

# CIS 581–Computer Vision–Fall 2015

## Final Project: Face Replacement

Checkpoint: December 3  
Presentation: December 11 11am-3pm  
Code Due: December 17 11:59pm

### Overview

The goal of this project is automatic detection and replacement of faces in videos. Given a test video, you will automatically replace a target face. Ideally, you should do this in the most stable way possible.

Seamlessly face replacement is a non-trivial process. For the rest of this document, we shall use the following terminology: face(s) in the video = target face(s), face(s) used for replacement = replacement face(s). First, you will build a model of the replacement face. You can use a single, well aligned photo, or a sequence under different poses. You will have to devise a scheme to be able to detect and localize similar looking faces in video frames. One way to do this is with feature detection, matching, and voting. Once you detect a face in a test image, you must estimate the warping between the candidate replacement face(s) and the detected candidate target face(s). We could suggest extending previously explored concepts such as simple affine, warp, or you can allow additional deformity using TPS. You are of course not limited to these and are encouraged to experiment with any other suitable approaches you may find. If you are using multiple faces for training, you should match against all of them and use the one that requires the least morphing.

Appearance matching may be necessary to compensate for shadows, lighting and skin tone. The target face will need to be cropped out, and a morph of the appropriate replacement should be inserted in its place, and the final video frame blended seamlessly. When dealing with videos, you might expect the faces in the video to move. For extra credit, you are also encouraged to explore methods for motion compensation.

We highly recommend you do this project with a partner. You may leverage code from previous projects.

## Concentration Options

We provide you with options for this relatively open-ended project. You can either go deep into certain topics, or use 3rd party libraries to produce full face replacement result.

**Option 1** Focus on a specific topic related to face replacement (eg, SIFT, HOG, etc.) and implement it on your own, apply it to the face replacement project, and analyze the result. For this option, you don't necessarily have to achieve a full face replacement result on other parts (such as blending). But we expect you to go deep into the certain algorithm. You will need to compare your results with one or more 3rd party implementation - you are welcome to seek advice on potential packages to use. *Reports will be expected from teams choosing this option.*

**Option 2** You are allowed to use standalone 3rd party libraries (that can be compiled directly on Windows 7/Linux/OS X) to achieve satisfactory overall results for face replacement. You do not need to limit yourself to MATLAB, but please provide comprehensive instructions on how to use your code/program. You will be facing challenges on exposure, multiple face instances, face distortion, etc. We will evaluate your performance based on overall visual results. *Presentations on the 11th will be compulsory for teams choosing this option.*

Please specify your option at your checkpoint, presentation, and submission. You are welcome to consult with us before starting the project.

## Procedure

Included on the course webpage will be links to download video clips that are of easy/medium/hard difficulty classifications. You can use these to develop and test your code. Four days before the project presentations, we will release new test videos and your code will be expected to work on these too. You will be graded based on the testing videos. Ideally, changing your code to work with the new set should not take much effort, but the 4 days should allow you some time for tweaking.

Again, you are encouraged to work with a partner on this project.

## Schedule

- **Dec 3: Checkpoint** We expect some progress to be made on face detection at this point. You need to submit 3 slides in PDF form to the appropriate canvas submission portal.

- **Dec 7: Official Test Set** The official test set will be released. It contains both easy and very difficult cases. Please run your code on this for your presentation.
- **Dec 11: Presentations (compulsory for option 2)** Each team doing option 2 will present the results between 11am-3pm on December 11th. The presentation format will be the same as previous projects: 5 minutes/5 slides. The time limit will be enforced so we can get through everyone. Teams doing option 1 are also welcome to present if they would like to share their work/results
- **Dec 17: Code Due** Submit code to canvas with additional information on anything we should know to compile/run your code. Include a **script** that runs your face replacement code on videos included in the official test set. Also include your **results** in a suitable format, eg processed pictures, or video with replaced faces; if you concentrated on Option 1 such that you did not do the complete replacement, perhaps a PDF with your in-depth algorithm is more appropriate.

## Tasks

Specific tasks are as follows.

1. **Replacement Face(s) Selection** Select the face(s) you will be inserting over the test sets.
2. **Target Face Detection** Attempt to locate instance(s) of faces in the test set using an automated detector. The lectures provide some details on HoG, Shape Context, Chamfer Matching, Harris Corners, SIFT features, Pictorial Structure, etc. that may be useful here. You may use 3rd party libraries (Option 2) or implement it yourself (Option 1). You may either target the face that looks most like your own, or target all faces in each test image.
3. **Face Deformation** For each instance of the target faces, find a deformable image transform that warps the reference face to the detected instance. Apply this transform to the replacement face. An affine warp or a TPS warp with low number of keypoints may be appropriate.
4. **Face Replacement** Compute the convex hull of the detected face and the warped replacement face. Replace the convex hull of the detected face with the convex hull of the warped replacement face in the test image.
5. **Refinement (compulsory for Option 2, extra credit for Option 1)**  
Here are some directions to make the replacement more natural.
  - (a) Use Laplacian image blending/Gradient domain blending to get a seamless integration of the new face.

- (b) Find the appearance difference of the detected face to the warped version of the detected face. This could be a per-pixel additive offset to compensate for shadowing and lighting of the target face. Apply this brightness/color offset to the warped replacement face to achieve a better appearance match.
6. **(Optional) Spiffify** Eternal glory (extra credit) will be awarded for teams that implement features such as motion compensation and any other cool features you can think of. This is a VERY open ended project, so go nuts and have fun!

## Scoring

- To receive full credit, your code must perform well on the easy videos in the test set by your presentation time.
- The optional tasks and successfully replacing faces in the difficult set will receive extra credit.
- Replacing the faces in the difficult set successfully will also compensate for not having working code at the presentation time, but this will be difficult and is not a recommended alternative to finishing before the presentation.
- And of course, as mentioned props (credit) for cool extra features

## FAQ

### Can I use late days?

Late days may be used for the final code submission, but not the checkpoint or presentation; it is ok if your preliminary results/presentation are not very good if your final submission is good. If you and your partner have different number of late days left, you can use the average.

### Can I reuse code from previous projects?

Yes, you are encouraged to do so.

### Changing project option

You may change your option any number of times before the checkpoint - as you're not expected to declare it until then. Any changes after the checkpoint will be considered on a case-by-case basis.