NUMERICAL METHODS AND DATA ANALYSIS

Veiling glare removal: synthetic dataset generation, metrics and neural network architecture

A.V. Shoshin [1,2], *E.A. Shvets* [1]
[1] *Kharkevich Institute for Information Transmission Problems, RAS,*
Bolshoy Karetny per. 19, build.1, Moscow, 127051, Russia,
[2] *Moscow Institute of Physics and Technology (State University),*
Institutsky per. 9, Dolgoprudny, 141701, Russia

Abstract

In photography, the presence of a bright light source often reduces the quality and readability of the resulting image. Light rays reflect and bounce off camera elements, sensor or diaphragm causing unwanted artifacts. These artifacts are generally known as “lens flare” and may have different influences on the photo: reduce contrast of the image (veiling glare), add circular or circular-like effects (ghosting flare), appear as bright rays spreading from light source (starburst pattern), or cause aberrations. All these effects are generally undesirable, as they reduce legibility and aesthetics of the image. In this paper we address the problem of removing or reducing the effect of veiling glare on the image. There are no available large-scale datasets for this problem and no established metrics, so we start by (i) proposing a simple and fast algorithm of generating synthetic veiling glare images necessary for training and (ii) studying metrics used in related image enhancement tasks (dehazing and underwater image enhancement). We select three such no-reference metrics (UCIQE, UIQM and CCF) and show that their improvement indicates better veil removal. Finally, we experiment on neural network architectures and propose a two-branched architecture and a training procedure utilizing structural similarity measure.

Keywords: lens flare, veiling glare, image enhancement, deep learning, synthetic data.

Citation: Shoshin AV, Shvets EA. Veiling glare removal: synthetic dataset generation, metrics and neural network architecture. Computer Optics 2021; 45(4): 615-626. DOI: 10.18287/2412-6179-CO-883.

Introduction

Glare removal is an important area of research in modern image enhancement. The presence of a glare in an image typically reduces its legibility and aesthetics: glare can cause loss of information in underlying pixels, make the image harder to interpret, reduce contrast and color variety. Practical applications of glare removal are numerous: uterine cervix cancer detection [1] and enhancement of laparoscopic images [2], vehicle license plate recognition [3], eyeglass reflection glare from frontal face shots [4] – both for aesthetic improvement of photos and as a part of recognition pipeline - and others. Typically, glare is caused by bright sources of light in the scene, however it can also be caused by water droplets on the camera lens [5].

A relatively recent review on removal of lens flare has been published in 2017 [6]. The approaches to glare removal vary and can be divided into two categories: approaches that modify the procedure of taking the image and approaches that modify an already taken image. An example of the first approach is described [5] for the water droplet flare removal: by turning on and off each element in the optical shutter array during the shooting, authors get several different images; with these images, they can locate the droplet position by comparing images' brightness. Then, by turning off elements that correspond to droplets they obtain image without the glare.

Another similar approach is described in [7] – where instead of blocking light inside the camera, authors propose to put a high frequency occlusion mask between the scene and the camera. The resulting occluded scene is shot with multiple exposures (the original scene without occlusion is shot as well); authors propose a method of estimating the glare component using photos of the occluded scene. The glare component is then subtracted from the unoccluded photo – which yields the photo without the veiling glare effect. The advantage of this method is that there is no need to change camera elements or the hardware. However, there are two clear drawbacks: firstly, to get one clear image one needs to take several shots of the scene; secondly, the technique requires a controlled environment, where occlusion mask can be put, and all the images can be shot with perfect alignment.

Removing the glare from a given image without access to the shooting procedure is generally a more difficult problem. There is a wide variety of possible glares, and the scenes can differ a lot – so localizing the glare in an arbitrary image can be a very challenging task. To solve it, most image-based approaches typically focus on a specific type of images and flares. Narrowing the expected variety of data allows to formulate the model of the image and the glare – and this model is then used for glare removal. For example, the eyeglass reflection removal from the frontal face photos in [4] is achieved by building a detailed model of an eyeglass reflection – the sparsity, piecewise constancy and specific color tint attributes of the reflection are formalized and utilized in the solution.

Another way to make the task easier is to utilize the user to help the localization process. An example of such semi-automatic process is proposed in [8]: the localization of flare on the image is done manually by user input; then the algorithm searches for the image region most similar to one under the flare and the flared area is then inpainted with the pixels of found "similar" area.

Finally, machine learning and deep networks can be utilized for glare removal without the necessity to formulate an accurate model of the glare. For example, in [3] authors use deep learning to remove flares from the images of license plates as a part of the recognition process in Electronic Toll Collection system. Their neural network consists of two main parts: glare detection and glare removal subnetworks.

Notably, the data used in [3] is simple and homogeneous – it contains the cropped images of license plates with digits and letters of a known font in known positions. A much more variable synthetic dataset was used to train an end-to-end flare removal network in a recent paper [4]. However, deep learning application to glare removal is still very limited – despite its general popularity in the computer vision field. One possible reason for this phenomenon is the lack of large relevant datasets. Table 1 presents the information regarding the datasets we were able to find in the literature that considers (or at least is related to) the glare removal problem.

Table 1. Number of images in datasets of different glare removal approaches

Article name	Number of images in dataset
[1] Automatic glare removal in reflectance imagery of the uterine cervix	111
[2] Fast Detection and Removal of Glare in Gray Scale Laparoscopic Images	10K
[3] Single Image Glare Removal Using Deep Convolutional Networks	83.6K
[4] Anti-glare: Tightly constrained optimization for eyeglass reflection removal	2.7K
[5] Removal of Glare Caused by Water Droplets.	100
[7] Use of an Occlusion Mask for Veiling Glare Removal in HDR Images	52
[8] Removing lens flare from digital photographs	1
[11] Veiling glare in high dynamic range imaging	36
[12] Liquid-filled camera for the measurement of high-contrast images.	1
[13] Anti-veiling-glare glass input window for an optical device and method for manufacturing such window	1

As seen from the table, most of these papers consider different types of images (digital photographs of large scenes, laparoscopic images, face images or cropped license plate images) and different types of glare: veiling glare, specular reflections, reflections from water droplets, etc. Different combinations of image type and glare typically need different solutions. Most of the used datasets consider only dozens of images and are too small to use as a training dataset. Three large datasets correspond to very narrow cases: [2] provides images of laparoscopic images distorted by specular reflection; [4] considers the face images with eyeglass reflection glare and [3] uses a dataset of road signs with added synthetic glares.

One specific type of glare is veiling glare – a global illumination effect that appears in the presence of a strong light source and lowers the contrast of the image, reducing the legibility of the image by partially or completely obscuring the details of faint objects. This problem is especially pronounced for High Dynamic Range (HDR) photography, which is typically used to capture scenes containing both a bright light source and a dark, underlighted region. Due to camera imperfections, the light from the source scatters over the whole image; and while the portion of scattered light is low, the intensity of the light source is much higher than that of the under-lighted parts of the image – so this stray light obscures the details in the dark part. Effectively, veiling glare limits the dynamic range of the camera [9] and the range of luminance [10] that can be accurately measured by it – so this is an important problem to solve.

As with other types of glare, there are methods for fighting veiling glare during the shooting procedure [11], by changing and improving camera parts and lens optics (by utilizing special lens coatings [12] or anti-glare window glass [13]) and on image-level – for example, Talvala et.al [11] assumes that glare spread function (GSF) is constant across the image, and therefore veiling glare compensation can be achieved by a single deconvolution, which is then found using gradient decent [14]. There are two main problems with such approach. Firstly, recovered area becomes noisy due to the fact that the recorded image simply does not have enough information about the captured scene. Secondly, in reality GSF is not constant and changes its color and shape across the image, so the result is not idealeven when the optimal GSF is found.

1. Synthetic data

For good performance, flare-removal networks would require huge amounts of training data consisting of images shot in the same scene with and without glare. Given the complexity of gathering such a dataset, a viable solution is generation of synthetic images. Veiling glare can appear in very different types of images. As seen from the Table 1, there is no valid, large dataset that can be used to train a glare removal network. Therefore, we have to collect our own dataset suited to the problem.

One way to create realistic synthetic data with glares is described in [15]. Using special programs (e.g. utilizing Constructive Solid Geometry technique), it is possible to render particular camera's lens optics system. After that, ray tracing allows to simulate real ray's paths, which eventually results in the presence of desired effects in the images – such as veiling flare or other flare types. Another similar approach, described in [16], formulate a mathematical model of lens optics and uses ray tracing for flare rendering.

Such methods create beautiful, real-like flares, but for most of them it takes too long to render even one image – at least an hour; others require accurate models of lens system. Using such an approach for creation of a dataset large enough for CNN training that would be applicable to any camera is infeasible. Another possibility is to use packages of pregenerated flare samples, such as provided by OpenGL [17]. However, the number and diversity of pregenerated glares are also very limited (dozens of examples). It is not enough for efficient training of neural networks.

Another problem is the lack of an established metric for measuring the glare removal effect. In somewhat related problems metrics of image dehazing -SSIM (structural similarity measure), PSNR (peak signal to noise ratio) - and underwater photography -UIQM (underwater image quality metric), USIQUE (Underwater color image quality evaluation) and CCF (colorfulness, contrast and fog density)- have been used.

In this paper we specifically consider the problem of veiling glare removal. The contribution of the paper is threefold:

1. We propose a simple and fast algorithm for generation of images with synthetic veiling glare. Our method does not require the model of lens system.
2. We propose and train a novel CNN for veiling glare removal using only the synthetic data generated by the proposed algorithm. By validating the resulting network on real images, we prove that while the proposed data generation method does not create completely realistic images, it can be efficiently used for training CNNs for veiling glare removal.
3. We study several metrics used in the related tasks and analyze their applicability to estimating the result of veiling glare removal.

2. Related problems and quality metric

As shown above, there are not many papers considering the veil removal problem – so there are no established metrics, benchmarks or accepted baseline methods for it. There are, however, two relatively well-studied problems that are somewhat similar to veiling glare removal - image dehazing [18] and underwater image enhancement [19]. These problems have different causes (veiling flare is caused by light scattering inside the camera lens system; dehazing occurs due to light scattering on fog particles; under water, both wavelength-dependent absorption and scattering occur) – however, the intuitive solutions to these problems and the glare removal problem are similar and include: restoring image contrast, restoring color palette [46], enhancing details. In subsection 1.1, we analyze some metrics used in these problems and their applicability to the measurement of glare removal algorithms. In subsections 1.2 and 1.3 we applied some publicly available image dehazing and underwater image enhancement algorithms to images with veiling glare to test if they can be used "out of the box".

2.1. Metrics used in dehazing and underwater image enhancement

Metrics for various types of image enhancement can be separated into three categories [20]. Full-reference methods require a ground truth (ideal) image and estimate how similar the enhanced image and ground truth image are. Peak Signal-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) belong to this category. In contrast, no-reference methods require no ground truth image – such metrics are often used in underwater imaging due to the natural lack of ground truth reference – for example, UCIQE [21] – see the next paragraph for detailed overview. Finally, reduced-reference methods assume that some partial information about reference image is available. Usually this partial information is a statistical model of the reference image: for example, statistical model in the wavelet transform domain [22] or the distribution of coefficients of the discrete cosine transform [23].

Since full-reference metrics like PSNR [24] and SSIM [25] compare the image to the "ideal" one, their high values almost certainly indicate that the enhancement was successful. No-reference methods don't have such property – even then the metric is "good", the enhanced image can be noticeably non-ideal. However, obtaining the reference image can be tricky or near impossible, and in many problems no-reference or reduced-reference methods have to be used.

Let us consider a brief overview of the existing metrics. Denote the enhanced image as I_1 and the reference image as I_2.

PSNR (Peak signal to noise ratio)

PSNR is a full-reference metric, with larger PSNR corresponding to smaller distortion. It can be computed as

$$PNSR\left(I_1,I_2\right)=20\log_{10}\left(\frac{255}{\sqrt{MSE\left(I_1,I_2\right)}}\right),$$

$$MSE\left(I_1,I_2\right)=\frac{1}{whc}\sum\left(I_1-I_2\right)^2,$$

where w, h and c are width, height and number of channels of the image respectively.

SSIM (structural similarity)

Structural similarity of the two images takes values between 0 and 1 (higher values are better). It can be computed as:

$$SSIM\left(I_1,I_2\right)=\frac{\left(2\mu\left(I_1\right)\mu\left(I_2\right)+c_1\right)}{\left(\mu\left(I_1\right)^2+\mu\left(I_2\right)^2+c_1\right)}\times$$
$$\times\frac{\left(2\sigma\left(I_1,I_2\right)+c_2\right)}{\left(\sigma\left(I_1\right)^2+\sigma\left(I_2\right)^2+c_2\right)},$$

where $\mu\,(I_i)$ are averages across the images, $\sigma\,(I_i)$ are dispersions, c_1 and c_2 are constants.

We don't have the ground truth for real images (because it is really hard to gather both "glared" and "clear" images of the same scene), so we use these metrics only during training – since we have the ground truth image for our synthetic data.

Even without the reference image, a natural assumption is that removing the veiling glare will increase image's contrast and colors variety and increase image readability. Some underwater enhancement metrics are based on similar assumptions.

UCIQE (Underwater color image quality evaluation)

UCIQE – underwater color image quality evaluation [21] is simply a linear combination of chroma, saturation and contrast. Experiments [21] show that there is a strong correlation between this metric and the strength of spectral absorption and scattering of the water that affect the image. UCIQE is capable of measuring the non-uniform color cast, blurring and low-contrast in the images.

UCIQE can be calculate as:

$$USIQE(I_1)=c_1\cdot\sigma_c\left(I_1\right)+c_2\cdot con_l\left(I_1\right)+c_3\cdot\mu_s(I_1),$$

where $c_1=0.4680$, $c_2=0.2745$, $c_3=0.2576$ are constants taken from [21], σ_c is the standard deviation of chroma in the image, con_l is the contrast of luminance [26] and μ_s is the average saturation.

UIQM (Underwater Image Quality Measure)

Another metric, UIQM [27], comprises three underwater image attribute measures: the underwater image colorfulness measure (UICM), the underwater image sharpness measure (UISM), and the underwater image contrast measure (UIConM). Each of them evaluates one aspect of the underwater image degradation and is inspired by the properties of human visual systems (HVSs) [27]. According to [27], combining quality of each feature, this metric gives an estimation of image readability and legibility.

UIQM can be calculated as:

$$UIQM(I_1)=c_1\cdot UICM\left(I_1\right)+c_2\cdot UISM\left(I_1\right)+$$
$$+c_3\cdot UIConM(I_1),$$

where $c_1=0.282$, $c_2=0.2953$, $c_3=3.5753$ are empiric constants taken from [27], UICM, UISM and UIConM represent image's colorfulness, sharpness and contrast respectively. The formulas for calculating the UIQMare rather complex and can be found in [27]. In this paper we used the implementation of UIQM publicly available at [28].

CCF (Color, Contrast and Fog density)

CCF is constructed as a linear combination of colorfulness index, contrast index and fog density index, which can quantify the color loss caused by absorption, the blurring caused by forward scattering and the fogging caused by backward scattering, respectively.

CCF is calculated as:

$$CCF\left(I_1\right)=c_1\cdot Color\left(I_1\right)+c_2\cdot Contrast\left(I_1\right)+$$
$$+c_3\cdot FogDensity\left(I_1\right),$$

where $c_1=0.17593$, $c_2=0.61759$, $c_3=0.33988$ are taken from [29].

Colorfulness

$$Color\left(I_1\right)=\frac{\sqrt{\sigma_\alpha^2\left(I_1\right)+\sigma_\beta^2\left(I_1\right)}}{85.59}+$$
$$+\frac{0.3\sqrt{\mu_\alpha^2\left(I_1\right)+\mu_\beta^2\left(I_1\right)}}{85.59},$$

where σ represent variance and μ represent mean along each color axis (we include the required conversion into logarithmic scale into the equations):

$$\alpha=(\log R-\mu_R)-(\log G-\mu_G),$$
$$\beta=0.5((\log R-\mu_R)+(\log G-\mu_G))-(\log B-\mu_B).$$

Contrast

Image is divided into 64 by 64 blocks and then Sobel operator is applied to each block to decide whether the block has edges in it. After that, the contrast estimation is summed up:

$$Contrast\left(I_1\right)=\sum_{i=1}^{T}\sqrt{\frac{1}{MN}\sum_{i=1}^{N-1}\sum_{j=1}^{M-1}(I_{ij}-\overline{I}_1)^2},$$

where intensity I_{ij} is the i, j-th element of the two-dimensional block with size M by N (here $M=N=64$), T indicates the number of blocks with edges, and $\overline{I}$ is the average intensity of all pixel values in the block.

FogDensity

FogDensity is a complex data-driven metric, see [30] for details. It is calculated by generating Multivariate Gaussian models (MVG) on features extracted from 500 fog-free images and 500 fogged images. Given a test image I_1 and an instantiated fog- and fog- free- MVGs, two Mahalonobis distances are calculated: distance $D_{ff}(I_1)$ between MVG of test image and MVG model of fog-free images, then distance $D_f(I_1)$ between MVG of test image and MVG of fogged images.

Fog density is then calculated as:

$$D\left(I_1\right)=\frac{D_f\left(I_1\right)}{D_{ff}\left(I_1\right)+1}.$$

2.2. Applicability of dehazing algorithms to the removing veiling glare

We tried several public implementations of dehazing algorithms (Qin et al. [31], Chen et al. [32] and K. He et al. [33]) to test if they are effective in the veiling glare case. We came to the conclusion that they are capable of partly removing veiling glare when the glare color is close to white – probably because some of these algorithms often make an a priori assumption that the haze has white color. In contrast, not only the veiling glare can be colored, itscolor in particular image is not known a priori, as there are light sources with different spectrum.

Another difference between dehazing, underwater and glare removal is that the image degrading effect is distributed differently in these cases: fog effect usually depends on the depth of the pixels of the scene (for example, it can reach maximum near the horizon), while glare intensity depends more on the distance to the light source in the image coordinates.

Some examples of different images and results of public dehazing algorithms are shown in fig. 1 and table 2. Our algorithm does not perform well in dehazing either – so we conclude that these tasks need different solutions.

Table 2. Metrics values for different dehazing algorithms, computed on real images with glare

Metric	Original	Qin et al.	Chen et al.	K. He et al.
UCIQE	3.16	3.21	3.42	**4.12**
UIQM	0.49	0.52	0.58	**0.75**
CCF	1.73	1.80	3.33	**5.27**

2.3. Applicability of underwater image enhancement algorithms to glare removal

Rectifying underwater photography requires color [34]–[39], [43]–[44] and geometric [42] corrections, andwe have tested the latter to remove the veiling glare. Outputs of several underwater image enhancement (UIE) algorithms are presented in Fig. 2; some of them are capable of partially removing veiling glareand the resulting images are better than those produced with dehazing algorithms. From the generated images we judge that the two best methods are «Single Image Haze Removal Using Dark Channel Prior» [33] and«Image enhancement by histogram transformation» [35]. In the experimental section we compare their efficiency in removing the veiling glare with our algorithm.

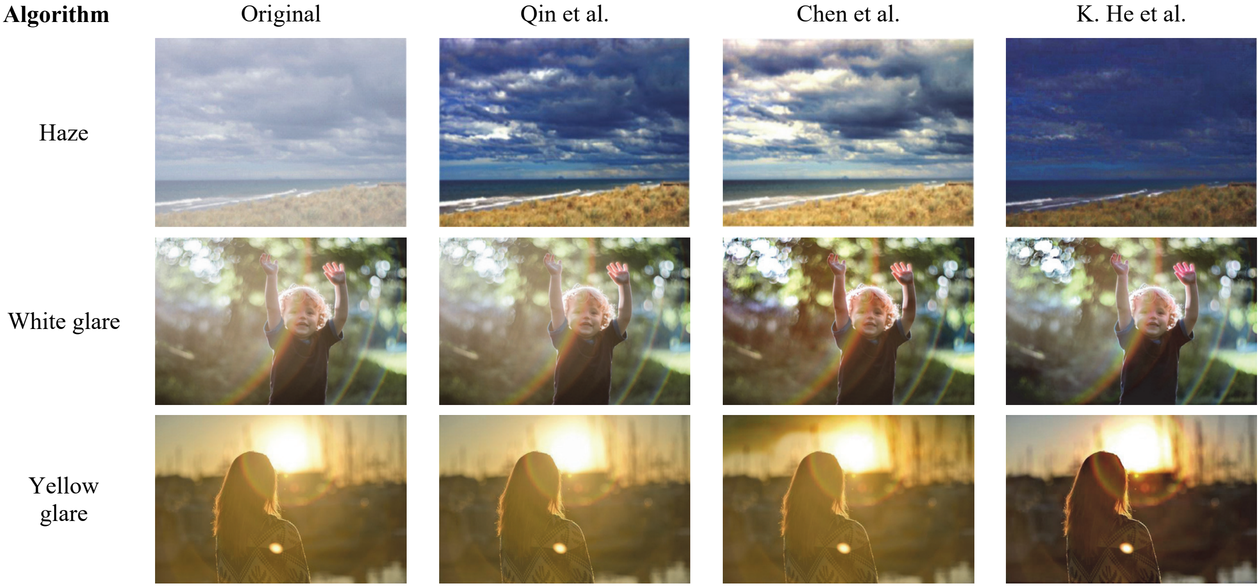

Fig. 1. Performance of different dehazing algorithms and our approach in images with haze, white and yellow glare

Following Table 3 presents metrics, obtained on our test glare images with different underwater image enhancement algorithms.

The list of considered algorithms and their abbreviations:

- CLAHE: Contrast limited adaptive histogram equalization [34].
- HE: Image enhancement by histogram transformation [35].
- RD: Underwater image quality enhancement through composition of dual-intensity images and Rayleigh-stretching [36].

Other considered algorithms, not presented in fig. 2:

- ICM: Underwater Image Enhancement Using an Integrated Colour Model [37].
- RGHS: Shallow-Water Image Enhancement Using Relative Global Histogram Stretching Based on Adaptive Parameter Acquisition [38].

UCM: Enhancing the low quality images using Unsupervised Colour Correction Method [39].

3. Data synthesis approach

Generating artificial data is a proven method used in training neural networks [4], [45]. To generate the training dataset, we used random images from COCO (found at *http://images.cocodataset.org/zips/ train2017.zip*) to generate "glared-ground truth" image pairs. The "glared" images are obtained from COCO images by adding a synthetic glare (see Algorithm 1). To obtain the ground truth training targets for CNN, original images also need to be modified – since the veiled images usually have a light source in them, and training on an image without light source would require the network to "inpaint" the light source – an effect that is not desirable and additionally complicates the network. These ground truth light sources are simply circles of (255, 255, 255) pixels.

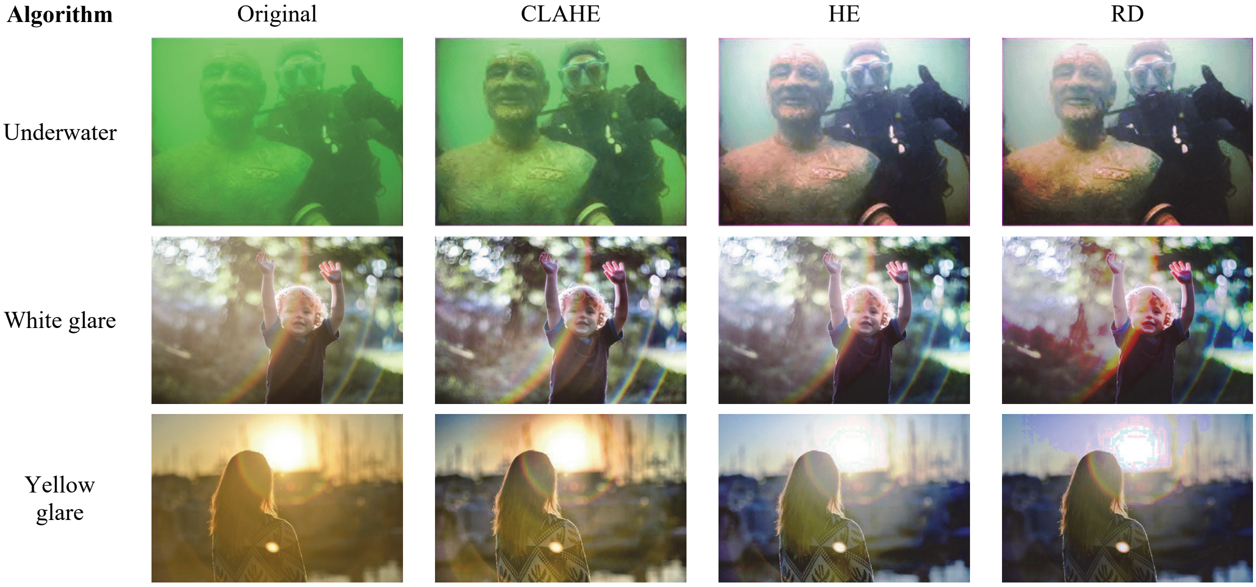

Fig. 2. Performance of different underwater image enhancement algorithms on underwater images and images with glare

Table 3. Metrics values for different UIE algorithms, computed on real images with glare

Metric	Original	CLAHE	HE	RD	ICM	RGHS	UCM
UCIQE	3.16	4.37	**4.50**	3.81	3.25	3.43	3.36
UIQM	0.49	0.73	**0.81**	0.51	0.57	0.71	0.55
CCF	1.73	3.25	3.40	3.94	2.79	2.92	2.83

We form each glared image as a superposition of original image and a synthetically generated glare. Let us denote the original image – free from glare – as ***I***, generated glare as *G* and the transmission mask (which governs the spatial distribution of glare) as *M*. Then the observed image with veiling glare, which is basically a result of spatial weighting [41], [46] of *G* and *M*, is denoted as *V* and formed as:

$$V = I \times M + G \times (1 - M),$$

where × stands for pixel-wise multiplication.

Let us consider the Algorithm 1 described below. From the clear image *I* with size (*H*, *W*, 3) (without a glare and without a strong light source) the algorithm generates (i) a synthetic ground truth image *GT* (an original image with added synthetic light source, but without glare; the intensities of this synthetic light source are (255, 255, 255)) and (ii) "veiled" output image *V* obtained by adding synthetic veiling glare to the *GT*. The examples of source and generated images are shown in fig. 4. On the left is the *GT* image (note the added "light source" – a white circle in the bottom right angle) and veiled image *V* is on the right.

3.1. Algorithm 1. Dataset synthesis

The algorithm consists of three parts. In the first one, we randomize the position of the light source and calculate the distance from it to each pixel of the image.

Table 4. Abbreviations and possible algorithm's parameters ranges

Abbrev.	Description	Range
LSR	Light source radius	Random uniform from (0.03, 0.1) *R*
GR	Glare radius	From LSR to *R*
GMC	Glare mask change	Determines how the flare fades based on the distance from light source. In our experiments has been chosen from range: (0.4, 1.6)
GCC	Glare color change	Determines how the color of the glare changes from its center to the periphery. In our experiments has been chosen from range: (0.2, 1.8)
MH	Mask high	The upper boundary on the value of mask. Has been set to 1
ML	Mask low	The lower boundary on the value of mask. In our experiments has been chosen from range: (0, 0.4)

Input: *I* – source image
Output: *V* – "veiled" image, *GT* – "ground truth" image.

Part 1: Distance matrix computation

1.1: Randomize the position of the light source center (C_x, C_y) within the image $I(x, y)$:

$$C_x = uniform(0, w),$$
$$C_y = uniform(0, h),$$

where function $uniform(a, b)$ returns a number uniformly distributed in a closed interval [*a*, *b*] and *w*, *h*are width and height of the image.

1.2: Compute distance *R* from the light source center to the farthest corner of the image (it represents maximum possible glare radius which can be generated).

1.3: Create two matrices $M_x(x, y) = x$ and $M_y(x, y) = y$: in M_x each column is filled with its number, in M_y each row is filled with its number.

1.4: Perform the following subtraction:

$$M_x(x,y) = M_x(x,y) - C_x,$$
$$M_y(x,y) = M_y(x,y) - C_y.$$

1.5: Compute distance matrix $D(x, y)$, representing distances from light source center, as following:

$$D = \sqrt{{M_x}^2 + {M_y}^2},$$

where rising to a power and square root are pixel-wise operations.

1.6: Randomize the values of Light Source Radius (LSR) and Glare Radius (GR):

$$LSR = uniform(0.03R,\ 0.1R),$$
$$GR = uniform(LSR,\ R).$$

1.7: Set each value in $D(x, y)$ less than $LSR$to LSR(this is the area close to the light source):

$$D(x,y) = max\big(LSR,\ D(x,y)\big).$$

1.8: Set each value in $D(x, y)$ larger than GR to GR (this area is not affected by the glare):

$$D(x,y) = min\big(GR,\ D(x,y)\big).$$

Part 2: Glare G synthesis

In the second part of the algorithm, we generate the glare that will be imposed on the original image.

2.1: Randomly choose the green g and blue b components of the glare:

$$g = uniform(0,255),$$
$$b = uniform(0,255).$$

2.2: The color of the glare changes from its center to its periphery; the «speed» of this change is parametrized by glare color change parameter – GCC, which is chosen randomly:

$$GCC = uniform(0.2,1.8).$$

2.3: Compute $G(x, y)$ using the following equation:

$$G(x,y) = \begin{pmatrix} 255 \\ 255 - \left(\dfrac{D(x,y) - LSR}{GR - LSR}\right)^{GCC} \cdot (255 - G) \\ 255 - \left(\dfrac{D(x,y) - LSR}{GR - LSR}\right)^{GCC} \cdot (255 - B) \end{pmatrix}.$$

Part 3: Mask M synthesis

In the third part of the algorithm, we generate the mask M, which controls how much each pixel of the image is affected by the glare.

3.1: Choose MH and ML – high and low boundaries of the possible values of the mask image:

$$MH = uniform(0.9,1),$$
$$ML = uniform(0,0.3).$$

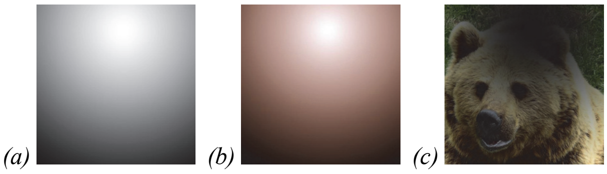

Fig. 3.Examples of generated mask (a), glare with applied mask (b) and source image with applied (1-mask) (c)

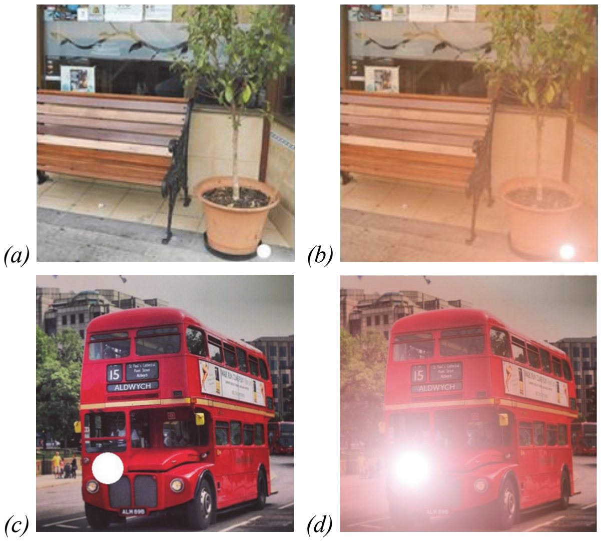

Fig. 4. Examples of synthesized GT: (a) and (c), and synthesized V: (b) and (d)

3.2: The highest mask value is achieved in the center of the light source (where the original image is completely "obscured" by the glare). Mask values decrease with the distance from that center, and the speed of decrease is controlled by glare mask change parameter (GMC) which is chosen as: $GMC = uniform\,(0.4, 1.6)$

3.3: The mask M it self is computed with:

$$M = MH - \left(\frac{D - LSR}{GR - LSR}\right)^{GMC} \cdot (MH - ML).$$

Part 4: Computation of resulting image V and ground truth image GT

4.1: Compute V as

$$V = I \times M + G \times (1 - M),$$

where × stands for pixel-wise multiplication.

4.2: Compute GT simply adding light source to the image I.

4. Neural network model

In our approach, we used UNet-like neural network [40], which doesn't change the image dimensions and is therefore convenient for image-to-image problems. Besides the architecture from the original UNet paper, we also tried a slightly deeper network; additionally, we split the network into two branches. This was done to test an assumption that image restoration effectively consists of two parts: (i) removing the glare and (i i) restoring the intensity of pixels of the source scene under the glare. Given input image V, our branched network outputs two images $Br1$ and $Br2$, and the restored image Out is then calculated as: $Out = V - Br1 + Br2$ where $Br1$ is "responsible" for glare removal and $Br2$ is "responsible" for restoring intensity of pixels of original image. The branched architecture is shown in fig. 5.

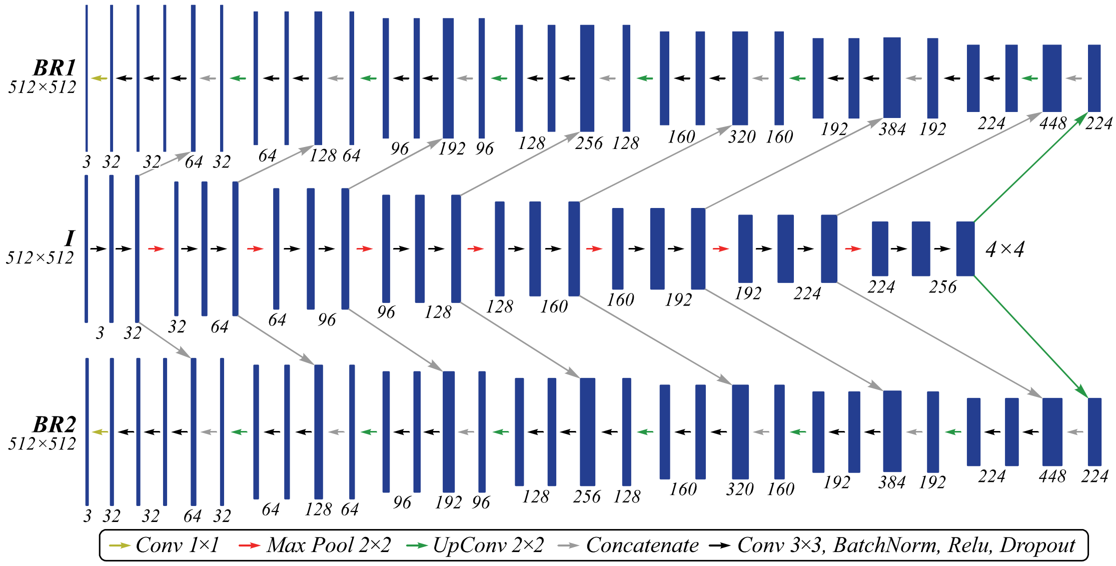

Fig. 5. Two-branched neural network architecture

Note, however, that our training loss didn't impose any incentive for these two branches to serve these exact functions – loss function operated with Out image, where $Br1$and $Br2$ parts of the output were mixed together. While, according to our experiments, the branched network outperformed the original one, its success can be also explained by the increased number of parameters. The details are provided below.

All convolution blocks in our papers consist of two 3 by 3 kernels. The number of filters in each layer linearly increases from 32 up to 256 or to 192 in the bottleneck (for models with depth 7 and 5 respectively) and goes back down to 32 over the deconvolutional part of the network. Each convolutional block also contains a batch normalization layer, dropout layer (drop rate linearly rises from 0.03 near the input up to 0.25 in the bottleneck, then symmetrically decreases in the deconvolutional part). ReLU are used as the activation functions. All maxpoolings and upconvolutions had 2 by 2 kernels.

Table 5. Data generator parameters used in all experiments in this section

Parameter name	*GR*	*GCC*	*GMC*	*ML*
Setup value	*R*	(1., 1.5)	(0.8, 1.2)	0

One-branched vs Two-branched network

Comparing UCIQE, UIQM and CCF for our real images test set,the best architecture is our two-branched network with depth equal to 7 – see table 6.

To test whether metrics improve because of this two-branched architecture (and not only due to the increased parameter count), we compared performance of two models with same number of parameters (about 1,4 million), one of them was two-branched and another one one-branched – but with increased width (channel count). table 7presents metrics on the test set. The results are not entirely conclusive, as UCIQE metric was slightly higher for one-branched model, however the advantage in CCF is much larger for two-branched model; additionally, visual inspection have shown that one-branched architecture created more artifacts on some test images near the light source (see fig. 6).

Table 6. Metrics on different architecture results

Architectur	CIQE	UIQM	CCF
Initial images metrics	3.16	0.49	1.73
Unet, depth equals 5	5.31	0.75	5.88
Unet, depth equals 7	5.71	0.80	6.01
Two-branched, depth = 5	5.62	0.79	5.97
Two-branched, depth = 7	**5.75**	**0.827**	**6.11**

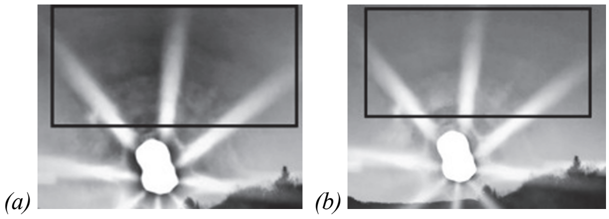

Fig. 6. Example of more distinguishable artifacts on one-branched model's test performance (a) in comparison with two-branched model (b)

Table 7. Metrics on models with different "branching", but same number of parameters

Architecture	UCIQE	UIQM	CCF
Two-branched	5.75	**0.83**	**6.11**
One-branched	5.77	0.80	6.03

Training regimen

Our network was trained on a relatively small dataset (14000 synthetic images), with Adam optimizer (with default parameters: $lr = 0.01$, $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 1e-8$) and MSE loss with batch size equal 4. Initial learning rate was set to 0.0001 and then was divided by 2 each time train loss decreased by less than 1 % for two epochs in a row on current *lr* (overall, the CNN trained for 40 epochs and learning rate typically equaled $1.25e-5$ in the last epoch).

The image of 512 by 512 pixels is processed in 77ms on Nvidia Tesla P100.

5. Experiments – choosing optimal dataset generation parameters

We found out that the degree of resulting changes in the image heavily depends on our data generation parameters. In this section we present some experiments aimed at finding the optimal parameters values of data generator.

Choosing appropriate metric

To test whether synthetic data is suitable for training, testing dataset should consist of real images.However, real images don't have the ground truth pairs – similar images without the glare. Therefore, no-reference metrics are required. In paragraph 1.1 we suggested to use underwater image enhancement metrics (UCIQE, UIQM and CCF); to test whether they are applicable, we used the synthetic (but held-out, not used in training – see Table 5 for parameter setup) part of the dataset and calculated both no-reference underwater metric and SSIM and PSNR. Then we compared the scores of images enhanced by various neural networks. According to table 8, the best network according to SSIM and PSNR was also the best according to UCIQE and CCF. This led us to the conclusion that underwater metricscan be used to estimate networks' glare removal quality on real images without a reference.

Table 8. PSNR, SSIM and UCIQE, UIQM and CCF on test part of the generated dataset

Metric	On *V*	On *Pred* (one branch, depth 5)	On *Pred* (one branch, depth 7)	On *Pred* (two branches, depth 5)	On *Pred* (two branches, depth 7)
PSNR (reference: *GT*)	10.7155	20.5070	21.8716	20.7210	**23.0192**
SSIM (reference: *GT*)	0.6235	0.8041	0.8242	0.8042	**0.8269**
UCIQE	2.1639	2.9895	3.1893	3.2494	**3.3545**
UIQM	0.0691	0.1069	**0.1131**	0.1012	0.1051
CCF	1.3226	3.4365	3.5899	3.6104	3.9216

Average PSNR and SSIM were computed both between *GT* and *V* images on the test part of our generated dataset and between *GT* and the prediction of neural network. These results show that the images obtained by the network are more similar to the original image (without the glare), when compared to the glared image.

5.1. Experiments setup

With fixed network architecture (two branches, depth equal to 7), we tuned dataset generation parameters and measured the results of a network trained on the resulting data. As shown below, changing parameters of data generation strongly affects the resulting network.

Final parameters of our method were chosen such that the network's metrics on the validation dataset (consisting of real images) were the highest. Parameters were optimized independently and sequentially (after finding the optimal value for the first parameter (given all others) we fixed it and tuned the second parameter – and so on. Initial parameters setup is shown it table 9 (intervals mean that algorithm performs random uniform choice of values from these ranges, and in the experiments we changed these ranges' limits). Abbreviation and parameters meanings are given in Section 2.

Table 9. Dataset generation parameters setup

Parameter	*GR*	*GCC*	*GMC*	*ML*
Value	*R*	(1., 1.5)	(0.8, 1.2)	0

5.2 Experiments with glare radius

First, we performed several experiments with **glare radius *GR***, changing the interval from which its value is sampled. table 10 shows that high *GR* values gave lower metrics – by visually analyzing the images, we concluded that it happened because some parts of the images became dark (color values become (0, 0, 0)). Smaller *GR* values don't cause image darkening, and network achieves high metrics values, except for the last experiment – too small *GR* value didn't "change" the images enough for the result to have high glare removal effect. The interval for *GR* wasset to (0.4, 1).

Table 10. Metrics for different GR intervals

GR interval	**UCIQE**	**UIQM**	**CCF**
Equals *R*	5.75	0.83	6.110
(1.0, 1.5)*R*	5.44	0.78	5.904
(0.7, 1) *R*	5.77	0.84	6.125
(0.4, 1) *R*	**5.79**	**0.845**	**6.129**
(0.1, 1) *R*	5.76	0.84	6.127
(0.1, 0.5) *R*	4.26	0.71	5.0321

5.3. Experiments with glare color change parameter

Next, we performed several experiments with **glare color change parameter *GCC*** – the parameter that determines how fast the color of the glare changes from its center to the periphery. The interval (0.5, 1) was optimal from the tested set, however, overall the difference between networks' performance was not a strong as in the *GR* experiments.

Table 11. Metrics for different GCC intervals

GCC interval	**UCIQE**	**UIQM**	**CCF**
(1., 1.5)	5.7931	0.8425	6.1293
(1., 1.8)	5.7348	0.8402	6.1032
(0.8, 1.2)	5.7989	0.8437	6.1313
(0.5, 1)	**5.8109**	**0.8463**	**6.1346**
(0.2. 0.8)	5.7345	0.8372	6.1243

5.4. Experiments with glare mask change parameter

Changing **glare mask change parameter *GMC*** almost didn't affect the resulting networks. Eventually we took (0.4, 1.) interval as optimal.

Table 12. Metrics for different GMC intervals

GMC interval	**UCIQE**	**UIQM**	**CCF**
(0.8, 1.2)	5.8109	0.8463	6.1346
(0.8, 1.6)	5.8113	0.8474	6.1371
(0.6, 1.2)	5.8165	0.8487	6.1363
(0.4, 1)	**5.8170**	**0.8483**	**6.1370**

5.5. Experiments with mask minimum value

We performed several experiments with *ML*. With larger values of *ML*, more glare is added to the image – as the result we get darker resulting images (same "darkening" effect that happened with high *GR* values). There is an intuitive explanation: with high *GR* and high *ML* values, all generated images are wholly covered by the glare; so, no portion of the dataset "shows" the CNN how to perform on a relatively «clear» parts of the images – which are present in the test data. This results in network darkening whole images.

Table 13. Metrics for different ML intervals

*ML*interval	**UCIQE**	**UIQM**	**CCF**
Equals 0	**5.82**	**0.848**	**6.137**
(0, 0.2)	5.75	0.842	6.129
(0, 0.3)	5.74	0.842	6.120
(0, 0.4)	5.73	0.839	6.119

Table 14 shows the final data generation parameters and metrics obtained by training on the resulting data: **UCIQE 5.817, UIQM 0.839,** and **CCF 6.137.**

Table 14. Optimal dataset synthesis parameters

Parameter	*GR*	*GCC*	*GMC*	*ML*
Value	(0.4, 1)*R*	(0.5, 1.)	(0.4, 1.)	0

6. Experiments – existing problems and improvingthe results

Experiments with SSIM with coefficients

One of the problems of resulting network was over-darkening the images (see fig. 7). To fight this problem, we added SSIM loss into the optimization function, which rationale that it will force the network to save more details of the original image – which is possible only by avoiding the over-darkening. The loss was implemented as follows:

$$Loss(y_{true}, y_{pred}) = MSE(y_{true}, y_{pred}) + \\ +C*(1 - SSIM(y_{true}, y_{pred})). \quad (1)$$

We conducted several experiments, varying SSIM weight *C*. For all of them was used the same dataset, generated with parameters, shown in table 15.

Table 15. Parameters set for SSIM experiments' dataset

Parameter name	*GR*	*GCC*	*GMC*	*ML*
Setup value	(1., 1.5)*R*	(1., 1.5)	(0.8, 1.2)	0

As seen from Table 16, best value of *C* (0.25) noticeably improved the metrics. With this value of C, SSIM loss term is roughly equal to the MSE loss term. fig. 7 show noticeable improvement of the darkened area.

Table 16. Metrics valuRes obtained with different C coefficients

C	**UCIQE**	**UIQM**	**CCF**
0	5.4367	0.7798	5.9038
0.1	5.4514	0.7692	5.9321
0.25	**5.5930**	**0.8153**	**6.1052**
0.4	5.4378	0.7773	5.8619
1	5.4280	0.7720	5.8516

Final approach metrics and alternative algorithms

Below we present a network trained on dataset with optimal parameters (table 14), with loss, constructed from MSE and SSIM ($C = 0.25$). There are no available veiling glare removal algorithms, so we conducted comparison with algorithms which originally solved similar problems dehazing and underwater image enhancement and have shown best performance (see fig. 1 and 2). As seen from the following table, we outperform both alternative methods, although in some images other algorithms sometimes look better than ours (see fig. 8 for example).

Table 17. Best metrics, obtained with optimal parameters and combined loss, and metrics, obtained with dehazing and underwater image enhancement algorithms

Experiment	**UCIQE**	**UIQM**	**CCF**
Proposed method	**6.01**	**0.84**	**6.20**
K. He et al. [33] (Dehazing)	4.12	0.75	5.27
R. Hummel[35] (UIE)	4.50	0.81	3.40

Conclusion

In this paper we considered the problem of veiling glare removal. Firstly, we have found that no task-specific metrics, no-reference methods have been established; we proposed to use metrics originally used in underwater image enhancement and shown that improvement in these metrics correlates with improvement with reference-based metrics. Secondly, to solve the problem of absent datasets and the complexity of obtaining the ground truth images, we proposed a simple algorithm for generation of synthetic veiling glare images.

Thirdly, we proposed a two-branch UNet-like neural network architecture and its training regimen for glare removal and shown the efficiency of adding SSIM term into the training objective. While there are no specific benchmarks or established methods for glare removal, our method outperforms algorithms originally developed for similar tasks of dehazing and underwater image enhancement.

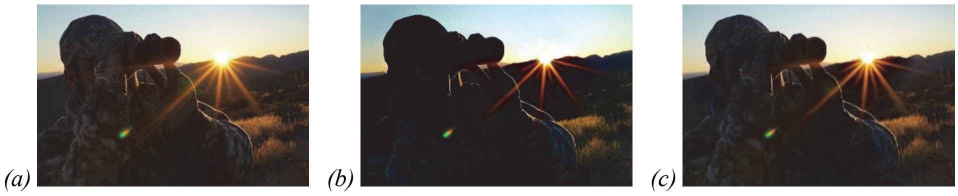

Fig. 7. Example of overdarkened image: source image (a), overdarkened image (b) and improved image (c)

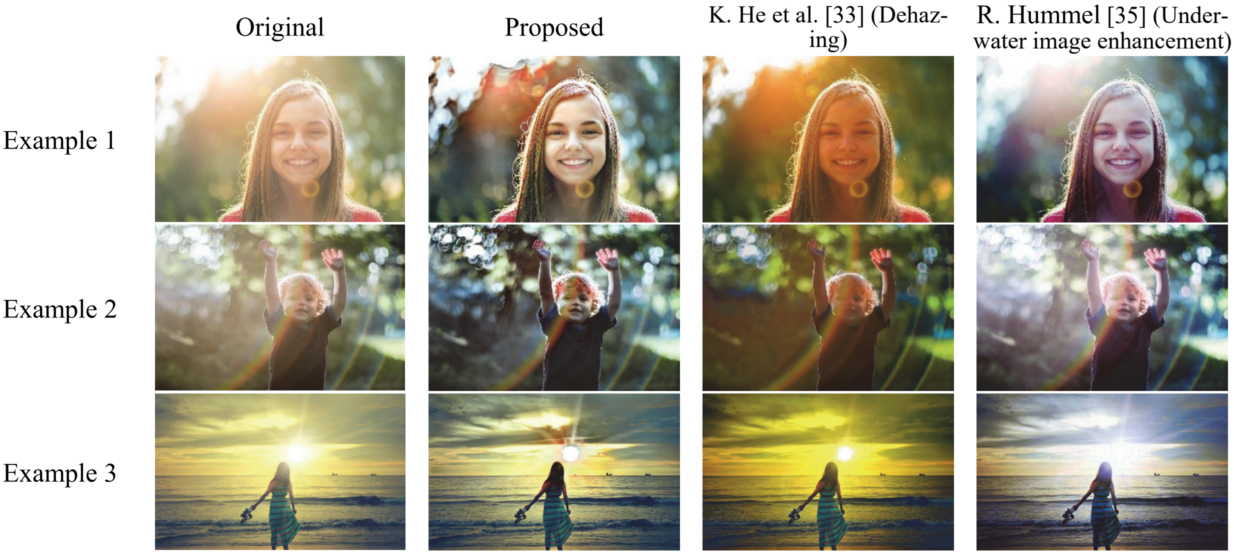
Original
Proposed
K. He et al. [33] (Dehazing)
R. Hummel [35] (Underwater image enhancement)
Example 1
Example 2
Example 3

Fig. 8. Examples of performance of proposed, dehazing and UIE algorithms

References

[1] Lange H. Automatic glare removal in reflectance imagery of the uterine cervix. Proc SPIE 2005; 5747: 2183-2192. DOI: 10.1117/12.596012.

[2] Lamprinou N, Psarakis E. Fast detection and removal of glare in gray scale laparoscopic images. Proc Int Conf on Computer Vision Theory and Applications 2018: 206-212. DOI: 10.5220/0006654202060212.

[3] Ye S, Yin J, Chen B-H, Chen D, Wu Y. Single image glare removal using deep convolutional networks. Proc IEEE Int Conf on Image Processing (ICIP) 2020: 201-205. DOI: 10.1109/ICIP40778.2020.9190712.

[4] Sandhan T, Choi J. Anti-Glare: Tightly constrained optimization for eyeglass reflection removal. Proc IEEE Conf on Computer Vision and Pattern Recognition (CVPR) 2017: 1675-1684. DOI: 10.1109/CVPR.2017.182.

[5] Hara T, Saito H, Kanade T. Removal of glare caused by water droplets. The Journal of the Institute of Image Information and Television Engineers 2009: 64. DOI: 10.1109/CVMP.2009.17.

[6] Safranek S. A comparison of techniques used for the removal of lens flare found in high dynamic range luminance measurements [Thesis]. Boulder, CO: University of Colorado; 2017.

[7] Cozzi F, Elia C, Gerosa G, Rocchetta F, Lanaro MP, Rizzi A. Use of an occlusion mask for veiling glare removal in HDR images. J Imaging 2018; 4: 100. DOI: 10.3390/jimaging4080100.

[8] Dušan P. Removing lens flare from digital photographs. Diploma thesis at Charles University in Prague Faculty of Mathematics and Physics. Prague: 2009.

[9] McCann J, Rizzi A. Veiling glare: The dynamic range limit of HDR images. Proc SPIE 2007; 6492: 649213. DOI: 10.1117/12.703042.

[10] McCann J, Rizzi A. Camera and visual veiling glare in HDR images. J Soc Inf Disp 2007; 15(9): 721-730. DOI: 10.1889/1.2785205.

[11] Talvala E-V, Adams A, Horowitz M, Levoy M. Veiling glare in high dynamic range imaging. ACM Trans Graph 2007; 26(3): 37. DOI: 10.1145/1275808.1276424.

[12] Boynton P, Kelley E. Liquid-filled camera for the measurement of high-contrast images. Proc SPIE 2003; 5080: 370-378. DOI: 10.1117/12.519602.

[13] Howorth JR. Anti-veiling-glare glass input window for an optical device and method for manufacturing such window. US Pat US4760307A of July 26, 1988.

[14] Gong D, Zhang Z, Shi Q, van den Hengel A, Shen C, Zhang Y. Learning deep gradient descent optimization for image deconvolution. IEEE Trans Neural Netw Learn Syst 2020; 31(12): 5468-5482. DOI: 10.1109/TNNLS.2020.2968289.

[15] Keshmirian A. A physically-based approach for lens flare simulation [Thesis]. San Diego: University of California; 2008.

[16] Hullin M, Eisemann E, Seidel H-P, Lee S. Physically-based real-time lens flare rendering. ACM Trans Graph 2011; 30: 108. DOI: 10.1145/2010324.1965003.

[17] Kilgard MJ. Fast OpenGL-rendering of lens flares. Source: ⟨https://www.opengl.org/archives/resources/features/KilgardTechniques/LensFlare/⟩.

[18] Zhang Z, Feng H, Xu Z, Li Q, Chen Y. Single image veiling glare removal. J Mod Opt 2018; 65(19): 2220-2230. DOI: 10.1080/09500340.2018.1506057.

[19] Li C, Guo J, Cong R, Pang Y, Wang B. Underwater image enhancement by dehazing with minimum information loss and histogram distribution prior. IEEE Trans Image Process 2016; 25(12): 5664-5677. DOI: 10.1109/TIP.2016.2612882.

[20] Wang W, Yuan X. Recent advances in image dehazing. IEEE/CAA Journal of Automatica Sinica 2017; 4(3): 410-436. DOI: 10.1109/JAS.2017.7510532.

[21] Yang M, Sowmya A. An underwater color image quality evaluation metric. IEEE Trans Image Process 2015; 24(12): 6062-6071. DOI: 10.1109/TIP.2015.2491020.

[22] Wang Z, Simoncelli E. Reduce-reference image quality assessment using a wavelet-domain natural image statistic model. Proc SPIE 2005; 5666: 149-159. DOI: 10.1117/12.597306.

[23] Ma L, Li S, Zhang F, Ngan K. Reduced-reference image quality assessment using reorganized DCT-based image representation. IEEE Trans Multimedia 2011; 13(4): 824-829. DOI: 10.1109/TMM.2011.2109701.

[24] Li ZG, Zheng J, Yao W, Zhu Z. Single image haze removal via a simplified dark channel. Proc IEEE Int Conf on Acoustics, Speech and Signal Processing (ICASSP) 2015: 1608-1612. DOI: 10.1109/ICASSP.2015.7178242.

[25] Fang Y, Ma K, Wang Z, Lin W, Fang Z, Zhai G. No-reference quality assessment of contrast-distorted images based on natural scene statistics. IEEE Signal Process Lett 2015; 22(7): 838-842. DOI: 10.1109/LSP.2014.2372333.

[26] Macintyre B, Cowan W. A practical approach to calculating luminance contrast on a CRT. ACM Trans Graph 2000; 11(4): 336-347. DOI: 10.1145/146443.146467.

[27] Panetta K, Gao C, Agaian S. Human-visual-system-inspired underwater image quality measures. IEEE J Ocean Eng 2015; 41(3): 541-551. DOI: 10.1109/JOE.2015.2469915.

[28] Publicly available implementation of UIQM. Source: ⟨https://github.com/xahidbuffon/SRDRM/blob/master/utils/uqim_utils.py⟩.

[29] Wang Y, Li N, Li Z, Gu Z, Zheng H, Zheng B, Sun M. An imaging-inspired no-reference underwater color image quality assessment metric. Comput Electr Eng 2017; 70: 904-913. DOI: 10.1016/j.compeleceng.2017.12.006.

[30] Choi LK, You J, Bovik AC. Referenceless prediction of perceptual fog density and perceptual image defogging. IEEE Trans Image Process 2015; 24(11): 3888-3901. DOI: 10.1109/TIP.2015.2456502.

[31] Qin X, Wang Z, Bai Y, Xie X, Jia H. FFA-Net: Feature fusion attention network for single image dehazing. Proceedings of the AAAI Conference on Artificial Intelligence 2020; 34(7). 11908-11915. DOI: 10.1609/aaai.v34i07.6865.

[32] Chen W-T, Ding J-J, Kuo S-Y. PMS-Net: Robust haze removal based on patch map for single images. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition (CVPR) 2019: 11673-11681. DOI: 10.1109/CVPR.2019.01195.

[33] He K, Sun J, Tang X. Single image haze removal using dark channel prior. IEEE Trans Pattern Anal Mach Intell 2011; 33. 2341-2353. DOI: 10.1109/CVPRW.2009.5206515.

[34] Contrast limited adaptive histogram equalization. contrast limited adaptive histogram equalization. Graphics Gems IV 1994: 474-485. DOI: 10.1016/B978-0-12-336156-1.50061-6.

[35] Hummel R. Image enhancement by histogram transformation. Comput Gr Image Process 1977; 6: 184-195. DOI: 10.1016/S0146-664X(77)80011-7.

[36] Ghani ASA, Isa NAM. Underwater image quality enhancement through composition of dual-intensity images and Rayleigh-stretching. SpringerPlus 2014; 3: 757. DOI: 10.1186/2193-1801-3-757.

[37] Iqbal K, Salam RA, Azam O, Talib A. Underwater image enhancement using an integrated colour model. IAENG Int J Comput Sci 2007; 2: 239-244.

[38] Huang D, Wang Y, Song W, Sequeira J, Mavromatis S. Shallow-water image enhancement using relative global histogram stretching based on adaptive parameter acquisition. In Book: Schoeffmann K, Chalidabhongse TH, Ngo CW, Aramvith S, O'Connor NE, Ho Y-S, Gabbouj M, Elgammal A, eds. MultiMedia modeling 2018: 453-465. DOI: 10.1007/978-3-319-73603-7_37.

[39] Iqbal K, Odetayo M, James A, Salam RA, Talib A. Enhancing the low quality images using Unsupervised Colour Correction Method. Proc IEEE Int Conf on Systems, Man and Cybernetics 2010; 1703-1709. DOI: 10.1109/ICSMC.2010.5642311.

[40] Ronneberger O, Fischer P, Brox T. U-Net: Convolutional networks for biomedical image segmentation. In Book: Navab N, Hornegger J, Wells WM, Frangi AF, eds. Medical image computing and computer-assisted intervention – MICCAI 2015 2015: 234-241. DOI: 10.1007/978-3-319-24574-4_28.

[41] Ilyukhin SA, Chernov TS, Polevoy DV. Improving the accuracy of neural network methods of verification of persons by spatialweighted normalization of brightness image [In Russian]. Informatsionnye Tehnologii I Vichislitelnye Systemi 2019; 4: 12-20. DOI: 10.14357/20718632190402.

[42] Senshina DD, Glikin AA, Polevoy DV, Kunina IA, Ershov EI, Smagina AA. Radial distortion correction for camera submerged under water [In Russian]. Sensory Systems 2020; 34(3): 254-264. DOI: 10.31857/S0235009220030087.

[43] Shepelev DA, Bozhkova VP, Ershov EI, Nikolaev DP. Simulating shot noise of color underwater images. Computer Optics 2020; 44(4): 671-679. DOI: 10.18287/2412-6179-CO-754.

[44] Shepelev DA. Color reproduction accuracy in channel-wise simulation of underwater images [In Russian]. Information Processes 2020; 20(3): 254-268.

[45] Gayer AV, Chernyshova YS, Sheshkus AV. Artificial training data generation for the task of character recognition of fields of russian passport [In Russian]. Sensory Systems 2018; 32(3): 230-235. DOI: 10.1134/S023500921803006X.

[46] Ilyuhin SA, Chernov TS, Polevoy DV, Fedorenko FA. A method for spatially weighted image brightness normalization for face verification. Proc SPIE 2019; 11041: 1104118. DOI: 10.1117/12.2522922.

[47] Polevoy DV, Panfilova EI, Ershov EI, Nikolaev DP. Color correction of the document owner's photograph image during recognition on mobile device. Proc SPIE 2021; 11605: 1160510. DOI: 10.1117/12.2587627.

Authors' information

Alexey Valeryevich Shoshin, (b. 1999) is currently studying at MIPT. Research interests include image processing, deep learning. E-mail: *shoshin.av@phystech.edu* .

Evgeny Alexandrovich Shvets, (b. 1990) graduated from Moscow Institute for Physics and Technology and received his PhD in 2017 in Institute for Information Transmission Problems. Research interests include image processing, deep learning and image registration. E-mail: *shvets@visillect.com* .

Code of State Categories Scientific and Technical Information (in Russian – GRNTI)): 28.23.15, 28.23.37, 20.19.29.
Received February 26, 2021. The final version – April 28, 2021.