Dmytro Kushnir CS@UCU

**USING CAMERA FOR ONLINE IMPROVING IMAGE QUALITY FROM PROJECTOR**

***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-Computer Interaction. But usually, it requires special devices to use all benefits of those technologies (as a rule: new, highly specified and expensive ones). But there are lots of “old” and cheaper devices which perform tasks they’ve been made for really good, and we are still capable of benefiting from them.*

*The user experience of interacting with old conventional display devices can be improved or "augmented" drastically; now they can become an active extension of our computerized systems, not only passive output devices. Usually, there is kind of PC proxy or other operation modules who sends 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).*

**GOAL:**

*In this particular work, we’ve implemented only part of the bigger idea of augmenting environment using a not bearable device (like glasses, etc), but a device which will make exact projections.*

*This part is projected picture refinement. As generally there is not something like a white screen or white-room in our environment, we should adapt usual projector to do be able to reproduce the image on surfaces with color, texture, shape, gloss and exterior luminosity. This project can be not only the part of a described system but can help in situations when you don’t have an expensive projector or ideal environment, but still, want to have best possible experience.*

*As the closest step from the future, this project can be used to extract shape and properties of a surface and use it to build the model of an environment.*

**IMPLEMENTATION**:

Components used:

1) Computing station;

2) Projector [MINI LED PROJECTOR “RD-802”, example ref: {<https://www.hibargain.com/rd-802-mini-led-projector-with-hdmi-av-vga-sd-usb-tv-home-theater.html>}]

3) Built-in 720p Web-camera for workstation HP 8770-W.

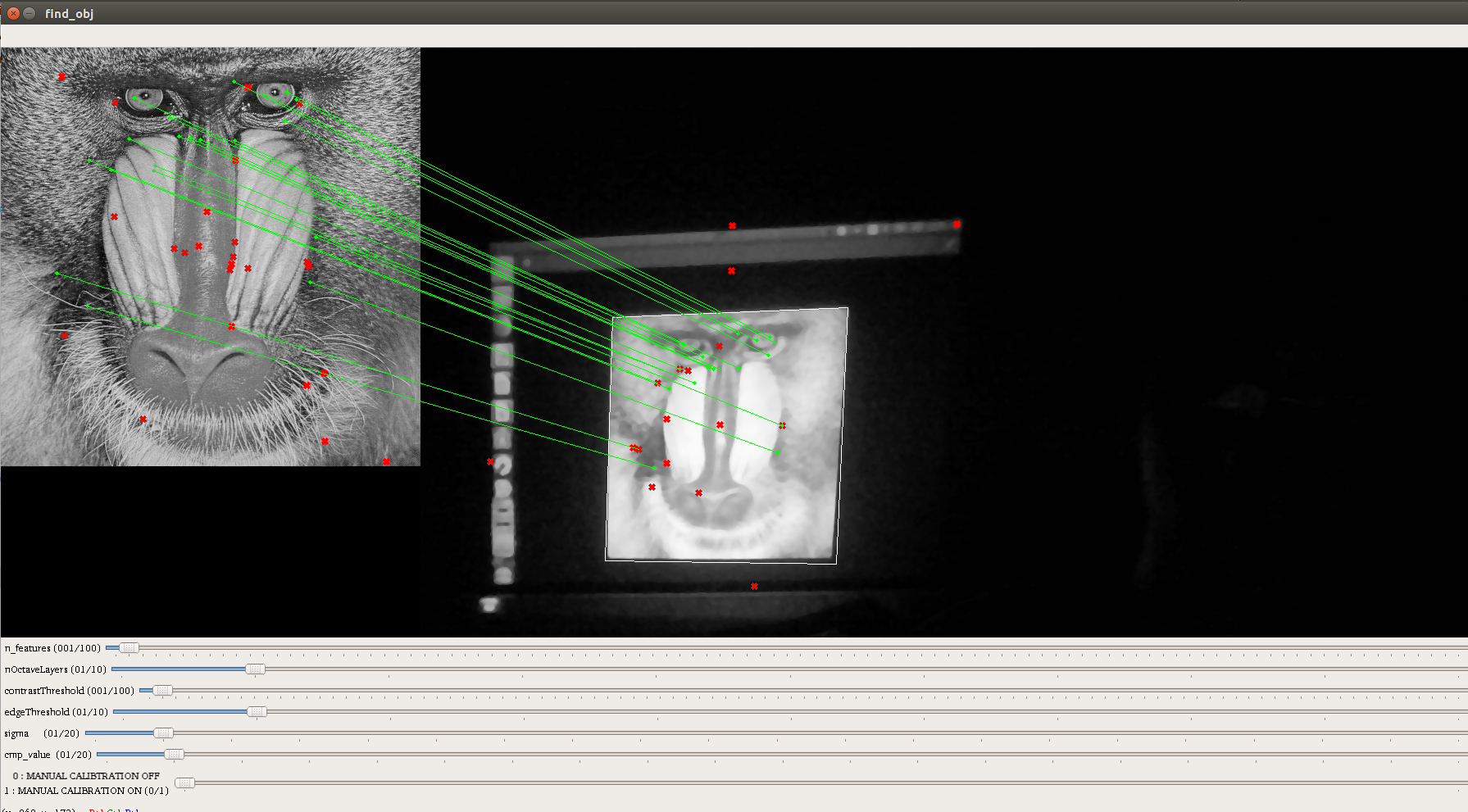
**In the code there been used many functions from OpenCV library. Some of them are used with a “contextual code” taken from openCV distributive demo samples.**

**SOLUTION:**

Detecting “the Region of interest” (ROI), as the projected image, corresponding to our original one. For this purpose we use features extractor for detecting image position works with the option of manual tuning (as we made this project aiming to work in unpleasant conditions. Therefore even state of art detectors is failing to work after some threshold of unpleasant conditions, as when the image becomes blurred too much due to focus breakdown or external light sources overwhelming low-power projector capabilities).

It can be done either in auto mode or manually adjust detector settings (as the condition of use is intended to be unpleasant, this tradeoff seems inevitable).

It worth mentioning that the same task can be done using "changing" picture to detect an altering region of the picture.

  
Illustration 1: SIFT detection with controls for paramaters

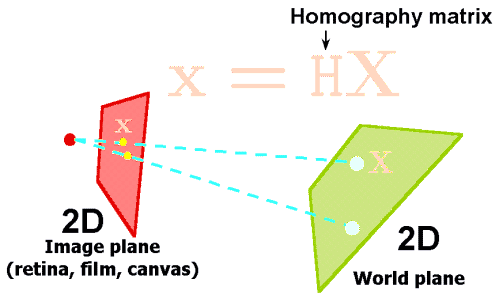
But we refused to use this approach as it has other tradeoffs:

1) recalibration interrupts ordinary usage process.

2) other moving or altering regions of the captured picture can spoil the results

Step 2) Calculating homography matrix between those images. As a result, obtaining “coupling” between elements of our system.

Eventually, this step became a part of final loop execution and helped to get consistent tracking of the image in case of devices movement.



**Step 3)** Creating learning *“feedback-loop”* and changing image being changed towards etalon image via our interpretation of linear approximation descent for our images. This performed starting from taking the usual difference between images. We hypothesize that such differencing will give us the color of the surface being projected to combined with distortions from devices. Testing had confirmed our hypothesis. On the image, we can observe this color (color of the wall in the testing environment).

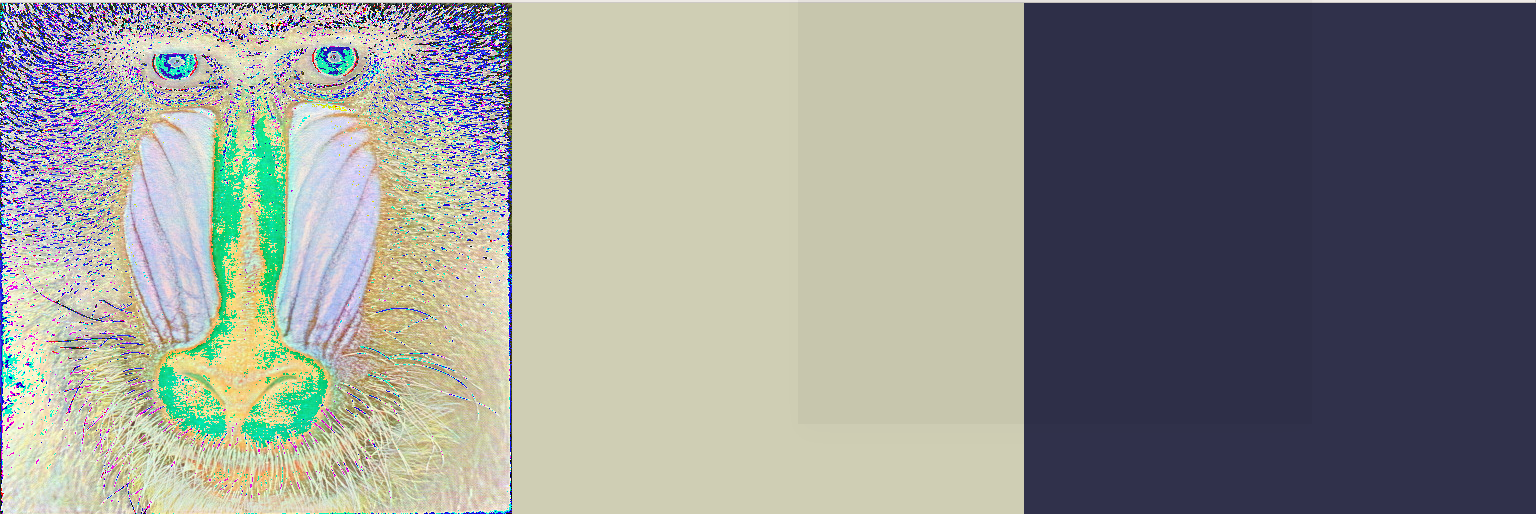
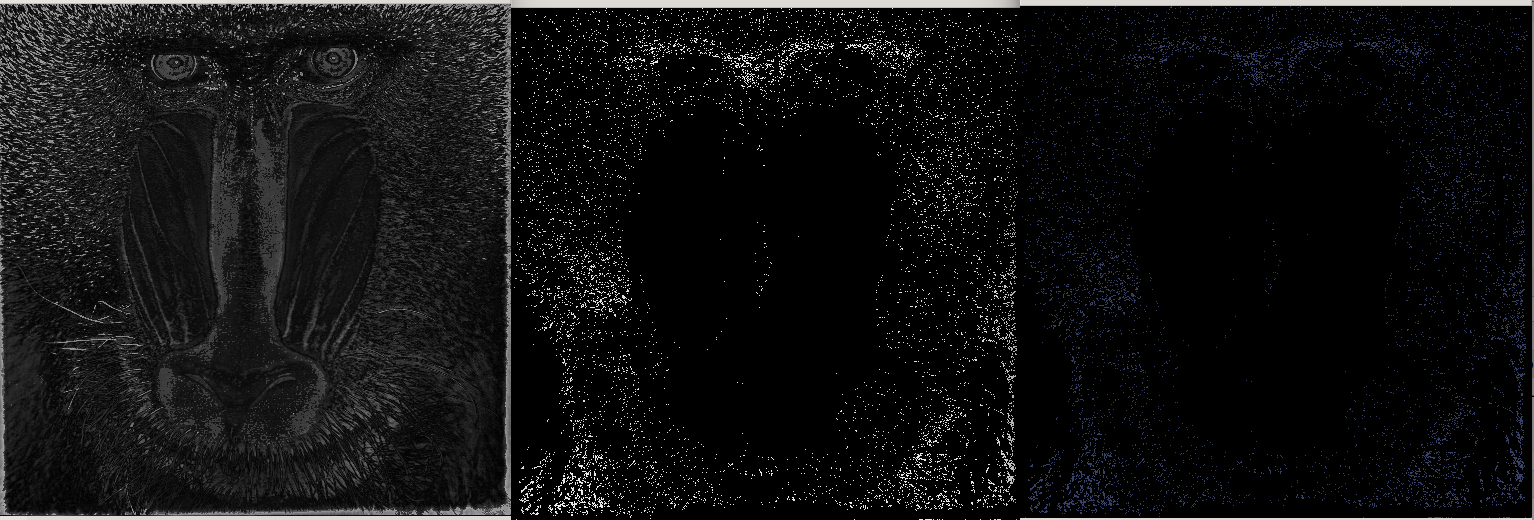
  
Illustration 3: difference, mean value of colour, inverted colour

  
Illustration 2: original, projected, difference

Then we use the function which makes adjustments to image, being now projected and reading it results from a surface. Adjusting the whole image is a bad idea because we took moda of our distribution, but this value not being observed from the whole image.

  
Illustration 4: distances to obtained moda colour, mask for points under distance treashhold, resulting applied colour-shifting mask

Therefore we define poins which are close enought to moda value, craeting mask from them and in this way filtering regions of adjustment.

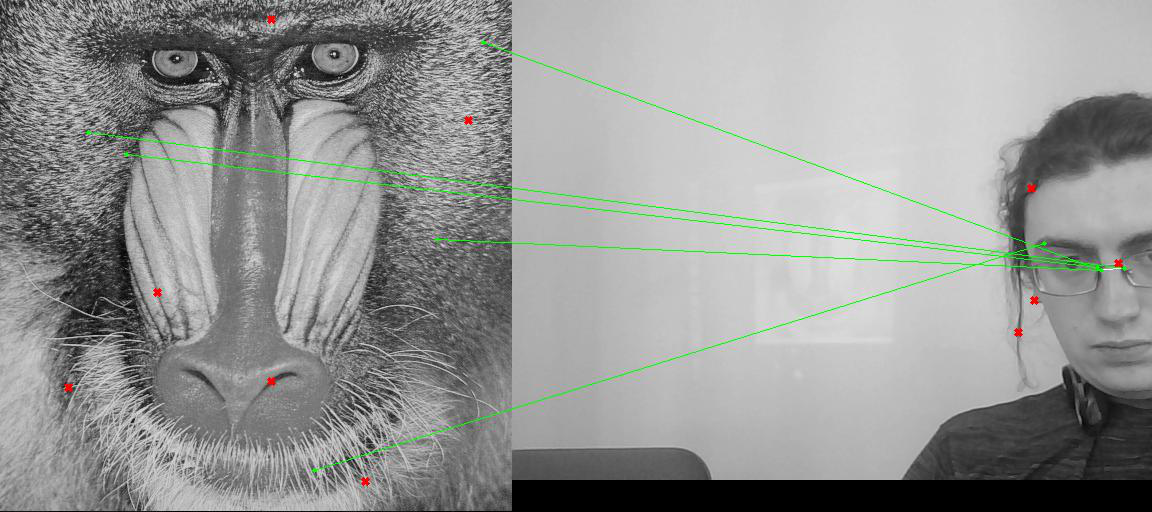
  
Illustration 5: Original, Adjusted, Image from camera

**Faced problems and perspectives**:

Worst in a project was a dealing with:

1) Technical issues with weak projector and camera. As a result, achieved only 5-7% of cost function decrease. Lots of approaches require better resolution.

2) Daylight, which made some development and testing days pure nightmare:

Perspectives:

Try to improve this approach with new equipment, and try the approach with “convolution-like” filters, applied to regions of the image to filter uneven surfaces.

After achieving

**Current project results:**

GITHUB: <https://github.com/KushnirDmytro/Active_displays_project/tree/master/ML_course_project_Projection_refinement>

General research results (branches in development or ones which met dead ends) :

GITHUB: <https://github.com/KushnirDmytro/Active_displays_project>