Semi-Supervised Learning for Monocular Gaze Redirection

Daniil Kononenko, Victor Lempitsky

Skolkovo Institute of Science and Technology (Skoltech), Moscow, Russia

***Abstract*—We present a new approach to monocular learning-based gaze redirection problem in images that is able to train on raw sequences of eye images with unknown gaze directions and a small amount of eye images, where the gaze direction is known. The proposed approach is based on a pair of deep networks, where the first encoder-like network maps eye images to a latent space, while the second network maps pairs of latent representations to warping fields implementing the transformation between the pair of the original images. In the proposed system, both networks are trained in an unsupervised manner, while the gaze-annotated images are only used to estimate displacements in the latent space that are characteristic to certain gaze redirections. Quantitative and qualitative evaluation suggests that such characteristic displacement vectors in the learned latent space can be learned from few examples and are transferable across different people and different imaging conditions.**

I. Introduction

In this work, we consider the task of gaze redirection in images and video frames. The problem is an important particular case of image (re)-synthesis and has immediate applications in video conferencing, where the purpose of gaze manipulation is to restore an eye-to-eye contact, as well as in movie and photo post-production. Recently, the approaches [1], [2], [3] have demonstrated that realistic gaze redirection is possible in monocular setting, i.e. without any additional hardware other than a single camera that is used to acquire the images or videos. The warping-based model of [1], [2], [3] however relies heavily on supervised machine learning, and in particular requires a considerable amount of eye images labeled with gaze direction. Acquiring such images is tedious and requires imaging of multiple people in constrained and uncomfortable setting.

Here, we extend the warping-based model of [1], [2], [3] to unsupervised and semi-supervised settings, where most of the learning happens in an unsupervised way using sequences of eye images of different people with varying and unknown gaze direction. In more detail, we use unsupervised learning to construct the deep embedding of eye images into a low-dimensional latent space (via the *encoder* network) and, in parallel, learn a *decoder* network that constructs a warping flow field based on the latent representation of two eyes from the same sequence.

Once the unsupervised training is accomplished, the system can redirect gaze of an arbitrary eye image by mapping it to a latent space, and then modifying its latent representation by adding a certain vector in order to estimate the latent representation of the target image. The decoder network can then be used to estimate the warping between the source and the unknown target images. The presented model is similar in spirit to the visual analogy making of [4] though it predicts the warping fields rather than the target images directly. The model can use a small amount of supervised data (e.g. a single pair of eye images with known difference in gaze direction) to estimate the displacements in latent space that are characteristics to certain gaze redirections (e.g. lifting the gaze by 15 degrees upwards as is practical to the videoconferencing scenario).

As we show in the experiments (Section III), the resulting semi-supervised solution achieves convincing gaze redirection, which outperforms in visual quality the fully supervised solution of [2] in the case when supervised data are limited. At the same time, relying on warping rather than direct re-synthesis ensures high realism of the resulting images and avoids the loss of high-frequency details.

Related work. As discussed above, our work is the continuation of the DeepWarp system [2]. Similar fully-supervised models for general types of images have also been suggested recently in [5], [6], [7]. All these works rely on bilinear sampling layer popularized by [8] as part of their spatial transformer networks. Our model is related to the deep-analogy making model of [4]. The model [4] however requires the knowledge of transformations that needs to be provided in the form of analogy-forming quadruplets, which is not required by our model. Perhaps even more related is the inverse graphics model of [9], which can be trained with a similar level of supervision to our system, i.e. with subsets of images where some factors of variations are fixed while others are varying arbitrarily. Both [4] and [9] however tend to produce blurry non-photorealistic images that are not suitable for the application scenarios of gaze redirection. This is overcome in our model by using warping instead of direct re-synthesis. Our work is also related to [10], where a physical model of the eye is used to solve the gaze redirection problem.

II. Unsupervised training of gaze redirection

Data collection and preprocessing. Before discussing the details of our approach, we review the data collection and preprocessing procedures. To collect sequences with annotated gaze direction, we follow [1], [2], [3] and record the images of a person with fixed head position and following a dot on a screen. A two minutes length sequence is recorded by a webcam mounted in the middle of the screen. We manually sort out bad shots with eye blinking, head shaking, or where the gaze is not changing monotonically as anticipated. For different sequences, we vary the head pose and lightning conditions. Gaze direction changes in range from -30 to 30 degrees in both x and y directions (relatively the camera in the center of the screen).

978-1-5386-2335-0/18/$31.00

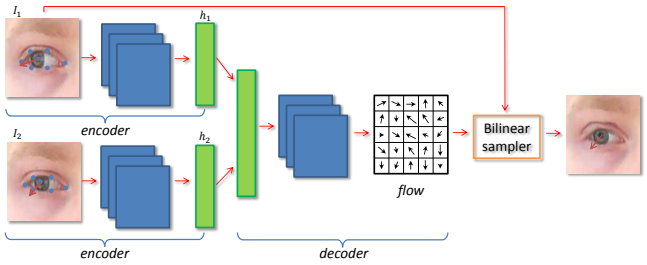

Figure 1: Architecture of unsupervised gaze redirection training. Two images of the same eye with different gaze direction are passed to the *encoder* network that outputs their latent representations. These representations are then concatenated and passed to the decoder network that outputs the predicted flow from the first image to the second one. The flow can be used by the bilinear sampler. The architecture is trained by minimizing the disparity between the output image and the second image. Blue dots on eyes represent the input distance maps, and arrows represent intended gaze direction.

For the unlabeled part of dataset the process is significantly simplified. The person is instructed to keep the head approximately still and to quickly move the gaze along the screen for about 10 seconds. This recording time is comfortable for not blinking and not shaking head, while sufficiently long for a person to gaze at different parts of the screen. This scenario also eliminates the problems with the person not following the dot on the screen as prescribed, which we found out to be a recurrent problem.

Following the approach of [1], we emulate the change in the appearance of eyes, when a person changes a gaze direction, keeping the head pose unchanged. Thus, all image modifications are concentrated in the close vicinity of each eye. We use off-the-shelf facial landmark detection software [11], [12] to localize eye. We use the method suggested in [2] for an eye cropping, which computes tight bounding box around eye landmarks and then enlarges it proportionally to the distance between the eye corners. Located eye landmarks are also embedded in the architecture as additional features in both training and testing of the model. All models assume they are dealing with right eyes, while left eyes are processed using symmetry.

Model architecture. We now discuss the architecture of our approach (Figure 1) as well as the training of the encoder and the decoder networks, which, as discussed above, happens in unsupervised mode and utilizes only image sequences with varying but unknown gaze direction.

In general, similarly to [1], [2], [3], we perform the gaze redirection by warping the input eye images. Thus, at the core of our system is the ability to model the change of appearance within the pair of the eye images (I_1, I_2) from the same video sequence using warping. Such warping is determined by the *latent* representations of the images $h_1 = \mathbf{E}(I_1; \psi)$, $h_2 = \mathbf{E}(I_2; \psi)$, where $\mathbf{E}$ denotes a feed-forward *encoder* network with learnable parameters ψ. The latent representations live in low-to-medium dimensional space (up to 500 dimensions in our experiments).

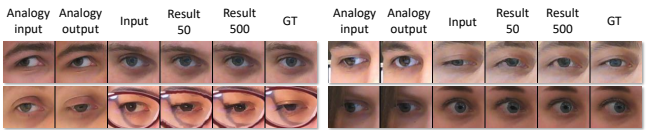

Figure 2: Demonstration of analogy property of the learned embedding (zoom-in recommended). The three left-most columns alongside the right-most one form analogy quadruplets (the difference in gaze direction between the first two columns is approximately the same as the difference in gaze direction between the third and the last columns). 'Results 50' and 'Results 500' demonstrate the warped images obtained using modification in latent space (as discussed in the text), after the encoder and the decoder are trained in an unsupervised setting on 50 or 500 eye sequences respectively. Overall, training on more unsupervised data (500 sequences) leads to better and more robust result.

Given the latent representations of the images, a *decoder* network $\mathbf{D}$ with learnable parameters ω is applied to the stacked latent representations of the image pair and outputs the warping field, corresponding to the transformation of image I_1 into I_2: $F = \mathbf{D}(h_1, h_2; \omega)$. Finally, the standard bilinear sampling layer $\mathbf{S}$ as defined in [8] outputs the result, which is the prediction $\hat{I}_2$ of image I_2. Overall, the warping process can be written as:

$$\hat{I}_2(I_1; \psi, \omega) = \mathbf{S}(I_1, \mathbf{D}(\mathbf{E}(I_1; \psi), \mathbf{E}(I_2; \psi); \omega))). \quad (1)$$

The training objective then naturally corresponds to minimizing the disparity between the true image I_2 and its predicted version (1). The training process then corresponds to sampling pairs $(I_1; I_2)$ and optimizing the parameters for the encoder and the decoder networks by minimizing the following ℓ_2-loss $L(\psi, \omega) = \sum_{(I_1, I_2)} \|\hat{I}_2(I_1; \psi, \omega) - I_2\|^2$, where the summation is taken over all training pairs of eye images that correspond to the same sequences.

Notably, the training process does not require gaze annotation, and, as will be verified below, learns meaningful latent representations that are *consistent* across eye sequences in the following sense. Let a *visual analogy* be quadruplet (I_1, I_2, I_3, I_4), in which I_1 and I_2 correspond to one eye sequence, and I_3 and I_4 correspond to other eye sequence (of a potentially different person and/or different lighting etc.), and where the change of gaze direction from I_1 to I_2 and from I_3 to I_4 are similar (Figure 2). The learned embeddings possess the property of having similar displacement vectors across the two pairs:

$$\mathbf{E}(I_1; \psi) - \mathbf{E}(I_2; \psi) \approx \mathbf{E}(I_3; \psi) - \mathbf{E}(I_4; \psi) \quad (2)$$

The property (2) facilitates easy semi-supervised training with limited amount of gaze-annotated data.

Details of the architectures. Using eye feature coordinates as an input was important for the success of the system in [2], and we follow this by making the encoder networks to accept the locations of the eye feature coordinates as part of the input alongside the input image. We thus add additional

14 maps to the input three-channeled image, one for each of seven eye landmarks' coordinates. Each of the added maps encodes the distance from the pixel to the landmark along the chosen coordinate (either x or y).

To describe the specific architectures, we denote conv(m, k, s) a convolutional layer with m maps, kernel size k and size of the stride s, and FC(m) a fully connected layer with m maps. The architecture of the encoder we used in our experiments is the following: conv$(48, 5, 1) \rightarrow$ conv$(48, 5, 2) \rightarrow$ conv$(96, 5, 2) \rightarrow$ conv$(96, 3, 2) \rightarrow$ FC$(800) \rightarrow$ FC(50). The decoder mirrors the architecture of the encoder, except for the input (which is the vector of length 100, being a concatenation of two representations of length 50) and the output, which are two maps of the warping field used in (1). The model is trained using Adam optimizer [13]. Each batch contains 128 randomly sampled pairs of images, each pair consisting of the input and output eye from the same sequence.

Semi-supervised learning. The architecture discussed above trains on pairs of eye images, and treat each of the images in the pair similarly. At test time, however, we are interested in computing the warping field *without* knowing the second image (which itself is the unknown that we wish to estimate). Fortunately, the analogy property (2) possessed by the embeddings allows us to estimate characteristic displacements in the latent space given some amount of gaze direction-annotated data obtained with the time-consuming process.

We consider the following test-time gaze redirection problem: given the query image I_q, obtain the image O_q corresponding to the same eye under the same imaging condition, with the gaze redirected by a given angle $\alpha_q = (\alpha_q^x, \alpha_q^y)$. As the angle α_q is given, we can query the direction-annotated part of the dataset for the set of pairs $P(\alpha_q) = \{(I_1^1, I_2^1), \ldots, (I_1^n, I_2^n)\}$ that would form an analogy with I_q and O_q, i.e. the pairs with the difference in the gaze direction within each pair approximately equal α_q (in practice we use a hard threshold ϵ to determine whether some angular difference is close enough to α_q).

We then consider two methods of computing O_q given the set of pairs $P(\alpha_q)$. The first (baseline) method is to use the *mean warping field* of the set of analogy pairs. Here, for each pair we calculate the predicted warping field from the first image in the pair to the second, and then apply the averaged warping field $\bar{F}$ to the query image:

$$\bar{F} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{D}(\mathbf{E}(I_1^i; \psi), \mathbf{E}(I_2^i; \psi); \omega). \quad (3)$$

However, the clear drawback of this method is that the same warping field $\bar{F}$ will be applied to all query images with the same desired angular redirection α_q, being independent from the content of the query I_q.

The second method that directly relies on the analogy property (2) computes the mean latent vector displacement

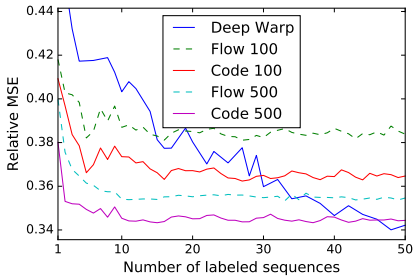

Figure 3: Quantitative comparison of methods: errors for redirection on 15 degrees upwards (zoom-in recommended). Horizontal axis shows the number of sequences labeled with gaze direction provided to the methods. All methods are trained for arbitrary redirection angle, but applied to a testing setting with 15° vertical redirection. "Code" corresponds to estimating the warped image latent representation (5), "flow" corresponds to warping the input image using mean flow (3). 100/500 corresponds to the number of unlabeled sequences used to train the encoder-decoder model. The single scale Deep Warp system [2] does not use unlabeled data. Semi-supervised system performing analogies in latent representation space outperforms other methods, and training this method on more unlabeled data helps a lot irrespective the amount of labeled data.

corresponding to angle α_q:

$$\Delta h(\alpha_q) = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{E}(I_2^i; \psi) - \mathbf{E}(I_1^i; \psi)). \quad (4)$$

Such precomputed vector can be used to estimate the desired output image as:

$$\hat{O}_q = \mathbf{S}(I_q,\ \mathbf{D}\left(\mathbf{E}(I_q; \psi),\ \mathbf{E}(I_q; \psi) + \Delta h(\alpha_q); \omega\right)). \quad (5)$$

All latent representations for labeled part of the dataset could be precomputed in advance and stored. Thus, performing the redirection following (5) requires only a single pass through the encoder and the decoder network at test time, which opens up a possibility for real-time gaze manipulation (on a device with a GPU).

III. Experiments

We perform our experiments using the dataset that consists of 640 sequences, each containing images of the same eye from one video (under the same lightning conditions, head pose, etc.) with different known gaze directions. Each sequence contains $100 - 220$ images. We use 500 sequences for training and validation, leaving 140 for testing (the train and the test sets do not contain sequences of the same people). The angular tolerance ϵ for picking up analogies from the labeled part of dataset was set to $0.5°$.

Quantitative evaluation of semi-supervised learning. We then perform quantitative evaluation for the task of fixed redirection angle 15° upwards (following the main setting in [2]). We consider the following methods:

- Our system trained on different amount of unlabeled sequences, as discussed in Section II, with two semi-supervised approaches for test-time prediction:

Method	DeepWarp	Flow100	Code100	Flow500	Code500
10 seqs	3.9	5.8	3.9	4.5	**2.7**
20 seqs	3.5	5.5	3.8	4.3	**2.7**

Table I: Mean absolute difference between the estimated actual redirection angle and the requested 15°. See text for details.

1) Based on mean displacement vector in representation space (5), denoted as "code".
2) Based on mean warping field (3), denoted as "flow".

- The single-scale Deep Warp system [2]. The single scale version was used as it is similar to our encoder-decoder network in complexity. Note that the encoder-decoder network architectures presented here also allow multi-scale extensions, as in [2], when a flow is predicted in a coarse-to-fine manner: residual flow is predicted on a finer scale, utilizing features from a coarse scale.

During training, all methods were trained for the task of redirection by an arbitrary angle (the redirection angle was fixed for the testing only). We vary the amount of labeled sequences shown to the methods. The unsupervised models were trained for 150 epochs on the unlabeled datasets containing either 100 or 500 sequences. For images from test set, we pick all possible analogies from given labeled sequences, and we vary the number of sequences in this labeled part. The Deep Warp system requires full supervision and therefore was trained only on the labeled part of the dataset for 150 epochs.

The quantitative comparison is represented in Figure 3. We evaluate the mean (over pixels) sum of squared errors across channels between the output and ground truth, and divide it by the MSE between the input and ground truth (referring to this measure as relative MSE). The semi-supervised models outperform the Deep Warp model, which does not exploit the unlabeled data, and, as expected, the advantage is bigger when the amount of labeled data is smaller. Increasing the number of unlabeled sequences also improves the performance of the model. The method based on latent representation (5) better exploits the trained unsupervised model, than the baseline which averages the warping flows. The performance of semi-supervised methods saturate after seeing approximately 15 labeled sequences. With an increase in the number of labeled sequences, Deep Warp begins to outperform the best of semi-supervised approaches (Code-500). In our experiment, this happens at around 45 labeled sequences. Due to a restricted demographics of our dataset (mostly Caucasian and young people were imaged), the number 45 is probably an underestimate of the number that would be observed on a more diverse/balanced dataset. For the reference, the *Unsupervised oracle* baseline corresponding to the unsupervised model that knows the latent representation of the ground truth and uses it to estimate the warping field, relative MSE 0.32 for 100 sequences of unlabeled data and 0.24 for 500 sequences.

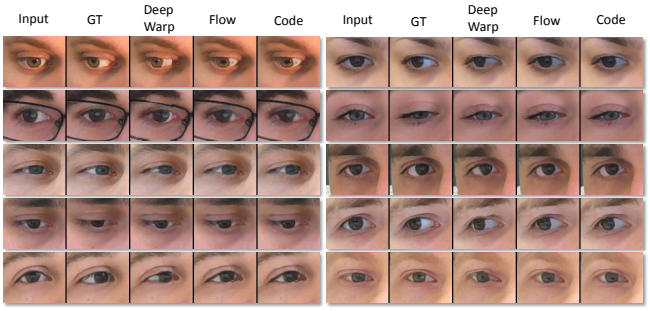

Figure 4: Sample results on the hold-out set (zoom-in recommended). Columns from left to right: the input image, the ground truth, the results of single scale deep warp system, the result of the semi-supervised model that uses mean warping field, the result of the semi-supervised model that uses mean difference in latent representation space. With limited labeled data the perceptual quality of the results is significantly improved using large dataset of unlabeled data.

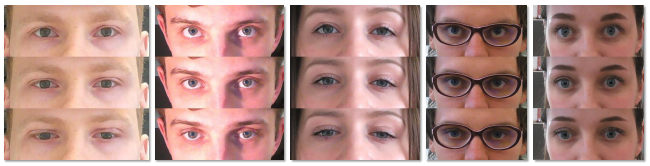

Figure 5: Vertical redirection by ± 15 degrees (zoom-in needed) by our semi-supervised model. The middle row shows the input. The rightmost column is an example of failure case.

To fully evaluate whether the gaze difference between the input image and the output equals the requested angle, we provide an additional assessment using evaluation model E. It was trained to estimate the vertical gaze difference between two input images on a large labeled training set (500 sequences). The validation error of the trained model is 1.1°. The evaluation score of the result O of the gaze redirection by 15° upwards in input image I is $|15 - E(I, O)|$, where $E(I, O)$ is the estimation of an actual redirection angle. Results for 10 and 20 labeled sequences are presented in the Table I. Semi-supervised system based on latent representation with 500 unlabeled sequences outperforms Deep Warp by a larger margin than in relative MSE comparison. However, the score of Deep Warp is increased with respect to other methods.

Qualitative evaluation of semi-supervised learning. Finally, we demonstrate the qualitative results of redirection on arbitrary angles in Figure 4 and Figure 5. All systems here (except DeepWarp) use 15 labeled and 500 unlabeled sequences. Performing analogies in the latent representation space allows to get a substantial perceptual improvement over the results of supervised model.

Acknowledgement. This study was supported by the Russian Federation MES grant 14.756.31.0001.

REFERENCES

[1] D. Kononenko and V. Lempitsky, "Learning to look up: realtime monocular gaze correction using machine learning," in *CVPR*, 2015.

[2] Y. Ganin, D. Kononenko, D. Sungatullina, and V. Lempitsky, "Deepwarp: Photorealistic image resynthesis for gaze manipulation," in *ECCV*, 2016.

[3] D. Kononenko, Y. Ganin, D. Sungatullina, and V. Lempitsky, "Photorealistic monocular gaze redirection using machine learning," *TPAMI*, 2017.

[4] S. E. Reed, Y. Zhang, Y. Zhang, and H. Lee, "Deep visual analogy-making," in *NIPS*, 2015.

[5] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros, "View synthesis by appearance flow," in *ECCV*, 2016.

[6] R. Yeh, Z. Liu, D. B. Goldman, and A. Agarwala, "Semantic facial expression editing using autoencoded flow," *CoRR*, vol. abs/1611.09961, 2016.

[7] Z. Liu, R. Yeh, X. Tang, Y. Liu, and A. Agarwala, "Video frame synthesis using deep voxel flow," *CoRR*, vol. abs/1702.02463, 2017.

[8] M. Jaderberg, K. Simonyan, A. Zisserman, *et al.*, "Spatial transformer networks," in *NIPS*, 2015.

[9] T. D. Kulkarni, W. F. Whitney, *et al.*, "Deep convolutional inverse graphics network," in *NIPS*, 2015.

[10] E. Wood, T. Baltrusaitis, L. Morency, P. Robinson, and A. Bulling, "Gazedirector: Fully articulated eye gaze redirection in video," *CoRR*, vol. abs/1704.08763, 2017.

[11] T. Baltrušaitis, P. Robinson, *et al.*, "Openface: an open source facial behavior analysis toolkit," in *WACV*, 2016.

[12] T. Baltrušaitis, P. Robinson, and L.-P. Morency, "Constrained local neural fields for robust facial landmark detection in the wild," in *CVPR workshops*, 2013.

[13] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.