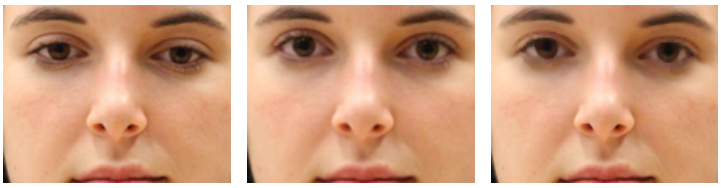
Learning To Look Up: Realtime Monocular Gaze Correction Using Machine Learning

Daniil Kononenko, Victor Lempitsky
Skolkovo Institute of Science and Technology (Skoltech).

The problem of gaze in videoconferencing has been attracting researchers and engineers for a long time. The problem manifests itself as the inability of the people engaged into a videoconferencing (the proverbial "Alice" and "Bob") to maintain gaze contact. The lack of gaze contact is due to the disparity between Bob's camera and the image of Alice's face on Bob's screen (and vice versa).



Input "Ground truth" Our output

Figure 1: The input and the "ground truth" frames (from the Columbia gaze dataset [3]) have ten degrees difference in gaze direction. The result of our method is shown on the right (redirecting ten degrees upwards). The computation time is 8 ms on a single laptop core (excluding feature point localization).

We revisit the problem of gaze correction and present a solution based on supervised machine learning. At training time, our system observes pairs of images, where each pair contains the face of the same person with a fixed angular difference in gaze direction. It then learns to synthesize the second image of a pair from the first one. After learning, the system becomes able to redirect the gaze of a previously unseen person by the same angular difference as in the training set (10 or 15 degrees upwards in our experiments). Unlike many previous solutions to gaze problem in videoconferencing, ours is purely monocular, i.e. it does not require any hardware apart from an in-built web-camera of a laptop. Being based on efficient machine learning predictors such as decision forests, the system is fast (runs in real-time on a single core of a modern laptop). In the paper, we demonstrate results on a variety of videoconferencing frames and evaluate the method quantitatively on the hold-out set of registered images. The supplementary video at the project website shows example sessions of our system at work.

We do not attempt to synthesize a view for a virtual camera, as in [1], or synthesize virtual view of the face only and stitch with the original videostream, as in [2]. Instead, our method emulates the change in the appearance resulting from a person changing her/his gaze direction by a certain angle (e.g. ten degrees upwards), while keeping the head pose unchanged.

We use an off-the-shelf real-time face alignment library to localize facial feature points. For each eye, we compute a loose axis-aligned bounding box. At test time, for every pixel (x,y) in the bounding box we determine a 2D *eye flow vector* $(u(x,y),v(x,y))$. The pixel value at (x,y) is then "copy-pasted" from another location in the input image that is determined by using the eye flow vector (u,v) as an offset.

The main variant of our system computes the appropriate eye flow vector using a special kind of randomized decision tree ensembles. The vector is determined based on the appearance of the patch surrounding the pixel, and the location of the pixel w.r.t. the feature points.

In more detail, a pixel is passed through a set of specially-trained ensemble of randomized decision trees (*eye flow trees*, Figure 2). Two kinds of tests are applied to a pixel. *An appearance test* compares the difference of two pixel values in some color channel with the threshold. *A location test* compares the distance to some of the feature points in one of two dimensions with the threshold. To handle scale variations, we rescale the training images to the same characteristic size, and we rescale the offsets in the tree tests and the obtained eye flow vectors by an appropriate ratio at test time.

This is an extended abstract. The full paper is available at the Computer Vision Foundation webpage.

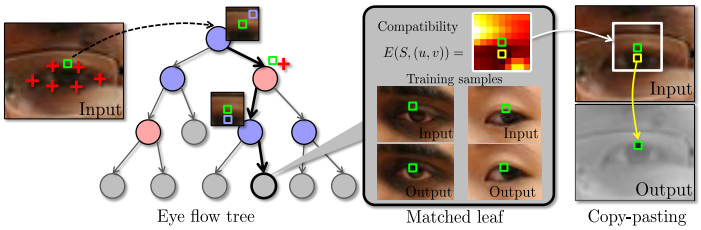


Figure 2: **Processing of a pixel (green square) at test time in an eye flow tree.** The pixel is passed through an eye flow tree by applying a sequence of tests. Once a leaf is reached, this leaf defines a matching of an input pixel with other pixels in the training data. The leaf stores the map of the compatibilities between such pixels and eye flow vectors. The system then takes the optimal eye flow vector and uses it to copy-paste an appropriately-displaced pixel in place of the input pixel into the output image.

In an eye flow tree, each leaf stores the map of *compatibilities* between the set of training examples and each possible offset (eye flow vector). We then sum together the compatibility maps from several trees, and pick the eye flow vector (u,v) that minimizes the aggregated map.

At learning phase we assume that a set of training image pairs (I^j,O^j) is given. We assume that within each pair, the images correspond to the same head pose of the same person, same imaging conditions, etc., and differ only in the gaze direction. We also rescale all pairs based on the characteristic radius of the eye in the input image. Eye flow trees are trained in a weakly-supervised manner, as each training sample does not include the target vectors $(u(x,y),v(x,y))$. The goal of the training is to build a tree that splits the space of training examples into regions, so that for each region replacement with the same eye flow vector (u,v) produces good result for all training samples that fall into that region.

We also investigate gaze redirection based on an *image-independent flow field* — a simple variant of our system, where each eye flow vector is independent on the test image content and is based solely on the relative position in the estimated bounding box. So, in the learning phase we consider all training examples for a given location (x,y) and find the offset minimizing the compatibility score. The two variants are compared in Figure 3.

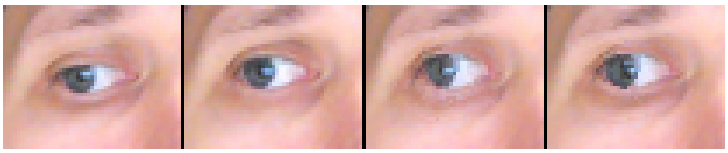


Figure 3: Redirection by 15 degrees. Left to right: the input, the "ground truth", the output of the eye flow forest, the output of the image-independent field.

[1] Antonio Criminisi, Jamie Shotton, Andrew Blake, and Philip HS Torr. Gaze manipulation for one-to-one teleconferencing. In *ICCV*, 2003.

[2] Claudia Kuster and et al. Gaze correction for home video conferencing. In *SIGGRAPH Asia*, 2012.

[3] Brian A Smith, Qi Yin, Steven K Feiner, and Shree K Nayar. Gaze locking: passive eye contact detection for human-object interaction. In *ACM Symposium on User interface Software and Technology*, 2013.