

CIS 581–Computer Vision–Fall 2014

Final Project: Face Replacement

Checkpoint: December 9
Presentation: December 19 6-8pm
Code Due: December 20 11:59pm

Overview

The goal of this project is automatic detection and replacement of faces in images (and videos). Given a test image, you will automatically replace the face that most closely matches yours with your own. For an additional optional challenge, you may attempt to do this in a short video sequence.

Automatic face replacement with seamlessly rendered replacement is a non-trivial process. First, you will build a model of your face. You can use a single, well aligned photo of your face, or a sequence under different poses. You will have to devise a scheme to be able to detect and localize similar looking faces in images or 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 your face and the detected face. You may use a simple, eg affine, warp, or you can allow additional deformity using TPS. If you are using multiple faces for training, you should match against all of them and use the one that requires the least morphing. Next, the morphed replacement face must be appearance matched to the detected face to compensate for shadowing and lighting, the detected face cropped out, and your face inserted, and the final image/video frame blended seamlessly.

We highly recommend you do this project with a partner. You may leverage code from previous projects and also third party tools.

Concentration Options

This year, 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 certain topic (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. We will compare your results with the 3rd party implementation.

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 will be facing challenges on exposure, multiple face instances, face distortion, etc. We will evaluate your performance based on overall visual results.

Please specify your option in your checkpoint, presentation, and submission. You are welcome to come talk about it with us at the beginning of the project.

Procedure

Included on the course webpage are two practice sets—one easy and one hard—containing some images with faces to replace. You can use these to develop your code. Four days before the project presentations, we will release a new test set containing a different set of images with faces to replace. Those are the images you will be graded on. 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. Your final code will be due on the day after the presentations at midnight.

You are encouraged to work with a partner on this project.

Schedule

- **Dec 9: Checkpoint** We expect some progress to be made on face detection at this point. You need to submit 3 slides in PDF form to cis581.Fall2014@gmail.com showing your progress. Please zip your slides as `4_pennid1_pennid2_cp.zip`, and email with the subject “*Project 4 checkpoint*”.
- **Dec 15: 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 19: Presentations** Each team will present the results between 6-8pm on December 19th. 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.
- **Dec 20: Code Due** E-mail your final assignment files to in a ZIP file named `4_pennid1_pennid2_code.zip` to cis581.Fall2014@gmail.com with the subject “*Project 4 code*”. Include a **readme.txt** file containing your name(s), and anything we should know to compile/run your code. Include a **script** that runs your replacement code on each of the official test set files. 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 will be awarded for teams that implement detection and replacement of their face in a short video sequence from a TV or movie scene. You should try to ensure smooth tracking of replaced faces over sequential video frames.

Scoring

- To receive full credit, your code must perform well on the easy images 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.

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.

What if I already made plans to leave/go home before the 19th?

If neither you nor your partner can make it to the presentation, you may instead submit a 5 minute video presentation at least 24 hours before the regular presentation time. This is to ensure we have enough time to work out any potential technical difficulties so that we can show your presentation to the class without problems. Name it `4_pennid1_pennid2_pres.<mp4|mov|avi|...>` (choose an appropriate video file extension), and send it in an email to `cis581.Fall2014@gmail.com` using the subject “*Project 4 video presentation*”. The rest of the project requirements and deadlines remains the same.

I/we tried to email the files, but GMail rejected the ZIP file. How can I/we submit the project?

You can share it via any cloud service. Or you can change the extension of the .zip to .piz, and letting us know to rename it back on our end.