Final Project: Enhanced Augmented Reality

## Description

For this topic you will explore techniques for inserting images and video into a scene without markers. Your final algorithm must identify a clear surface in a video and project an advertisement onto that surface. The advertisement must remain in a fixed position in real-world coordinates as the camera is moved. Your algorithm must also handle the case where the advertisement location temporarily moves out of the camera’s field of view.  
  
For this topic only, your report must contain run-time benchmarking of your code and a discussion of the processing power required to run your code in real-time (30 frames / second) on HD video. Your code must be optimized to run in as close to real-time as possible. This replaces the requirement to benchmark your algorithms against a public imagery database.  
  
Related Lectures (not exhaustive): 3A-3D, 4A-4C, 6A-6B, 7A-7D

## Problem Overview

**Methods to be used:** Use methods to detect features (corners, lines) in images, these can be from OpenCV and use them to identify a flat surface in a scene. This surface will become the area where an image / video will be projected to simulate the presence of a flat object in the scene.

**RULES**:

* **Don’t use external libraries for core functionality**You are encouraged to use libraries while writing code for your final report. However you will receive a low score if the main functionality of your code is provided via an external library.
* **Don’t copy code from the internet**The course honor code is still in effect during the final project. All of the code you submit must be your own. You may consult tutorials for libraries you are unfamiliar with, but your final project submission must be your own work. Any instance that does not follow the Honor Code and the class rules will be directly reported to the Office of Student Integrity.
* **Don’t use pre-trained machine learning models**If you choose a topic that requires the use of machine learning techniques, you are expected to do your own training. Downloading and submitting a pre-trained model is not acceptable for any project topic.
* **Don’t rely on a single source**We want to see that you performed research on your chosen topic and incorporated ideas from multiple sources in your final results. Your project must not be based on a single research paper and definitely must not be based on a single online tutorial.

**Please do not use absolute paths in your submission code. All paths must be relative to the submission directory. Any submissions with absolute paths are in danger of receiving a penalty!**

## Programming Instructions

You may use the python 3 environment provided for the project. This new environment is simply a list of the versions of libraries that will be used during grading. You may install them however you wish. We recommend conda. Include a README.md file with usage instructions that are clear for the grader to run your code. Remember to specify what version of python you are using. Notice that despite having Tensorflow and Pytorch in the environment, you are not allowed to use them.

Windows Users Warning:

Be warned that TA’s grade exclusively on linux machines. Thus, it is your responsibility to make sure that your code is platform independent. This is particularly important when using paths to files. If your code doesn’t run during grading due to some incompatibility you will incur a heavy penalty.

## Write-up instructions

The report must be a PDF of 3-6 pages including images and references. Not following this requirement will incur in a significant penalty and the content will be graded up to page 6. **Note that the report will be graded subject to a working code.** There will be no report templates provided with the project materials.

The report must contain:

1. A clear and concise description of the algorithms you implemented. This description must include references to recently published computer vision research and show a deep understanding of your chosen topic.
2. Results from applying your algorithm to images or video. Both positive and negative results must be shown in the report and you must explain why your algorithm works on some images, but not others.

You report must be written to show off your work and demonstrate a deep understanding or your chosen topic. The discussion in your report must be technical and quantitative wherever possible.

## How to submit

Similar to the class assignments, **you will submit the code and the report to Gradescope (note: there will be no autograder part).** Find the appropriate project and make your submission into the correct project.

**Important: Submissions sent to Email, Piazza or anything that is not Gradescope will not be graded.**

## Grading

The report will be graded following the scheme below:

* Code (30%): We will verify that the methods and rules indicated above have been followed.
* Report (70%): Subject to a working code.
  + Description of existing methods published in recent computer vision research.
  + Description of the method you implemented.
  + Results obtained from applying your algorithms to images or videos.
  + Analysis on why your method works on some images and not on others.
  + Proposals on how your methods can be improved.
  + References and citations.

## Assignment Overview

This project can be seen as an extension of Problem Set 3. The main task is to **detect a flat surface** in a scene and project an image on it **without the use of predefined markers**. A flat surface can be found in common large items such as walls, tables, doors, etc. As part of the process, your method must use existing features in a scene as anchor points for the projected image. Placing markers in a scene will not be accepted. Determining which these can be is part of your research and project requirements. To find interest points, you could use any of the distance transform methods.

### 1. AR Requirements

The projected image must behave similar to an object present in the scene. This means that your approach must be able to handle cases where part of the projected image is outside the frame boundaries while keeping the remaining part in the scene. If the camera view leaves the area where the projected image is, the projected image will not show in the current scene. However, the object must show again if the camera moves back to a view where the object should be visible. One way to address this is to track the image / camera motion using features to track and use this information to “guess” where these are outside the image frame. This approach must be tested using these cases as part of the project requirements.

### 2. Input Requirements

This method must be tested in videos with scenes that present multiple points of view. This means the videos you select will present changes in rotation and perspective. As stated above, one of your test cases must contain an instance of the projected image being partially visible, outside the scene (not visible), and visible after the camera field of view returns to the flat surface. Additionally, at least three different scenes must be used to demonstrate your method’s robustness. The projected image can be the same in all videos.

**Important:** The projected image must remain in place during the entire sequence. Flickering effects in the projected image will not qualify as good results.

### 3. Final Results

#### 3.1 Video Results

Video with the target image projected in the scene. Include the link to this video in your report and in the presentation. This means you will provide two video links: one showing the target image projection (similar to the output videos in PS3) and one for the video presentation.

#### 3.2 Image results

Include sample images in your report that showcase your method’s performance. Make sure you present all cases mentioned in the project instructions.