**LIBRARY FOR ACTIVE DISPLAYING THAT ADAPT TO USER POSITION AND OTHER EXTERNAL CONDITIONS**

**Motivation:** Now in the epoch when computer vision used more and more extensively and becomes a common part of user experience. This trend changes ordinary human Human-Computer Interaction. But usually, it requires special devices (new, highly specified and expensive ones) to use all bennefits of those technologies. But there are lots of “old” devices which perform their tasks good, and we are still capable of benefiting from them. I want to use several tactics to “augment” user experience from canonical and widely spread devises.

**Goal:** To change the User experience of interacting with old convenient display devices, when now they can become an active extension of our computerized systems. Usually, there are king of PC proxy or other operation modules who emits the signal to display. Therefore we can use those elements to combine those components into the loosely coupled system to improve the experience of using usual displays without a need to buy a whole new costly system. To make it possible we don’t need to make them “smart”, we can alter the video signal itself (a.e. rotate or augment the picture the picture and display it already rotated).

Steps of incremental development of this project:

**Phase 1)** Common Face/eyes detecting program working with web camera signal.

- Already done. Results are on project’s GitHub page. For now, we have MVP version of sight tracker; it gets the position of the point between the user's eyes as an orienteer for farther calculations.

Was used the approach of face cascade detection, - at first, we use face detector; then on it, we use eye-detector. This makes computation easier and eliminates the problem with false “eyes” on the picture.

**Phase 2)** Use this detector to triangulate the position of face regarding the display (the simple situation with a fastened web camera and display in one corpus). Modify approach used for mouse positioning with a gaze. Produce illusion of steady screen picture while moving head.

Currently in progress, must be done in a few days. To do it we’ll calculate in 3D sheer matrix which moves normal vector of our image plane to vector we have between user’s sight point and image center. Applying this transformation to our image in “pseudo-3D” space we’ll get resulting 2D projection of our image in the shape we want. (As invariant for the user). Also, we can apply resizing of an image.

**Phase 3)** Extrapolate those results to the external display and external webcam. Add triangulation of user’s position in a room, regarding screen and web camera. Here we move to the true 3D modeling of the working environment.

- On this stage, MVP will be ready, for improvement of framerate and processing time possible ports to library based on C++ optimized code.

- Try to use GPU for such processing.

Following phases can be changed:

**Phase 4)** Obtain similar effect for projecting over curved screens, geometric shapes, moving object, colored background. Here we want to use approaches of form extraction using canonical.

**Phase 5)** Use projector and camera as a “scanner” to acquire information about the shape of a surface to convert it to 3d.

**Ongoing vision of project development:** Further steps with 3d reconstruction... use approaches like space carving, shape detection from shading, etc too. Improve project quality.

**Tools:** OpenCV, Web-camera, RGB-D camera, External projector, external screen.

**Methods:** Pose/face/eyes detection, Image rectification, image transformation, triangulation of position in 3D space, multicamera vision, epipolar geometry, etc.

**Current project results:**

GITHUB: https://github.com/KushnirDmytro/Active\_displays\_project