DeepWarp: Photorealistic Image Resynthesis for Gaze Manipulation

Yaroslav Ganin, Daniil Kononenko, Diana Sungatullina, Victor Lempitsky

Skolkovo Institute of Science and Technology,
{ganin,daniil.kononenko,d.sungatullina,lempitsky}@skoltech.ru

Abstract. In this work, we consider the task of generating highly-realistic images of a given face with a redirected gaze. We treat this problem as a specific instance of conditional image generation and suggest a new deep architecture that can handle this task very well as revealed by numerical comparison with prior art and a user study. Our deep architecture performs coarse-to-fine warping with an additional intensity correction of individual pixels. All these operations are performed in a feed-forward manner, and the parameters associated with different operations are learned jointly in the end-to-end fashion. After learning, the resulting neural network can synthesize images with manipulated gaze, while the redirection angle can be selected arbitrarily from a certain range and provided as an input to the network.

Keywords: gaze correction, warping, spatial transformers, deep learning

1 Introduction

In this work, we consider the task of learning deep architectures that can transform input images into new images in a certain way (deep image resynthesis). Generally, using deep architectures for image generation has become a very active topic of research. While a lot of very interesting results have been reported over recent years and even months, achieving photo-realism beyond the task of synthesizing small patches has proven hard.

Previously proposed methods for deep resynthesis usually tackle the resynthesis problem in a general form and strive for universality. Here, we take an opposite approach and focus on a very specific image resynthesis problem (gaze manipulation) that has a long history in the computer vision community [20,26,27,1,13,24,18,7,16] and some important real-life applications. We show that by restricting the scope of the method and exploiting the specifics of the task, we are indeed able to train deep architectures that handle gaze manipulation well and can synthesize output images of high realism (Figure 1).

Generally, few image parts can have such a dramatic effect on the perception of an image like regions depicting eyes of a person in this image. Humans (and even non-humans [23]) can infer a lot of information about of the owner of the eyes, her intent, her mood, and the world around her, from the appearance of

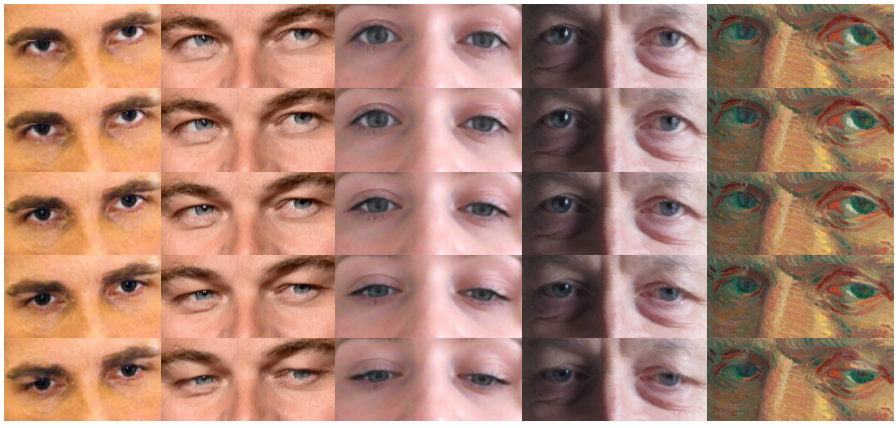


Fig. 1. Gaze redirection with our model trained for vertical gaze redirection. The model takes an input image (middle row) and the desired redirection angle (here varying between -15 and +15 degrees) and re-synthesize the new image with the new gaze direction. Note the preservation of fine details including specular highlights in the resynthesized images.

the eyes and, in particular, from the direction of the gaze. Generally, the role of gaze in human communication is long known to be very high [15].

In some important scenarios, there is a need to digitally alter the appearance of eyes in a way that changes the apparent direction of the gaze. These scenarios include gaze correction in video-conferencing, as the intent and the attitude of a person engaged in a videochat is distorted by the displacement between the face on her screen and the webcamera (e.g. while the intent might be to gaze into the eyes of the other person, the apparent gaze direction in a transmitted frame will be downwards). Another common scenario that needs gaze redirection is "talking head"-type videos, where a speaker reads the text appearing alongside the camera but it is desirable to redirect her gaze into the camera. One more example includes editing of photos (e.g. group photos) and movies (e.g. during postproduction) in order to make gaze direction consistent with the ideas of the photographer or the movie director.

All of these scenarios put very high demands on the realism of the result of the digital alteration, and some of them also require real-time or near real-time operation. To meet these challenges, we develop a new deep feed-forward architecture that combines several principles of operation (coarse-to-fine processing, image warping, intensity correction). The architecture is trained end-to-end in a supervised way using a specially collected dataset that depicts the change of the appearance under gaze redirection in real life.

Qualitative and quantitative evaluation demonstrate that our deep architecture can synthesize very high-quality eye images, as required by the nature of the applications, and does so at several frames per second. Compared to several recent methods for deep image synthesis, the output of our method contains

larger amount of fine details (comparable to the amount in the input image). The quality of the results also compares favorably with the results of a random forest-based gaze redirection method [16]. Our approach has thus both practical importance in the application scenarios outlined above, and also contributes to an actively-developing field of image generation with deep models.

2 Related work

Deep learning and image synthesis. Image synthesis using neural networks is receiving growing attention [19,3,8,2,5,9]. More related to our work are methods that learn to transform input images in certain ways [17,6,22]. These methods proceed by learning internal compact representations of images using encoder-decoder (autoencoder) architectures, and then transforming images by changing their internal representation in a certain way that can be trained from examples. We have conducted numerous experiments following this approach combining standard autoencoders with several ideas that have reported to improve the result (convolutional and up-convolutional layers [28,3], adversarial loss [8], variational autoencoders [14]). However, despite our efforts (see the Appendix), we have found that for large enough image resolution, the outputs of the network lacked high-frequency details and were biased towards typical mean of the training data ("regression-to-mean" effect). This is consistent with the results demonstrated in [17,6,22] that also exhibit noticeable blurring.

Compared to [17,6,22], our approach can learn to perform a restricted set of image transformations. However, the perceptual quality and, in particular, the amount of high-frequency details is considerably better in the case of our method due to the fact that we deliberately avoid any input data compression within the processing pipeline. This is crucial for the class of applications that we consider.

Finally, the idea of spatial warping that lies in the core of the proposed system has been previously suggested in [12]. In relation to [12], parts of our architecture can be seen as spatial transformers with the localization network directly predicting a sampling grid instead of low-dimensional transformation parameters.

Gaze manipulation. An early work on monocular gaze manipulation [24] did not use machine learning, but relied on pre-recording a number of potential eye replacements to be copy-pasted at test time. The idea of gaze redirection using supervised learning was suggested in [16], which also used warping fields that in their case were predicted by machine learning. Compared to their method, we use deep convolutional network as a predictor, which allows us to achieve better result quality. Furthermore, while random forests in [16] are trained for a specific angle of gaze redirection, our architecture allows the redirection angle to be specified as an input, and to change continuously in a certain range. Most practical applications discussed above require such flexibility. Finally, the realism

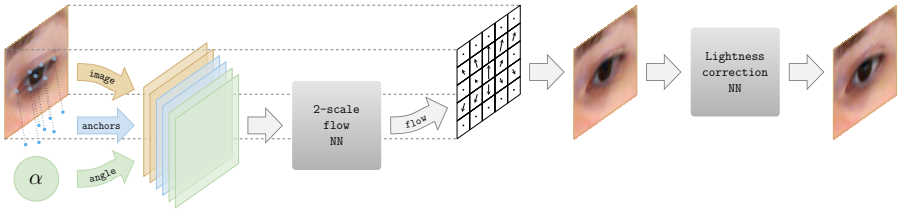


Fig. 2. The **proposed system** takes an input eye region, feature points (`anchors`) as well as a correction `angle` α and sends them to the multi-scale neural network (see Section 3.2) predicting a `flow` field. The flow field is then applied to the input image to produce an image of a redirected eye. Finally, the output is enhanced by processing with the lightness correction neural network (see Section 3.4).

of our results is boosted by the lightness adjustment module, which has no counterpart in the approach of [16].

Less related to our approach are methods that aim to solve the gaze problem in videoconferencing via synthesizing 3D rotated views of either the entire scene [20,1,26] or of the face (that is subsequently blended into the unrotated head) [18,7]. Out of this works only [7] works in a monocular setting without relying on extra imaging hardware. The general problem with the novel view synthesis is how to fill disoccluded regions. In cases when the 3D rotated face is blended into the image of the unrotated head [18,7], there is also a danger of distorting head proportions characteristic to a person.

3 The model

In this section, we discuss the architecture of our deep model for re-synthesis. The model is trained on pairs of images corresponding to eye appearance before and after the redirection. The redirection angle serves as an additional input parameter that is provided both during training and at test time.

As in [16], the bulk of gaze redirection is accomplished via warping the input image (Figure 2). The task of the network is therefore the prediction of the warping field. This field is predicted in two stages in a coarse-to-fine manner, where the decisions at the fine scale are being informed by the result of the coarse stage. Beyond coarse-to-fine warping, the photorealism of the result is improved by performing pixel-wise correction of the brightness where the amount of correction is again predicted by the network. All operations outlined above are implemented in a single feed-forward architecture and are trained jointly end-to-end.

We now provide more details on each stages of the procedure, starting with more detailed description of the data used to train the architecture.

3.1 Data preparation

At training time, our dataset allows us to mine pairs of images containing eyes of the same person looking in two different directions separated by a known angle α. The head pose, the lighting, and all other nuisance parameters are (approximately) the same between the two images in the pair. Following [16] (with some modifications), we extract the image parts around each of the eye and resize them to characteristic scale. For simplicity of explanation, let us assume that we need to handle left eyes only (the right eyes can be handled at training and at test times via mirroring).

To perform the extraction, we employ an external face alignment library [25] producing, among other things, $N = 7$ feature points $\{(x_i^{\text{anchor}}, y_i^{\text{anchor}}) \,|\, i = 1, \ldots, N\}$ for the eye (six points along the edge and also the pupil center). Next, we compute a tight axis-aligned bounding box $\mathcal{B}'$ of the points in the *input* image. We enlarge $\mathcal{B}'$ to the final bounding-box $\mathcal{B}$ using a characteristic radius R that equals the distance between the corners of an eye. The size of $\mathcal{B}$ is set to $0.8R \times 1.0R$. We then cut out the interior of the estimated box from the input image, and also from the output image of the pair (using exactly the same bounding box coordinates). Both images are then rescaled to a fixed size ($W \times H = 51 \times 41$ in our experiments). The resulting image pair serves as a training example for the learning procedure (Figure 4-Right).

3.2 Warping modules

Each of the two warping modules takes as an input the image, the position of the feature points, and the redirection angle. All inputs are expressed as maps as discussed below, and the architecture of the warping modules is thus "fully-convolutional", including several convolutional layers interleaved with Batch Normalization layers [11] and ReLU non-linearities (the actual configuration is shown in the Appendix). To preserve the resolution of the input image, we use 'same'-mode convolutions (with zero padding), set all strides to one, and avoid using max-pooling.

Coarse warping. The last convolutional layer of the first (half-scale) warping module produces a pixel-flow field (a two-channel map), which is then upsampled $\mathbf{D}_{\text{coarse}}(I, \alpha)$ and applied to warp the input image by means of a bilinear sampler $\mathbf{S}$ [12,21] that finds the *coarse estimate*:

$$O_{\text{coarse}} = \mathbf{S}\left(I, \mathbf{D}_{\text{coarse}}(I, \alpha)\right) . \tag{1}$$

Here, the sampling procedure S samples the pixels of O_{coarse} at pixels determined by the flow field:

$$O_{\text{coarse}}(x, y, c) = I\{x + \mathbf{D}_{\text{coarse}}(I, \alpha)(x, y, 1), y + \mathbf{D}_{\text{coarse}}(I, \alpha)(x, y, 2), c\} , \tag{2}$$

where c corresponds to a color channel (R,G, or B), and the curly brackets correspond to bilinear interpolation of $I(\cdot, \cdot, c)$ at a real-valued position. The sampling procedure (1) is piecewise differentiable [12].

Fine warping. In the fine warping module, the rough image estimate O_{coarse} and the upsampled low-resolution flow $\mathbf{D}_{\text{coarse}}(I, \alpha)$ are concatenated with the input data (the image, the angle encoding, and the feature point encoding) at the original scale and sent to the $1\times$-scale network which predicts another two-channel flow $\mathbf{D}_{\text{res}}$ that amends the half-scale pixel-flow (additively [10]):

$$\mathbf{D}(I, \alpha) = \mathbf{D}_{\text{coarse}}(I, \alpha) + \mathbf{D}_{\text{res}}(I, \alpha, O_{\text{coarse}}, \mathbf{D}_{\text{coarse}}(I, \alpha)) \,, \tag{3}$$

The amended flow is used to obtain the final output (again, via bilinear sampler):

$$O = \mathbf{S}\left(I, \mathbf{D}(I, \alpha)\right) \,. \tag{4}$$

The purpose of coarse-to-fine processing is two-fold. The half-scale (coarse) module effectively increases the receptive field of the model resulting in a flow that moves larger structures in a more coherent way. Secondly, the coarse module gives a rough estimate of how a redirected eye would look like. This is useful for locating problematic regions which can only be fixed by a neural network operating at a finer scale.

3.3 Input encoding

As discussed above, alongside the raw input image, the warping modules also receive the information about the desired redirection angle and feature points also encoded as image-sized feature maps.

Embedding the angle. Similarly to [6], we treat the correction angle as an attribute and embed it into a higher dimensional space using a multi-layer perceptron $\mathbf{F}_{\text{angle}}(\alpha)$ with ReLU non-linearities. The precise architecture is `FC(16)` $\rightarrow$ `ReLU` $\rightarrow$ `FC(16)` $\rightarrow$ `ReLU`. Unlike [6], we do not output separate features for each spatial location but rather opt for a single position-independent 16-dimensional vector. The vector is then expressed as 16 constant maps that are concatenated into the input map stack. During learning, the embedding of the angle parameter is also updated by backpropagation.

Embedding the feature points. Although in theory a convolutional neural network of an appropriate architecture should be able to extract necessary features from the raw input pixels, we found it beneficial to further augment 3 color channels with additional 14 feature maps containing information about the eye anchor points.

In order to get the anchor maps, for each previously obtained feature point located at $(x_i^{\text{anchor}}, y_i^{\text{anchor}})$, we compute a pair of maps:

$$\begin{aligned} \Delta_x^i[x, y] &= x - x_i^{\text{anchor}} \,, \\ \Delta_y^i[x, y] &= y - y_i^{\text{anchor}} \,, \end{aligned} \quad \forall (x, y) \in \{0, \ldots, W\} \times \{0, \ldots, H\} \,, \tag{5}$$

where W, H are width and height of the input image respectively. The embedding give the network "local" access to similar features as used by decision trees in [16].

Ultimately, the input map stack consists of 33 maps (RGB + 16 angle embedding maps + 14 feature point embedding maps).

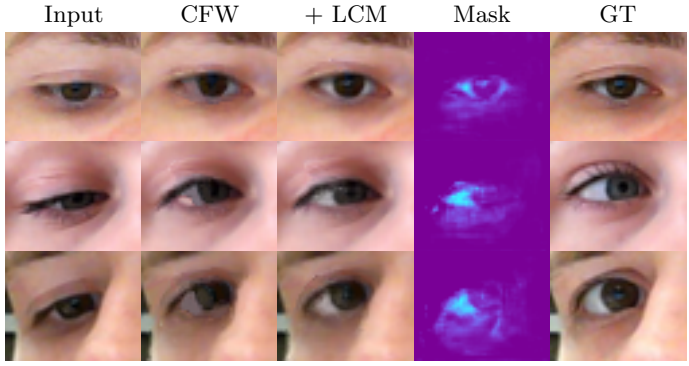


Fig. 3. Visualization of three challenging redirection cases where **Lightness Correction Module** helps considerably compared to the system based solely on coarse-to-fine warping (CFW) which is having difficulties with expanding the area to the left of the iris. The 'Mask' column shows the soft mask corresponding to parts where lightness is increased. Lightness correction fixes problems with inpainting disoccluded eye-white, and what is more emphasizes the specular highlight increasing the perceived realism of the result.

3.4 Lightness Correction Module

While the bulk of appearance changes associated with gaze redirection can be modeled using warping, some subtle but important transformations are more photometric than geometric in nature and require a more general transformation. In addition, the warping approach can struggle to fill in disoccluded areas in some cases.

To increase the generality of the transformation that can be handled by our architecture, we add the final lightness adjustment module (see Figure 2). The module takes as input the features computed within the coarse warping and fine warping modules (specifically, the activations of the third convolutional layer), as well as the image produced by the fine warping module. The output of the module is a single map M of the same size as the output image that is used to modify the brightness of the output O using a simple element-wise transform:

$$O_{\text{final}}(x, y, c) = O(x, y, c) \cdot (1 - M(x, y)) + M(x, y)\,, \tag{6}$$

assuming that the brightness in each channel is encoded between zero and one. The resulting pixel colors can thus be regarded as blends between the colors of the warped pixels and the white color. The actual architecture for the lightness correction module in our experiments is shown in the Appendix.

This idea can be, of course, generalized further to a larger number of colors in the *palette* for admixing, while these colors can be defined either manually or made dataset-dependent or even image-dependent. Our initial experiments along these directions, however, have not brought consistent improvement in photorealism in the case of the gaze redirection task.

4 Experiments

4.1 Dataset

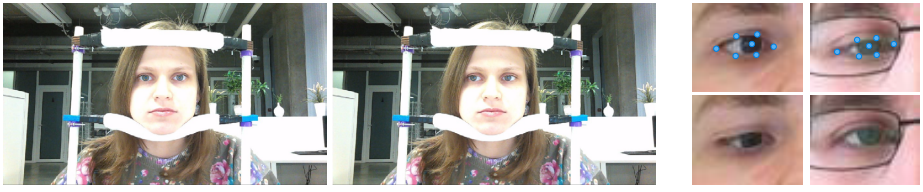


Fig. 4. Left – dataset collection process. Right – examples of two training pairs (input image with superimposed feature points on top, output image in the bottom).

There are no publicly available datasets suitable for the purpose of the gaze correction task with continuously varying redirection angle. Therefore, we collect our own dataset (Figure 4). To minimize head movement, a person places her head on a special stand and follows with her gaze a moving point on the screen in front of the stand. While the point is moving, we record several images with eyes looking in different fixed directions (about 200 for one video sequence) using a webcam mounted in the middle of the screen. For each person we record $2-10$ sequences, changing the head pose and light conditions between different sequences. Training pairs are collected, taking two images with different gaze directions from one sequence. We manually exclude bad shots, where a person is blinking or where she is not changing gaze direction monotonically as anticipated. Most of the experiments were done on the dataset of 33 persons and 98 sequences. Unless noted otherwise, we train the model for vertical gaze redirection in the range between $-30°$ and $30°$.

4.2 Training procedure

The model was trained end-to-end on 128-sized batches using Adam optimizer [14]. We used a regular ℓ_2-distance between the synthesized output O_{output} and the ground-truth O_{gt} as the objective function. We tried to improve over this simple baseline in several ways. First, we tried to put emphasis on the actual eye region (not the rectangular bounding-box) by adding more weight to the corresponding pixels but were not able to get any significant improvements. Our earlier experiments with adversarial loss [8] were also inconclusive. As the residual flow predicted by the $1\times$-scale module tends to be quite noisy, we attempted to smoothen the flow-field by imposing a total variation penalty. Unfortunately, this resulted in a slightly worse ℓ_2-loss on the test set.

Sampling training pairs. We found that biasing the selection process for more difficult and unusual head poses and bigger redirection angles improved the results. For this reason, we used the following sampling scheme aimed at reducing the dataset imbalance. We split all possible correction angles (that is, the range between $-30°$ and $30°$) into 15 bins. A set of samples falling into a bin is further divided into "easy" and "hard" subsets depending on the input's *tilt* angle (an angle between the segment connecting two most distant eye feature points and the horizontal baseline). A sample is considered to be "hard" if its tilt is $\geqslant 8°$. This subdivision helps to identify training pairs corresponding to the rare head poses. We form a training batch by picking 4 correction angle bins uniformly at random and sampling 24 "easy" and 8 "hard" examples for each of the chosen bins.

4.3 Quantitative evaluation

We evaluate our approach on our dataset. We randomly split the initial set of subjects into a development (26 persons) and a test (7 persons) sets. Several methods were compared using the mean square error (MSE) between the synthesized and the ground-truth images extracted using the procedure described in Section 3.1.

Models. We consider 6 different models:

1. A system based on Structured Random Forests (*RF*) proposed in [16]. We train it for $15°$ redirection only using the reference implementation.
2. A single-scale (*SS* ($15°$ only)) version of our method with a single warping module operating on the original image scale that is trained for $15°$ redirection only.
3. A single-scale (*SS*) version of our method with a single warping module operating on the original image scale.
4. A multi-scale (*MS*) network without coarse warping. It processes inputs on two scales and uses features from both scales to predict the final warping transformation.
5. A coarse-to-fine warping-based system described in Section 3 (*CFW*).
6. A coarse-to-fine warping-based system with a lightness correction module (*CFW + LCM*).

The latter four models are trained for the task of vertical gaze redirection in the range. We call such models *unified* (as opposed to single angle correction systems).

15° correction. In order to have the common ground with the existing systems, we first restrict ourselves to the case of $15°$ gaze correction. Following [16], we present a graph of sorted normalized errors (Figure 5), where all errors are divided by the MSE obtained by an input image and then the errors on the test set are sorted for each model.

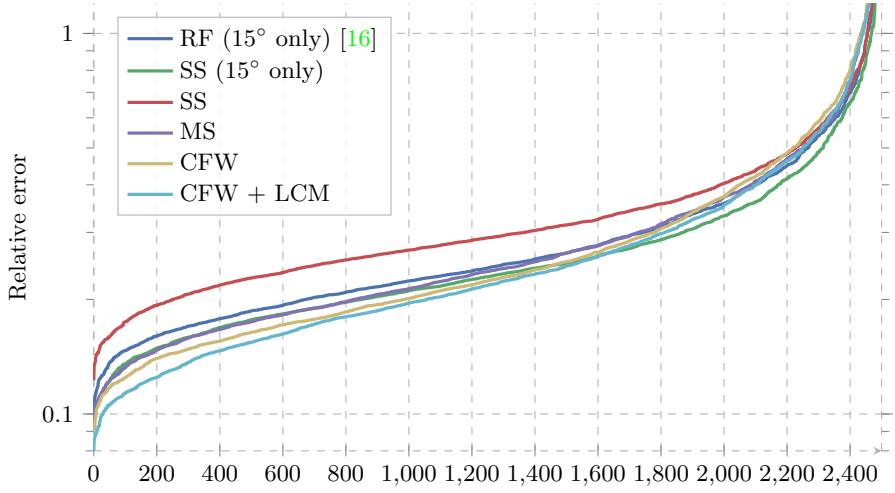


Fig. 5. Ordered errors for 15° redirection. Our multi-scale models (MS, CFW, CFW + LCM) show results that are comparable or superior the Random Forests (RF) [16].

It can be seen that the unified multi-scale models are, in general, comparable or superior to the RF-based approach in [16]. Interestingly, the lightness adjustment extension (Section 3.4) is able to show quite significant improvements for the samples with low MSE. Those are are mostly cases similar to shown in Figure 3. It is also worth noting that the single-scale model trained for this specific correction angle consistently outperforms [16], demonstrating the power of the proposed architecture. However, we note that results of the methods can be improved using additional registration procedure, one example of which is described in Section 4.5.

Arbitrary vertical redirection. We also compare different variants of unified networks and plot the error distribution over different redirection angles (Figure 6). For small angles, all the methods demonstrate roughly the same performance, but as we increase the amount of correction, the task becomes much harder (which is reflected by the growing error) revealing the difference between the models. Again, the best results are achieved by the palette model, which is followed by the multi-scale networks making use of coarse warping.

4.4 Perceptual quality

We demonstrate the results of redirection on 15 degrees upwards in the Figure 7. CFW-based systems produce the results visually closer to the ground truth, than RF. The effect of the lightness correction is pronounced: on the input image with the lack of white Random Forest and CFW fail to get output with sufficient eye-white and copy-paste red pixels instead, whereas CFW+LCM achieve good

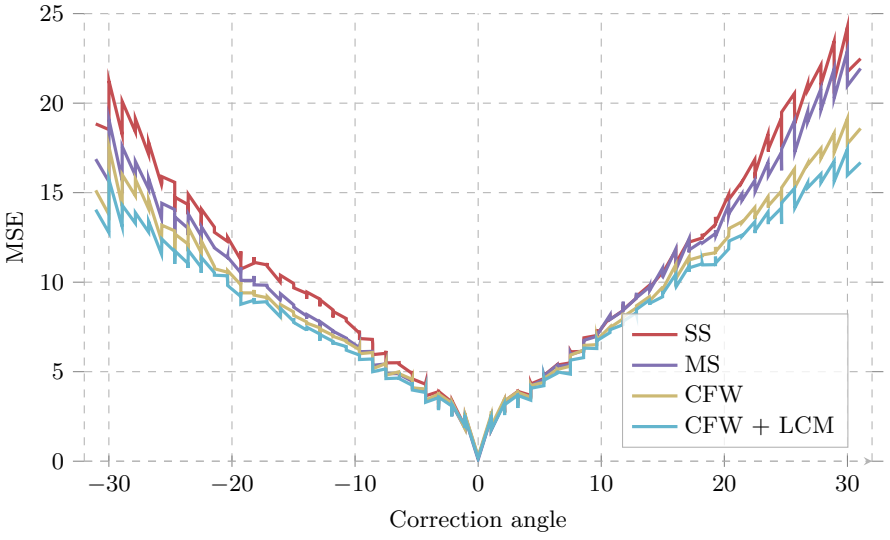


Fig. 6. Distribution of errors over different correction angles.

correspondence with the ground-truth. However, the downside effect of the LCM could be blurring/lower contrast because of the multiplication procedure (6).

User study To confirm the improvement corresponding to different aspects of the proposed models, which may not be adequately reflected by ℓ_2-measure, we performed an informal user study enrolling 16 subjects unrelated to computer vision and comparing four methods (RF, SS, CFW, CFW+LCM). Each user was shown 160 quadruplets of images, and in each quadruplet one of the images was obtained by re-synthesis with one of the methods, while the remaining three were unprocessed real images of eyes. 40 randomly sampled results from each of the compared methods were thus embedded. When a quadruplet was shown, the task of the subject was to click on the artificial (re-synthesized) image as quickly as possible. For each method, we then recorded the number of correct guesses out of 40 (for an ideal method the expected number would be 10, and for a very poor one it would be 40). We also recorded the time that the subject took to decide on each quadruplet (better method would take a longer time for spotting). Table 1 shows results of the experiment. Notably, here the gap between methods is much wider then it might seem from the MSE-based comparisons, with CFW+LCM method outperforming others very considerably, especially when taking into account the timings.

Horizontal redirection. While most of our experiments were about vertical gaze redirection, the same models can be trained to redirect the gaze horizontally (and, with trivial generalization, by a 2D family of angles). In Figure 8,

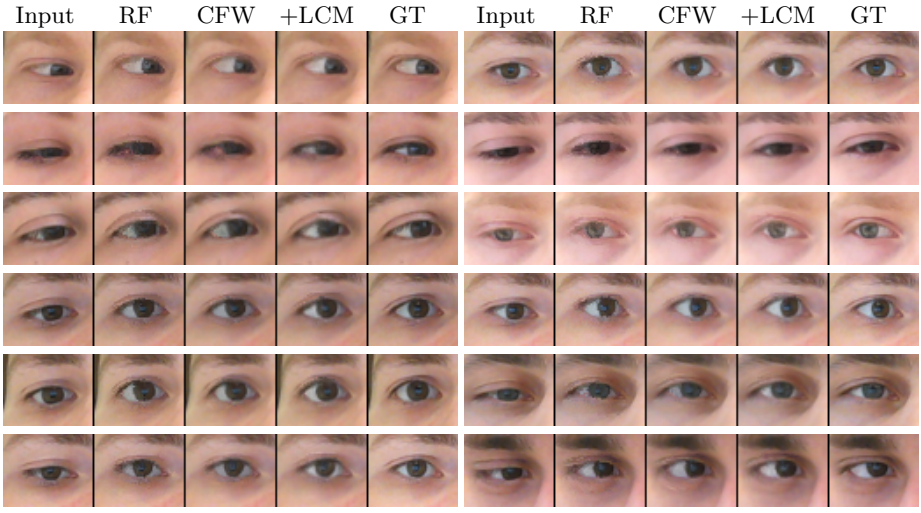


Fig. 7. Sample results on a hold-out. The full version of our model (CFW+LCM) outperforms other methods.

Table 1. User assessment for the photorealism of the results for the four methods. During the session, each of the 16 test subjects observed 40 instances of results of each method embedded within 3 real eye images. The participants were asked to click on the resynthesized image in as little time as they could. The first three parts of the table specify the number of correct guesses (the smaller the better). The last line indicates the mean time needed to make a guess (the larger the better). Our full system (coarse-to-fine warping and lightness correction) dominated the performance.

	Random Forest	**Single Scale**	**CFW**	**CFW+LCM**
Correctly guessed (out of 40)				
Mean	36.1	33.8	28.8	**25.3**
Median	37	35	29	**25**
Max	40	39	38	**34**
Min	26	22	20	**16**
Correctly guessed within 2 seconds (out of 40)				
Mean	26.4	21.1	11.7	**8.0**
Median	28.5	20.5	10	**8**
Max	35	33	23	**17**
Min	13	11	3	**0**
Correctly guessed within 1 second (out of 40)				
Mean	8.1	4.4	1.6	**1.1**
Median	6	3	**1**	**1**
Max	20	15	7	**5**
Min	**0**	**0**	**0**	**0**
Mean time to make a guess				
Mean time, sec	1.89	2.30	3.60	**3.96**

we provide qualitative results of CFW+LCM for horizontal redirection. Some examples showing the limitations of our method are given. The limitations are concerned with cases with severe disocclusions, where large areas have to be filled by the network.

We provide more qualitative results on the project webpage [4].

4.5 Incorporating registration

We found that results can be further perceptually improved (see [4]) if the objective is slightly modified to take into account misalignment between inputs and ground-truth images. To that end, we enlarge the bounding-box $\mathcal{B}$ that we use to extract the output image of a training pair by $k = 3$ pixels in all the directions. Given that now O_{gt} has the size of $(H + 2k) \times (W + 2k)$, the new objective is defined as:

$$\mathcal{L}(O_{\text{output}}, O_{\text{gt}}) = \min_{i,j} \text{dist}\left(O_{\text{output}}, O_{\text{gt}}[i : i + H, j : j + W]\right) , \tag{7}$$

where $\text{dist}(\cdot)$ can be either ℓ_2 or ℓ_1-distance (the latter giving slightly sharper results), and $O_{\text{gt}}[i : i+H, j : j+W]$ corresponds to a $H \times W$ crop of O_{gt} with top left corner at the position (i, j). Being an alternative to the offline registration of input/ground-truth pairs [16] which is computationally prohibitive in large-scale scenarios, this small trick greatly increases robustness of the training procedure against small misalignments in a training set.

5 Discussion

We have suggested a method for realistic gaze redirection, allowing to change gaze continuously in a certain range. At the core of our approach is the prediction of the warping field using a deep convolutional network. We embed redirection angle and feature points as image-sized maps and suggest "fully-convolutional" coarse-to-fine architecture of warping modules. In addition to warping, photorealism is increased using lightness correction module. Quantitative comparison of MSE-error, qualitative examples and a user study show the advantage of suggested techniques and the benefit of their combination within an end-to-end learnable framework.

Our system is reasonably robust against different head poses (e.g., see Figure 3) and deals correctly with the situations where a person wears glasses (see [4]). Most of the failure modes (e.g., corresponding to extremely tilted head poses or large redirection angles involving disocclusion of the different parts of an eye) are not inherent to the model design and can be addressed by augmenting the training data with appropriate examples.

We concentrated on gaze redirection, although our approach might be extended for other similar tasks, e.g. re-synthesis of faces. In contrast with autoencoders-based approach, our architecture does not compress data to a representation with lower explicit or implicit dimension, but directly transforms the

Fig. 8. Horizontal redirection with a model trained for both vertical and horizontal gaze redirection. For the first six rows the angle varies from $-15°$ to $15°$ relative to the central (input) image. The last two rows push the redirection to extreme angles (up to $45°$) breaking our model down.

input image. Our method thus might be better suited for fine detail preservation, and less prone to the "regression-to-mean" effect.

The computational performance of our method is up to 20 fps on a mid-range consumer GPU (NVIDIA GeForce-750M), which is however slower than the competing method of [16], which is able to achieve similar speed on CPU. Our models are however much more compact than forests from [16] (250 Kb vs 30-60 Mb in our comparisons), while also being universal. We are currently working on the unification of the two approaches.

Speed optimization of the proposed system is another topic for future work. Finally, we plan to further investigate non-standard loss functions for our architectures (e.g. the one proposed in Section 4.5), as the ℓ_2-loss is not closely enough related to perceptual quality of results (as highlighted by our user study).

Acknowledgements

We would like to thank Leonid Ekimov for sharing the results of his work on applying auto-encoders for gaze correction. We are also grateful to all the Skoltech students and employees who agreed to participate in the dataset collection and in the user study. This research is supported by the Skoltech Translational Research and Innovation Program.

References

1. Criminisi, A., Shotton, J., Blake, A., Torr, P.H.: Gaze manipulation for one-to-one teleconferencing. In: ICCV (2003)
2. Denton, E.L., Chintala, S., Fergus, R., et al.: Deep generative image models using a laplacian pyramid of adversarial networks. In: NIPS (2015)
3. Dosovitskiy, A., Tobias Springenberg, J., Brox, T.: Learning to generate chairs with convolutional neural networks. In: CVPR (2015)
4. Ganin, Y., Kononenko, D., Sungatullina, D., Lempitsky, V.: Project website. http://sites.skoltech.ru/compvision/projects/deepwarp/ (2016), [Online; accessed 22-July-2016]
5. Gatys, L., Ecker, A.S., Bethge, M.: Texture synthesis using convolutional neural networks. In: NIPS (2015)
6. Ghodrati, A., Jia, X., Pedersoli, M., Tuytelaars, T.: Towards automatic image editing: Learning to see another you. arXiv preprint arXiv:1511.08446 (2015)
7. Giger, D., Bazin, J.C., Kuster, C., Popa, T., Gross, M.: Gaze correction with a single webcam. In: IEEE International Conference on Multimedia & Expo (2014)
8. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS (2014)
9. Gregor, K., Danihelka, I., Graves, A., Rezende, D., Wierstra, D.: Draw: A recurrent neural network for image generation. In: ICML (2015)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385 (2015)
11. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. ICML (2015)
12. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: NIPS (2015)
13. Jones, A., Lang, M., Fyffe, G., Yu, X., Busch, J., McDowall, I., Bolas, M.T., Debevec, P.E.: Achieving eye contact in a one-to-many 3D video teleconferencing system. ACM Trans. Graph. 28(3) (2009)
14. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
15. Kleinke, C.L.: Gaze and eye contact: a research review. Psychological bulletin 100(1), 78 (1986)
16. Kononenko, D., Lempitsky, V.: Learning to look up: realtime monocular gaze correction using machine learning. In: CVPR (2015)
17. Kulkarni, T.D., Whitney, W.F., Kohli, P., Tenenbaum, J.: Deep convolutional inverse graphics network. In: NIPS (2015)
18. Kuster, C., Popa, T., Bazin, J.C., Gotsman, C., Gross, M.: Gaze correction for home video conferencing. In: SIGGRAPH Asia (2012)
19. Mahendran, A., Vedaldi, A.: Understanding deep image representations by inverting them. In: CVPR (2015)
20. Okada, K.I., Maeda, F., Ichikawaa, Y., Matsushita, Y.: Multiparty videoconferencing at virtual social distance: Majic design. In: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work. pp. 385–393. CSCW '94 (1994)
21. Oquab, M.: Torch7 modules for spatial transformer networks. https://github.com/qassemoquab/stnbhwd (2015)
22. Reed, S.E., Zhang, Y., Zhang, Y., Lee, H.: Deep visual analogy-making. In: NIPS (2015)

23. Wallis, L.J., Range, F., Müller, C.A., Serisier, S., Huber, L., Virányi, Z.: Training for eye contact modulates gaze following in dogs. Animal behaviour 106, 27–35 (2015)
24. Wolf, L., Freund, Z., Avidan, S.: An eye for an eye: A single camera gaze-replacement method. In: CVPR (2010)
25. Xiong, X., Torre, F.: Supervised descent method and its applications to face alignment. In: CVPR (2013)
26. Yang, R., Zhang, Z.: Eye gaze correction with stereovision for video-teleconferencing. In: ECCV (2002)
27. Yip, B., Jin, J.S.: Face re-orientation using ellipsoid model in video conference. In: Proc. 7th IASTED International Conference on Internet and Multimedia Systems and Applications. pp. 245–250 (2003)
28. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: ECCV, pp. 818–833. Springer (2014)

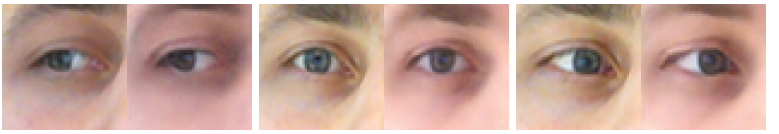


Fig. 9. Examples of reconstructions produced by a modern **encoder-decoder** architecture (following the approach in [17,22,6]) trained on our data. In each pair, the left image is the input and the right is the output. Despite our efforts, a noticeable loss of fine-scale details and "regression-to-mean" effect make the result not good enough for most applications of gaze manipulation. Similar problems can be observed in [17,22,6].

Appendix A Drawbacks of conventional architectures

In order to determine the applicability of conventional generative architectures for gaze correction, we used our data to train several auto-encoders. The best model has 200-dimensional latent space and consists of several convolutional and fully-connected layers in the encoder and the decoder. We use a combination of ℓ_2 and GAN [8] losses to achieve the best possible results. Unfortunately, due to inherently lossy encoding procedure, the model exhibits noticeable fine-scale details dropping and "regression-to-mean" effect (see Figure 9). That makes incorporation of such kind of approach into a gaze correction system problematic.

Appendix B Details of the proposed method

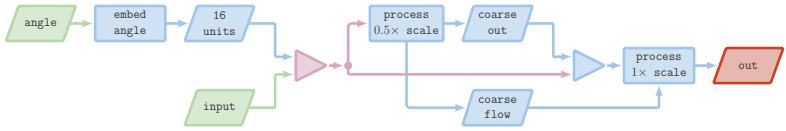
Here we give a more detailed view of the architecture that we use in our gaze correction system.

Appendix B.1 Warping stage

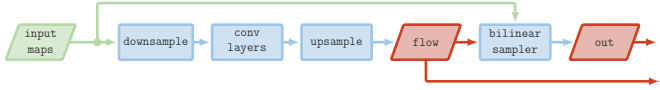
A flowchart of the pipeline *without* the lightness correction module is depicted in Figure 10. To allow for more interaction between the input data and the correction angle embedding, we choose not to perform late fusion [6] and feed the embedding vector as an additional input to the warping network. More concretely, we replicate the 16-dimensional $\mathbf{F}_{\text{angle}}(\alpha)$ for every spatial location creating a tensor of size $16 \times H \times W$ which is then concatenated to the rest of the input (*pink* triangle in Figure 10(a)). The architectures of the two warping modules are same (modulo the number of input maps) and are shown in Figure 10(d).

Appendix B.2 Lightness correction module

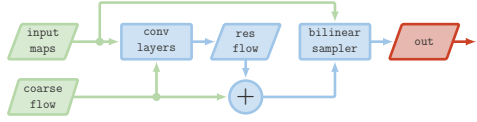
Figure 11 shows the actual architecture for the lightness correction module. Per-pixel weights are predicted based on the internal activations (the third convolutional layer) of the warping modules ($0.5\times$`-scale` and $1\times$`-scale features` in the scheme respectively).



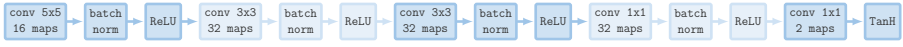
(a) High-level pipeline



(b) 0.5×-scale processing module



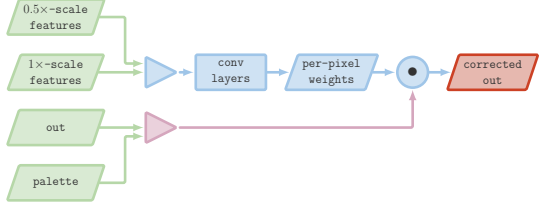
(c) 1×-scale processing module



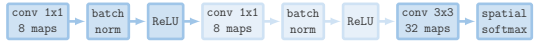
(d) Convolutional layers

(e) Legend

Fig. 10. The **basic warping architecture** 10(a) takes an input eye region augmented with eye feature points information (`input`) as well as a correction `angle` and produces an image of the redirected eye. The model contains three main blocks: angle embedding module (`embed angle`) calculating a vector representation of the correction `angle` and two warping modules (`process 0.5×-scale` 10(b) and `process 1×-scale` 10(c)) predicting and applying pixel-flow to the input image.



(a) Architecture.



(b) Convolutional layers.

Fig. 11. Lightness Correction Module increases lightness of selected regions. 11(a) shows the actual architecture of the module. Multi-scale features are processed by the convolutional neural network presented in 11(b).