CS 763/CS 764: Lab 02

Your Zoom background image

- Announced 24/01. Due 28/01 23:29
- Please write (only if true) the honor code. You can find the honor code on the web page. If you used any source (person or thing) explicitly state it.
- **This is NOT an individual assignment**

1 Geometric Transformations

In the lectures, we saw how an app such as Camscanner or Adobe Scan or Dropbox undistorts an image. In this lab, we go the other way.

1.1 Perspective Transformations

In the current time of attending online video calls via Google Meet, Zoom, or MSTeams, many of you would have tried the background change feature in order to avoid showing cluttered background of your room or for some other reason. Designing such feature requires some sophisticated methods and are beyond the scope of this lab.
In this part, we will tackle a simpler task, namely, overlaying a sample image such as Fig. 1d on the frame present in Fig. 1a as seen in Fig. 1b using concepts from planar homography. As discussed in the class, corresponding points in two images are related by a homography matrix H.
In this lab, you will manually find the coordinates of corresponding points images. Finding the homography automatically will be a future topic. **Note:** You can use `imshow` with `matplotlib` (for example) to get the coordinates of the point in the image corresponding to the cursor.

(a)

(b)

(c)


(d)

Tasks

1. You are provided four images similar to Fig. 1a and Fig. 1c, The first three images consist of a painting hanging on a wall, while the fourth image is, well you know this, right?

   (a) Project the image shown in Fig. 1d on the frame PQRS in each of the three provided images `{1,2,3}.jpg` so that the result is natural (remember, the zoom audience?). Use four points and `cv2.getPerspectiveTransform()`. Note that the letters PQRS are indicative, use your own judgment. Intention for running:

```
$ python3 image_perspect.py
```

(b) Now, instead of just projecting a single image, project the video v (viz. `test.mp4`) provided in the data folder. On running your program, a window should pop-up displaying v but projected onto the PQRS frame of 1a similar to part (a) of this question. Try adjusting your code to get a meaningful frame-rate, but there will be some overhead which will cause a slower video, but that's okay for this task.

```
$ python3 video_perspect.py
```

(c) This is going to look like a mind experiment. Assume the group consists of students α, β and γ. The instructor has access to `arch.jpg` and `1.jpg` and is able to nicely do the desired mapping shown in Fig. 1b.

However, α has access only to images `{1,2}.jpg`. β has access only to images `{2,3}.jpg`. γ has access only to `arch.jpg` and `3.jpg`.

How does γ emulate the instructor? In this mind experiment, γ can only call functions written by the other two. He cannot pool resources such as images from the other two.

To concretize the notion of images being emulated, your program should output the (`rmse`) relative mean squared difference between the images averaged over the pixels. But wait... Is averaging over all pixels a correct evaluation method? Describe a suitable way for calculating the `rmse` value in `answersB.pdf`.

```
$ python3 sequential.py   # produce the output
$ python3 compare.py  # comparison, print rmse code (?.xx)
```

(d) Similar to `getPerspectiveTransform()`, `getHomography()` takes arrays with more than 4 corresponding points. Now, repeat part (c) using 16 corresponding points rather than 4 and report the `rmse`. Compare with the value obtained in part (c) and provide justification for the result in `answersB.pdf`.

(e) Repeat tasks 1(a) and 1(b) using your own images and a video with a different encoding format. You can capture three images of a planar surface from three different viewpoints such as the ones provided. Be creative about the fourth image and possibly the video (but be aware you shouldn't put your stuff on the Internet due to copyright violations).

Questions Note that these are indicative questions. Any implicit questions that you faced should be mentioned here and properly highlighted. In each case a paragraph of explanation should suffice.

1. What linear algebra technique is used in 1(a)? Explain.
2. What attempts did you make to get a meaningful frame-rate? Explain.
3. How did γ achieve the task? Explain.
4. Explain your understanding of `rmse` computation over the domain of choice.
5. Is (d) same as (c)? Explain.
6. Why did you choose the images that you did for (e)?
7. What codec did you use in (e)?
8. (Optional) What is missing in Fig. 1d and how is that relevant to "students and testtakers"?

As you can see most (but not all) questions have an "explain briefly" in them. That is the intent of these questions, so please explain when appropriate and do not wait to be asked.

Submission Guidelines

Submission guidelines generally remain the same. However there are some parts that can change, so be flexible.
Your submission folder should look something like:

```
130010009_140076001_150050001_Task02/
├── ReflectionEssay.pdf
├── answersB.pdf
├── answersA.pdf
├── code
│   ├── affine-trans.py
│   ├── compare.py
│   ├── image_perspect.py
│   ├── video_perspect.py
│   └── sequential.py
├── data
├── results
├── convincingDirectory
└── readme.txt
```