5.

Deconvolutional layers. The concatenation is followed by the deconvolution layers, which reconstruct the final visibility map. To prevent checker-board patterns, each deconvolution is performed as a sequence of upsampling and convolution operations. Such patterns are a common problem if deconvolution is realized as a transposed convolution. Further refinement is achieved by skip-connections, which concatenate feature maps from the convolution layers of the difference branch with the deconvolution layers. Such skip-connections create new paths inside the neural network, and help to avoid the issue of vanishing or exploding gradients.

6.2 Training and testing

The network is trained by minimizing the likelihood function from Equation 3. The images are split into patches of 48×48 pixels, without overlapping parts. We found that patches of that size better preserve high-frequency details in the visibility maps. To increase the size of dataset and prevent overfitting, we augment the data by horizontal and vertical flips the rotations of 90, 180, 270 degrees. We also ignore all the patches for which there is no difference between their distorted and reference versions. The total number of patches is approximately 400,000. We train the network in 50,000 iterations, with a learning rate of 0.00001 and a learning decay rate of 0.9. To speed up the training process, we use a mini-batch technique with 48 batch size. The network is trained using Adaptive Moment Estimation (Adam), which calculates adaptive learning rates for the parameters. The CNN architecture is implemented in TensorFlow 1.4 [3]. We perform training and testing exploiting Tensorflow GPU

[3] https://www.tensorflow.org/

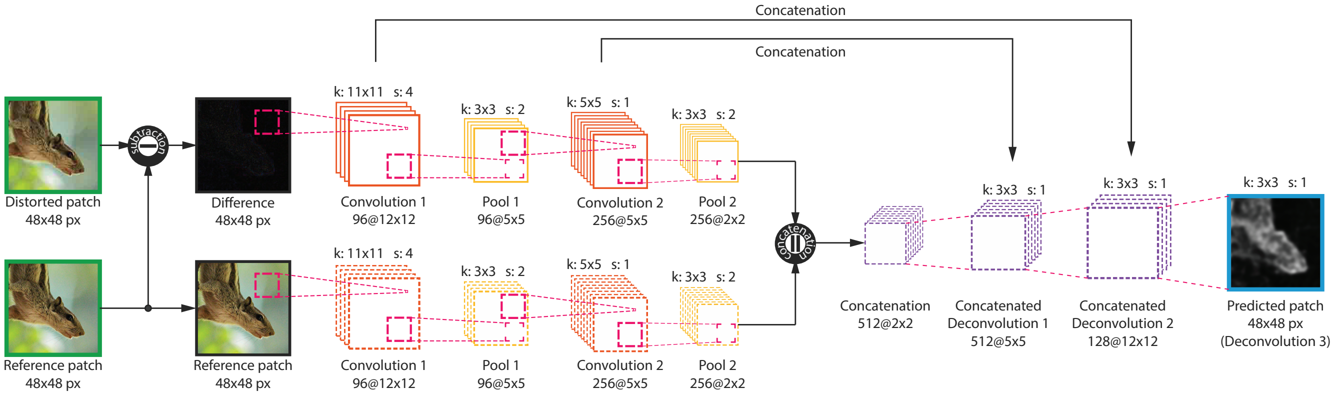


Fig. 8. Two-branch fully convolutional CNN architecture with the difference branch. The difference branch takes a difference between the distorted and reference images as the input, while the other branch accepts the reference image. The output is a visibility map, achieved by regression, with the same size as the input patch. Each branch contains two convolution layers with 11×11 kernel and stride 4 followed by another layer with 5×5 kernel and stride 1. The deconvolution section uses convolution layers with 3×3 kernel and stride 1.

support on an NVIDIA GeForce GTX 980 Ti.

To predict a visibility map for a full-size image, we split it into 48×48 patches with 42-pixel overlap, infer a visibility map for each patch and finally assemble the complete map by averaging each pixel shared by the overlapping patches. Prediction for an 800×600 pixel distorted image takes approximately 3.5 seconds.

7 METRIC RESULTS

To compare metric predictions, we computed the results for 5-fold cross-validation with an 80:20 split between training and testing sets. The split ensured that the testing set did not contain any of the scenes used for training, regardless of the distortion level. The results for all metrics, shown in Figure 9, indicate the CNN metric compares favorably to other metrics, where good performance can also be observed for T-Butteraugli and T-HDR-VDP. We did not find any evidence in our dataset that the color difference formula (T-CIEDE2000) offers an improvement over T-ABS computed between luma values. T-sCIELab performed slightly better than T-ABS, but for many datasets they are quite comparable. It is worth noting that some datasets are more difficult to predict (lower likelihood) than the others; notably, COMPRESSION was the most difficult dataset, followed by PERCEPTIONPATTERNS.

It must be noted that the metrics have been retrained and modified or extended to better predict our dataset. In Figure 10 we show the predictions of the original metrics compared to the predictions of the newly retrained metrics. In general, the original metrics were too sensitive and predicted mostly invisible distortions. The code and the parameters of the retrained metrics can be found in the supplementary materials.

Figure 11 shows metric predictions for a few selected interesting cases from our dataset. The complete set of predictions can be found in the supplementary materials. Image *uncorrelated noise* contains three noisy circular patterns modulated by a Gaussian envelope, presented on the background of a lower amplitude noise. The reference

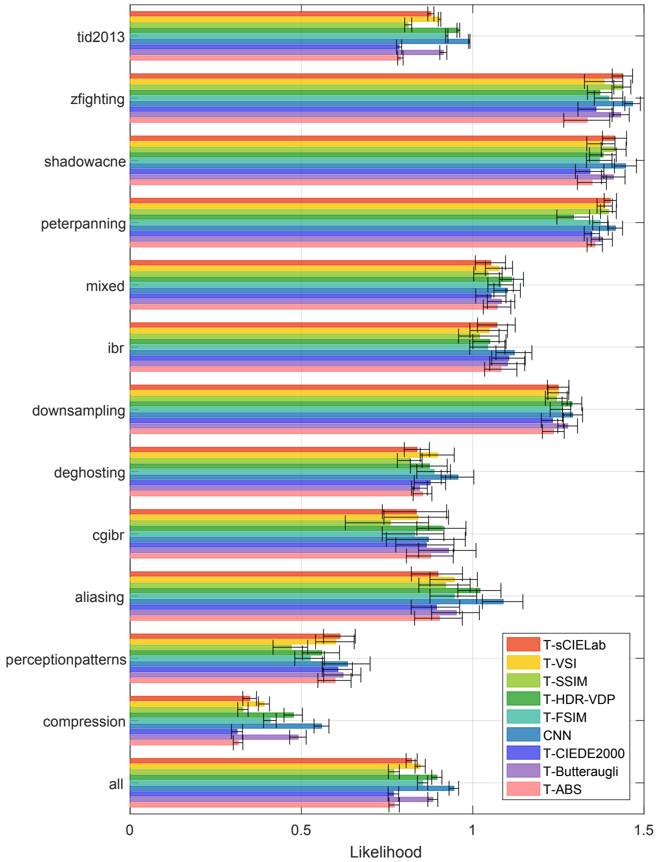


Fig. 9. The results of cross-validation of the quality metrics for each dataset and for all datasets together. The error bars denote standard error when averaging across images in the dataset.

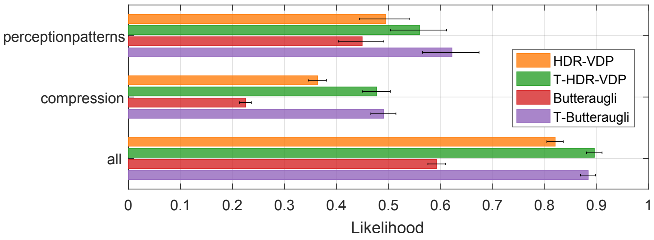


Fig. 10. Improvement in prediction performance between the original and retrained metrics. For brevity, only two selected subsets and overall performance are reported on.

image contains only the background pattern, without the circular patterns, but the random seed used to generate the background noise was different than in the distorted image. While observers could not spot the difference in the background noise pattern in a side-by-side presentation, such a difference triggered detection for all metrics. T-Butteraugli and CNN performed better, partially ignoring the difference in background noise..

The *gorilla* image was distorted by image compression. It contains complex masking patterns, which modulate the visibility of the distortions. The simple metrics, such as T-SSIM, T-FSIM and T-VSI, failed to predict such masking. More advanced metrics, CNN, T-Butteraugli and T-HDR-VDP, were more accurate in indicating the higher visibility of distortions on the gorilla's face while for the chest area was marked correctly only by CNN.

The *peter panning* image contains distortion caused by shadow mapping where the shadow is detached from an object casting it [Piórkowski et al. 2017]. The images also contain small differences in pixel values due to shading and post-processing effects in the game engine. T-FSIM metric failed to mask such small differences (best seen in the electronic version), while other metrics correctly ignore the visibility of those small differences. T-Butteraugli and T-HDR-VDP tend to excessively expand the region with the difference.

The *car* image contains distortion on the body of the car, but there is also a readily visible noise pattern in the bottom right corner. Very few observers marked that noise pattern in the corner, as most people were paying attention to the car. This is an example of a case in which we cannot treat observers' markings as ground truth and we need the statistical model from Section 4 to correctly model uncertainty in the data. It is worth noting that all metrics correctly predicted the visibility of the noise pattern.

The *classroom* image contains a rendering of the same scene, but from a slightly different camera position in the distorted and reference images. While the observers could not notice any differences, such pixel misalignment triggered a lot of false positives for most metrics. CNN could only partially compensate for pixel misalignment.

8 APPLICATIONS

In this section we present three applications that demonstrate the utility of the best performing visibility metrics. In Section 8.1 we investigate visually lossless JPEG compression controlled by the metrics. In Section 8.2 the metrics are used to determine maximum subsampling level for a single-image super-resolution. In Section 8.3 we use CNN metric to adjust content-dependent watermarks so that their intensity is maximized while they remain imperceptible.

8.1 Visually lossless image compression

The objective of visually lossless image compression is to compress an image to the lowest bit rate while ensuring that compression artifacts remain invisible. JPEG quality setting is rarely manually fine-tuned and most images are saved using the default quality setting, which is typically set to 90 (out of 100) in a majority of software, producing much larger files than needed. In this section, we show that visibility metrics can be used to automatically find the JPEG quality setting for visually lossless compression, which can significantly reduce the file size.

To validate metrics' performance in this application, we conducted an additional experiment, in which observers indicated the lowest JPEG quality setting for which distortions remained invisible in a side-by-side presentation. In the experiment, we use the standard JPEG codec (libjpeg[4]). To avoid using the same images for training, we used Rawzor's free dataset[5] which contains a rich set of image content. The images are cropped to 960 × 600 pixels to fit on our screen. The images are distorted by compression with a standard JPEG codec using a range compression qualities. The experimental procedure involved selecting a distorted image from 4 presented, where only one image was distorted (four-alternative-forced-choice protocol). The quality setting was adaptively adjusted using the QUEST method. Between 20 and 30 trials were collected per image to find the quality settings at which an observer could select a distorted image with 75% confidence probability level in the QUEST procedure. 10 observers completed the experiment. To predict the visually lossless quality setting for JPEG compression, we take the maximum value of the metrics' predicted visibility map, which gives a conservative estimate. A similar approach was in [Alakuijala et al. 2017]. We search the quality settings from 0 to 98 and chose the lowest quality setting that produced the visibility map of maximum value less than 0.5, which corresponds to 50% of observers spotting the difference. We select the best three metrics, CNN, T-HDR-VDP and T-Butteraugli, and their original versions, HDR-VDP and Butteraugli, for evaluation.

The results of the experiment and the predictions of the top-performing visibility metrics are shown in Figure 12. The blue line denotes the median value computed across observers and the blue shaded area represents the range between the 20th and 80th percentiles. CNN, T-HDR-VDP and T-Butteraugli correlate reasonably well with the experiment results, although the distortions in scenes 9 and 11, were under-predicted by the trained metrics. The most visible distortions in those images are due to contouring in smoothly shaded regions. Such distortion types were missing in our training set, which could lead to the worse-than-expected performance.

We quantify the accuracy of metrics' predictions as the mean squared error (MSE) between the predictions and levels found in the experiment. Among the top three metrics, CNN's performance is the best with a MSE of 367.7 followed by T-HDR-VDP with a MSE of 467.5 and T-Butteraugli with a MSE of 479.4. The original (untrained)

[4] https://github.com/LuaDist/libjpeg
[5] http://imagecompression.info/test_images/

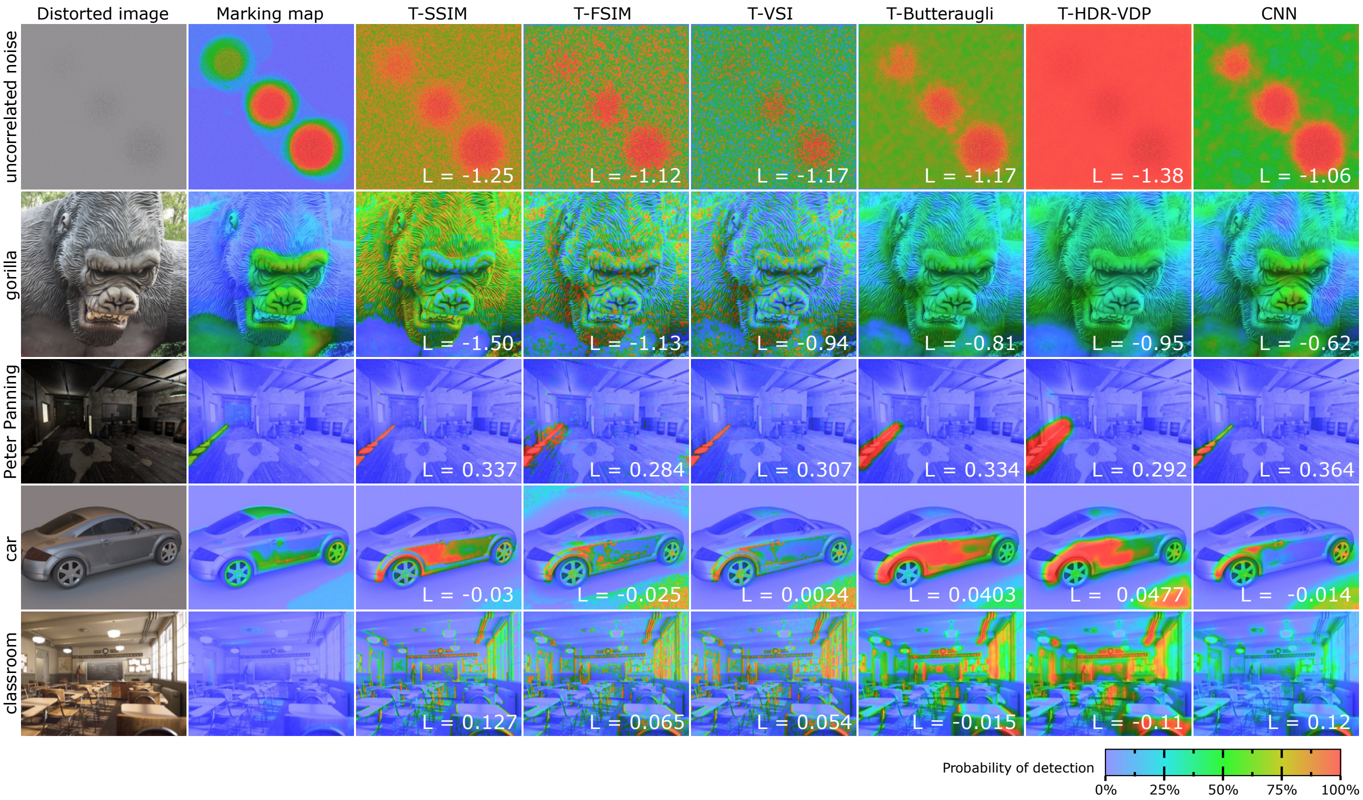


Fig. 11. Distorted images, observers' markings and metric predictions for a few selected images from the dataset. Metric predictions must be viewed in color.

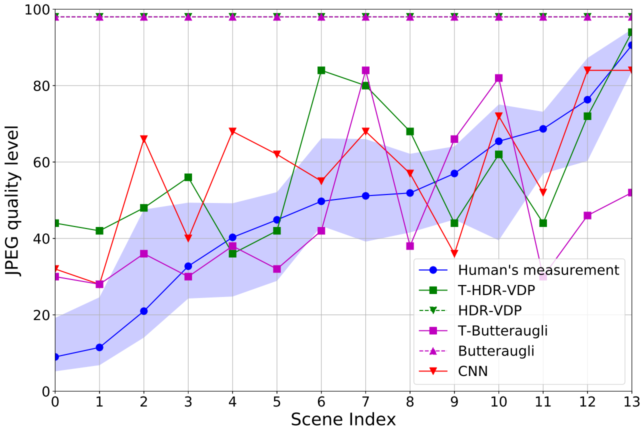


Fig. 12. Results for visual lossless image compression. The blue line is the median and the blue shaded region is 20th and 80th percentile of manual adjustment.

versions of HDR-VDP and Butteraugli resulted in strongly over-predicted visibility of JPEG artifacts. This result confirms that, with our proposed dataset, the trained metrics could generalize well to different distortion types (we did not include JPEG distortions in the dataset) and different content. Compared with the common practice of setting the quality to a fixed value of 90, the best CNN metric could help to reduce file size on average by 60% for the selected set of images.

8.2 Super-resolution from downsampled images

Super-resolution (SR) methods reconstruct a higher-resolution image from a low resolution (LR) image or images. In this section we consider a single-image super-resolution, where the method is used to reduce image resolution in a such a way that visually indistinguishable full-resolution image can be reconstructed from a low-resolution image.

We consider three scenes generated with Arnold renderer[6] (Figure 13). The scenes depict objects on a quasi-uniform background with small amount of noise, caused by global illumination approximation. Simple scenes were selected to make easier for observers to find the distortions in an experiment.

We employ the traditional projection onto convex set (POCS [Panda et al. 2011]) algorithm for reconstructing the SR image from a single LR image. We use a MATLAB implementation, generating LR images by applying downsampling factors from 1.1 to 6.0.

To validate the metric performance, we collected the user data on the distortion visibility for various downsampling levels with respect to the full resolution reference. We ran the same 4AFC QUEST procedure as in Section 8.1. Between 20 and 40 QUEST trials

[6] https://www.solidangle.com/arnold/

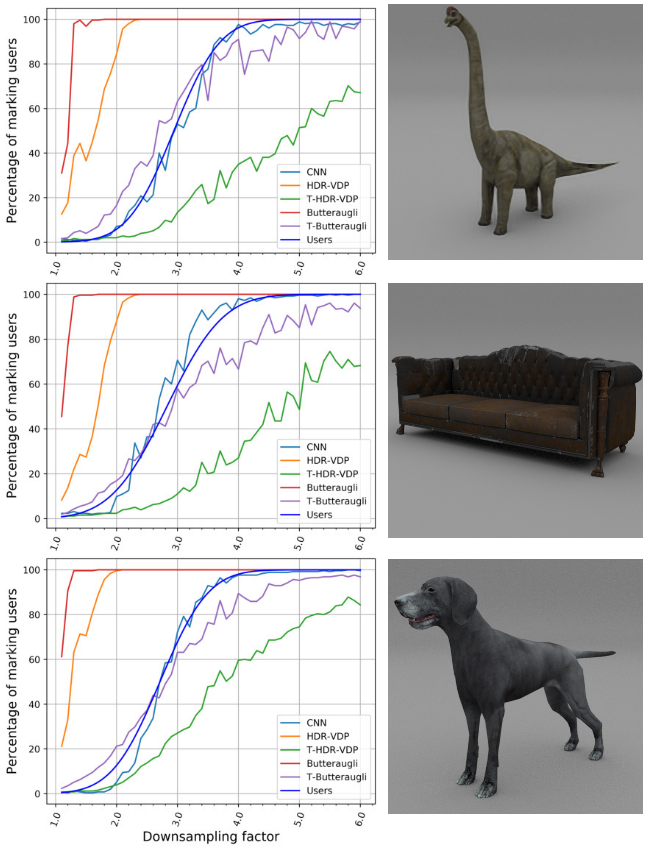


Fig. 13. Comparison of the metric performance in the super-resolution application. Three scenes (the right column) are considered for which the user data on the distortion visibility have been collected as depicted in blue in the corresponding graphs (the left column). The graphs show also the metric prediction of the user data for various downsampling factors.

were collected per scene and per observer to find the detection threshold for level of downsampling factor. 20 observers completed the experiment.

The blue line in Figure 13 shows the percentage of the population that selected lower downsampling than the one shown on the x-axis. The smooth shape of that line was estimated by fitting a normal distribution to the collected thresholds and then plotting the cumulative of that distribution. To plot metric predictions, we take the maximum value in the visibility difference map to account for the most visually critical distortion, similarly as in the JPEG application. To compare metric performance, we compute MSE between each metric prediction and the user data for all downsampling levels. After averaging the MSE values for all three scenes, CNN resulted in the smallest error of 19.95, followed by T-Butteraugli with the error of 96.96.

8.3 Content-adaptive watermarking

Watermarking is the technique of changing the signal in order to embed information about the data. For visual media, the watermark is usually a logo or text superimposed on an image that remains imperceptible. Our application is inspired by watermarking techniques and its objective is to show that our metric is able to detect correctly the areas where contrast masking lowers the visibility of a watermark. Contrast masking is an important characteristic of the HVS that results in the reduced visibility of a distortion that is imposed on the image pattern with similar spatial frequencies and orientations [Chandler 2013; Mantiuk et al. 2011]. Contrast masking becomes stronger with increasing contrast of the masker, which we employ to increase the watermark intensity.

To add a watermark, we add a grid of small gray-shaded watermark patches (64×64 pixels) to the reference image. We start with high-intensity watermarks so that the differences are clearly visible. Then, we employ the CNN metric to generate the distortion visibility map and to determine which watermark patches result in visible difference (we use a custom threshold of 6%). The intensity of all watermark patches in which at least one pixel contains visible difference is reduced by 1 (the minimum step). This process is repeated until the metric does not detect any visible difference. Note that this procedure is equivalent to pooling the maximum value per patch, rather than per image as was done for JPEG and super-resolution applications.

Although our metric was not trained for this particular distortion type, the watermarks optimized by our metric remain imperceptible in side-by-side comparison with their references (Figure 14). This result demonstrates that our metric can deal with distortion types not present in the training dataset. For a better illustration, we also provide the watermarks multiplied by a constant, which show that our metric is able to correctly detect the areas that mask distortions. For example in the rocks area (top row) and city lights (bottom row), which contain high frequencies of relatively high contrast, visual masking becomes stronger and our metric allows for higher intensity of the watermark. The watermark intensity remains small in the smooth gradient regions.

9 LIMITATIONS

Our approach to data collection and modeling is not free from limitations. The experimental task, although intuitive and very efficient, leads to ambiguity between the ability to find and to detect distortion. We address this problem by inferring the likely probability of observing a particular distortion. However, we also found that using smaller images, such as the 512×512-pixel crops used in the COMPRESSION set, greatly reduces the effect of visual search and allows us to collect more consistent data. Finally, our dataset does not capture variation of visibility with the viewing distance and absolute luminance level, the effects that are modeled by some visibility metrics, such as HDR-VDP. The data on the effects of both factors still needs to be collected.

10 CONCLUSIONS

Prediction of visible distortions in images is a challenging task and no existing visibility metric can provide robust predictions. The challenge comes from the complexity of human visual perception, but also from the lack of sufficiently large datasets which could be used to train and validate visibility metrics. This work contributes

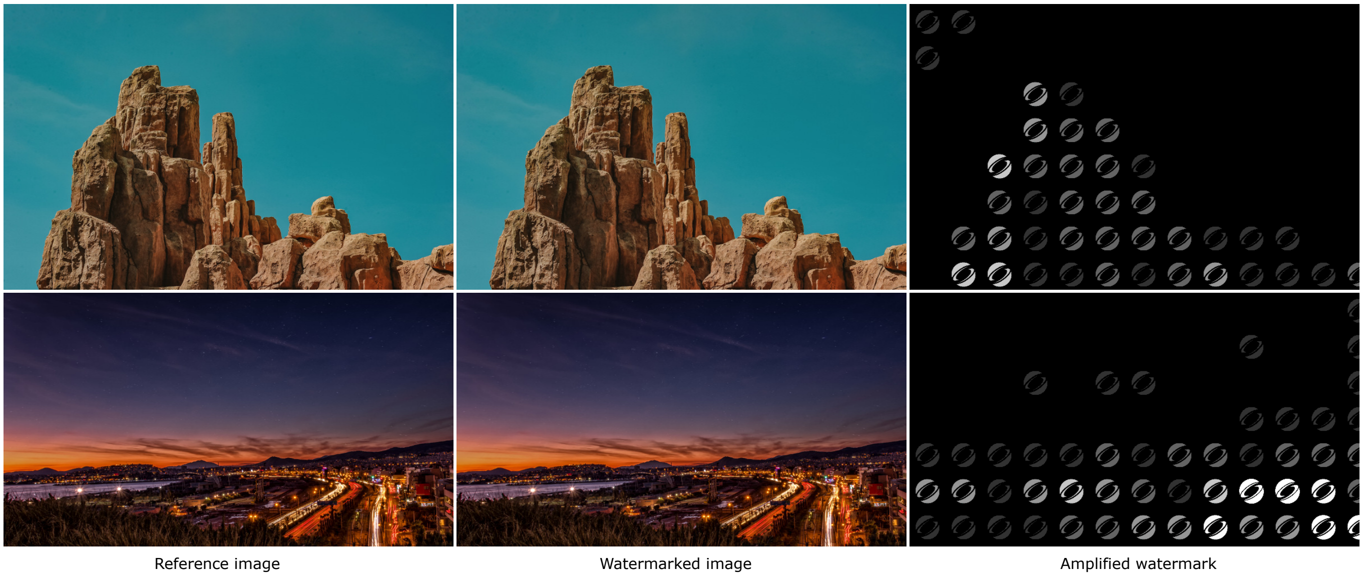


Fig. 14. Examples of content-adaptive watermark application. Our metric applies a brighter watermark in high frequency and contrast areas, like the rocks in the first example and city lights in the second example.

towards creating such a large dataset. We propose to use an efficient experimental method, which, however, can misrepresent the true detection performance when the observers are not able to locate all visible distortions. For that reason, we create a statistical model, which can describe uncertainty in the data and serve as a loss function for training metrics.

We use the dataset to train the existing visibility metrics and demonstrate that their performance can be much improved. We also train a new CNN-based metric, which vastly outperforms existing metrics. We demonstrate the utility of our CNN metric in three practical applications: visually lossless JPEG compression, superresolution, and watermarking. In future work, we would like to investigate other applications, such as automatic simplification of 3D scene complexity (textures, geometry, shading) while ensuring no loss of quality, or comparison of rendering methods while ignoring differences below the visibility threshold.

11 ACKNOWLEDGMENTS

The authors would like to thank Shinichi Kinuwaki for helpful discussions. The project was supported by the Fraunhofer and Max Planck cooperation program within the German pact for research and innovation (PFI). This project has also received funding from the European Union's Horizon 2020 research and innovation programme, under the Marie Skłodowska-Curie grant agreements No 642841 (DISTRO), No 765911 (RealVision) and from the European Research Council (ERC) (grant agreement n° 725253/EyeCode). The project was partially funded by the Polish National Science Centre (decision number DEC-2013/09/B/ST6/02270).

REFERENCES

Vamsi K. Adhikarla, Marek Vinkler, Denis Sumin, Rafal K. Mantiuk, Karol Myszkowski, Hans-Peter Seidel, and Piotr Didyk. 2017. Towards a quality metric for dense light fields. In *Computer Vision and Pattern Recognition*.

Jyrki Alakuijala, Robert Obryk, Ostap Stoliarchuk, Zoltan Szabadka, Lode Vandevenne, and Jan Wassenberg. 2017. Guetzli: Perceptually guided JPEG encoder. *arXiv:1703.04421* (2017).

M. M. Alam, K. P. Vilankar, David J Field, and Damon M Chandler. 2014. Local masking in natural images: A database and analysis. *Journal of Vision* 14, 8 (jul 2014), 22. DOI: https://doi.org/10.1167/14.8.22

Seyed Ali Amirshahi, Marius Pedersen, and Stella X Yu. 2016. Image quality assessment by comparing CNN features between images. *Journal of Imaging Science and Technology* 60, 6 (2016), 60410–1.

Tunç Ozan Aydin, Rafał Mantiuk, Karol Myszkowski, and Hans-Peter Seidel. 2008. Dynamic range independent image quality assessment. *ACM Transactions on Graphics (TOG)* 27, 3 (2008), 69.

Simone Bianco, Luigi Celona, Paolo Napoletano, and Raimondo Schettini. 2016. On the use of deep learning for blind image quality assessment. *arXiv:1602.05531* (2016).

Sebastian Bosse, Dominique Maniry, Klaus-Robert Mueller, Thomas Wiegand, and Wojciech Samek. 2016. Full-reference image quality assessment using neural networks. In *Int. Work. Qual. Multimdia Exp.*

S. Bosse, D. Maniry, K. R. Müller, T. Wiegand, and W. Samek. 2018. Deep neural networks for no-reference and full-reference image quality assessment. *IEEE Transactions on Image Processing* 27, 1 (2018), 206–219.

Sebastian Bosse, Dominique Maniry, Thomas Wiegand, and Wojciech Samek. 2016a. A deep neural network for image quality assessment. In *IEEE International Conference on Image Processing (ICIP)*. IEEE, 3773–3777.

Sebastian Bosse, Dominique Maniry, Thomas Wiegand, and Wojciech Samek. 2016b. Neural network-based full-reference image quality assessment. In *Proceedings of the Picture Coding Symposium (PCS)*. 1–5.

Martin Čadík, Robert Herzog, Rafał Mantiuk, Radosław Mantiuk, Karol Myszkowski, and Hans-Peter Seidel. 2013. Learning to predict localized distortions in rendered images. In *Computer Graphics Forum*, Vol. 32. 401–410.

Martin Čadík, Robert Herzog, Rafał K. Mantiuk, Karol Myszkowski, and Hans-Peter Seidel. 2012. New measurements reveal weaknesses of image quality metrics in evaluating graphics artifacts. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 31, 6 (2012), 147.

Damon M Chandler. 2013. Seven challenges in image quality assessment: Past, present, and future research. *ISRN Signal Processing* (2013), Article ID 905685. DOI: https://doi.org/doi:10.1155/2013/905685

Scott Daly. 1993. The visible differences predictor: An algorithm for the assessment of image fidelity. In *Digital Images and Human Vision*. MIT Press, 179–206.

Scott J Daly. 1992. Visible differences predictor: An algorithm for the assessment of image fidelity. In *Human Vision, Visual Processing, and Digital Display III*, Vol. 1666. International Society for Optics and Photonics, 2–16.

Robert Herzog, Martin Čadík, Tunç O. Aydin, Kwang In Kim, Karol Myszkowski, and Hans-Peter Seidel. 2012. NoRM: No-reference image quality metric for realistic image synthesis. *Computer Graphics Forum* 31, 2 (2012), 545–554.

Le Kang, Peng Ye, Yi Li, and David Doermann. 2014. Convolutional neural networks for no-reference image quality assessment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1733–1740.

Kanita Karađuzović-Hadžiabdić, Jasminka Hasić Telalović, and Rafał K Mantiuk. 2017. Assessment of multi-exposure HDR image deghosting methods. *Computers & Graphics* 63 (2017), 1–17.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 1097–1105.

Weisi Lin and C.-C. Jay Kuo. 2011. Perceptual visual quality metrics: A survey. *J. Visual Communication and Image Representation* (2011), 297–312.

Jeffrey Lubin. 1995. *Vision models for target detection and recognition.* World Scientific, Chapter A Visual Discrimination Model for Imaging System Design and Evaluation, 245–283.

Rafal Mantiuk, Kil Joong Kim, Allan G. Rempel, and Wolfgang Heidrich. 2011. HDR-VDP-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions. *ACM Transactions on Graphics (Proc. SIGGRAPH)* (2011). Article 40.

A.K. Moorthy and A.C. Bovik. 2010. A two-step framework for constructing blind image quality indices. *IEEE Signal Processing Letters* 17, 5 (2010), 513 –516.

Manish Narwaria and Weisi Lin. 2010. Objective image quality assessment based on support vector regression. *IEEE Transactions on Neural Networks* 21, 3 (2010), 515–9.

S. S. Panda, M. S. R. S. Prasad, and G. Jena. 2011. POCS based super-resolution image reconstruction using an adaptive regularization parameter. *CoRR* abs/1112.1484 (2011). arXiv:1112.1484 http://arxiv.org/abs/1112.1484

Rafał Piórkowski, Radosław Mantiuk, and Adam Siekawa. 2017. Automatic detection of game engine artifacts using full reference image quality metrics. *ACM Transactions on Applied Perception (TAP)* 14, 3 (2017), 14.

Nikolay Ponomarenko, Lina Jin, Oleg Ieremeiev, Vladimir Lukin, Karen Egiazarian, Jaakko Astola, Benoit Vozel, Kacem Chehdi, Marco Carli, Federica Battisti, and C.-C. Jay Kuo. 2015. Image database TID2013: Peculiarities, results and perspectives. *Signal Processing: Image Communication* 30 (Jan. 2015), 57–77. DOI: https://doi.org/10.1016/j.image.2014.10.009

M.A. Saad, A.C. Bovik, and C. Charrier. 2012. Blind image quality assessment: A natural scene statistics approach in the DCT domain. *IEEE Trans. on Image Processing* 21, 8 (2012), 3339–3352.

H.R. Sheikh, M.F. Sabir, and A.C. Bovik. 2006. A statistical evaluation of recent full reference image quality assessment algorithms. *IEEE Trans. on Image Processing* 15, 11 (2006), 3440–3451.

Huixuan Tang, N. Joshi, and A. Kapoor. 2011. Learning a blind measure of perceptual image quality. *Proc. of IEEE Computer Vision and Pattern Recognition* (2011), 305–312.

Zhou Wang and Alan C. Bovik. 2006. *Modern image quality assessment.* Morgan & Claypool Publishers.

L. Zhang, Y. Shen, and H. Li. 2014. VSI: A visual saliency-induced index for perceptual image quality assessment. *IEEE Transactions on Image Processing* 23, 10 (2014), 4270–4281. DOI: https://doi.org/10.1109/TIP.2014.2346028

L. Zhang, L. Zhang, X. Mou, and D. Zhang. 2011. FSIM: A feature similarity index for image quality assessment. *IEEE Transactions on Image Processing* 20, 8 (2011), 2378–2386. DOI: https://doi.org/10.1109/TIP.2011.2109730

X. Zhang and B. A. Wandell. 1997. A spatial extension of CIELAB for digital color-image reproduction. *Journal of the Society for Information Display* 5, 1 (1997), 61.