16-822: Geometry-based Methods in Vision (F17) Released: Nov-15., Due: Dec 06

Homework 4

Lecturer: Martial Hebert *TA: Aayush Bansal*

1 Exploring 3D Reconstruction! (60 Points)

In this part of the homework, we will explore 3D reconstruction. Especially, we will study a full pipeline of a typical 3D reconstruction system using Structure from Motion (SfM) and Stereo Matching. Slightly different from the previous homeworks, this time you do not implement everything from scratch. Instead, you will delve into one of the existing packages for this purpose (Here we use the *SFMedu* system from the computer vision group at Princeton University) by analyzing the results and modifying some steps of particular interest and importance. This is also a very good way to understand the material taught in the class.

1.1 What you are supposed to do:

You need to both implement and answer questions as follows:

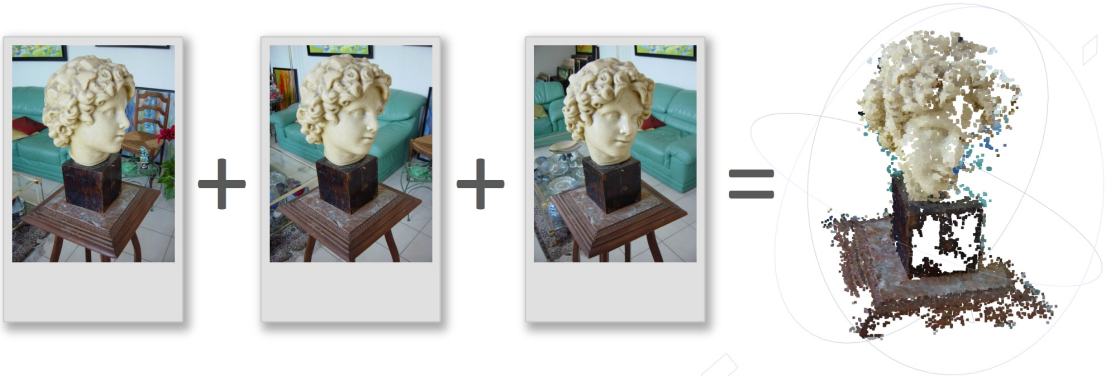

Figure 1: An illustration of the SFMedu system: Structure from Motion (SfM) and Stereo Reconstruction.

- **Run the SFMedu system**
  We have included the SFMedu package in `./code/` folder. Run the SFMedu system by running the `SFMedu2.m` script in MATLAB. Make sure that it runs successfully and produces a `dense.ply` point cloud file. Install MeshLab from `http://meshlab.sourceforge.net`, and open the point cloud in MeshLab and rotate around the model. Take a screen capture of the point cloud and put it in the report as the answer to this question. An illustration is shown in Figure 1.
- **List the Major Steps**
  Read the code carefully, especially `SFMedu2.m`. Summarize the major steps for SfM and dense stereo matching, and write it down in the report as answer to this questions.

- **Principal Point**
  What is the assumption for the principal points in this system? Under what condition will this assumption get violated?

- **Reconstruct Your Own Images**
  Capture/Find two sequences of images (at least 5 images per set) on some interesting objects or scenes, run the system on these images. If it fails, figure out why it fails, and change the code or retake good images until it works. Submit the input images, as well as three screen captures of your reconstruction per image set in the report. Explain all the changes or things you have to do to make it work. Which step do you think is the most unstable one? Explain why. List suggestion for a normal user without any computer vision background about how to take good images to make it work.

- **Compute the Reprojection Errors**
  Write a function

  ```
  function printReprojectionError(graph)
  ```

  that takes a graph (as defined in the SFMedu system), and prints out the current reprojection error, so that you can insert the function into many steps for debugging purpose, to check if the reconstruction errors are getting smaller. A call to this function is at Line 160 of `SFMedu2.m`. Submit the function as code. Also, plot the reprojection errors in the report for the image sequence provided and the two sets of images you captured in the report PDF file.

- **Visualize the Reprojection Points**
  Write a function

  ```
  function visualizeReprojection(graph,frames)
  ```

  that takes a graph (as defined in the SFMedu system), and draw the 3D keypoint point cloud projected onto each image, as well as their observed location. Figure 2 shows an example of the output for your function. You are required to write a function to produce the same kind of visualization. Each observed keypoint is represented by a red ×, and the reprojection of its 3D estimated location is represented by a green +, and these two points are connected by a blue line. For the 3D points that are projected to the image but not observed from the image, it should be shown as a yellow ○. A call to this function is at Line 162 of `SFMedu2.m`. Submit the function as code, and also include the visualization results of the image sequence provided and the two sets of images you captured in the report PDF file.

- **Levenberg-Marquardt**
  `bundleAdjustment.m` uses matlab function `lsqnonlin` to minimize the objective function via the Levenberg-Marquardt algorithm:

  ```
  [vec,resnorm,residuals,exitflag] =
  lsqnonlin(@(x) reprojectionResidual(graph.ObsIdx,graph.ObsVal,px,py,f,x),
  [Mot(:);Str(:)], [], [], options);
  ```

  Please read all the code and figure out what is the objective function it is optimizing for. Write down the math equation of this objective function in your report.

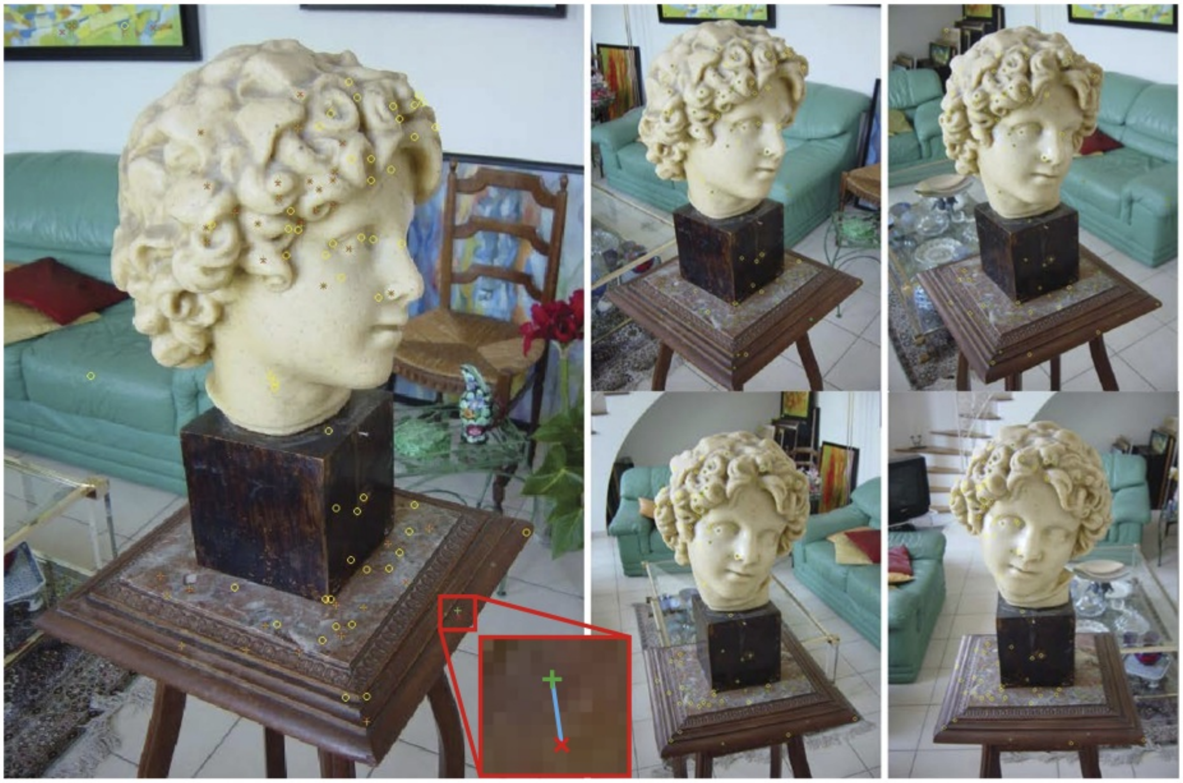

Figure 2: An example result produced by `function visualizeReprojection(graph,frames)`.

The Matlab implementation lsqnonlin of Levenberg-Marquardt is not suitable for very large scale problem. Read the document of lsqnonlin and explain why this statement is true in your report.

- **Motion Adjustment Only**
  Bundle adjustment typically optimizes the Structure (the 3D location of the points) and the Motion (the camera parameters) together, as the above lines of code. But we can also fix the Structure, to adjust the Motion only, i.e., not allow the 3D points to move, but only changing the rotaiton and location of the cameras to optimize the objective function. With some good initialization, this can be used as a fine adjustment to solve camera resectioning task as well (`http://en.wikipedia.org/wiki/Camera_resectioning`). Fill in Line 34 in `bundleAdjustment.m`.

  Please write down the math equation for the objective function, and also change the above line of code to do this. Submit the code and illustrate the results.

- **Structure Adjustment Only**
  In the same way of thinking, we can also fix the Motion, to adjust Structure only, i.e., not allow the cameras to move, but only allows the 3D points to change their locations. Similarly, with some good initialization, this can be used as a fine adjustment to do “triangulation” for reconstructing the 3D point locations. Fill in Line 30 in `bundleAdjustment.m`.

Please write down the math equation for the objective function, and also change the above line of code to do this. Submit the code and illustrate the results.

Although this can be used to do triangulation, there is certain drawback. What is that? Answer this in the report.

- **Smarter Graph Merging**
  The current reconstruction is merging the sequence sequentially, by adding one camera into the `mergedGraph` in each step and do bundle adjustment again. A better graph merging will help to make the system more robust. For example, we can merge the graphs that have the maximal number of keypoint correspondences, instead of sequentially picking one. Change the code to implement a smarter graph merging. Submit the code and illustrate the results.

- **Intrinsics**
  Change the code to optimize for the full intrinsics matrix. Submit the code and illustrate your results.

1.2 What you have to submit:

You should submit three files:

- A report that contains all the images (both the images you captured and your result images) and answers to the questions. (**25 Points**)
- Your **code** folder that contains all source code (including **your** readme) for your system, and your *SFMedu2.m* script that takes no parameter as input and runs directly in Matlab to generate the results reported in your PDF file. (**25 Points**)
- Your **image** folder that contains all the input, intermediate and output images. (**10 Points**)

2 Geometric cues from Single Image (60 Points)

Learning-based approaches have become popular to estimate geometric properties from single image. Here we will reason about the recent formulation for the problem of estimating depth and surface normal from single image. We will first explore how to formulate these problems, what are the problems with the current formulation, and how could we improve these formulation.

2.1 Depth estimation from a single image (40 Points)

Estimating depth from a single image can be formulated as a regression problem, where you want to estimate the depth at each pixel. This problem has therefore largely been treated as a learning problem when a single image is provided. A lot of training data consisting of images and corresponding depth maps (obtained using Kinect or other 3D sensors) is used to train a model. At test time, this trained model is used to do inference over test images (generally not seen earlier, but roughly belonging to the same settings). This formulation however undermines the knowledge of geometry that we have learnt so far. In this section, our objective is to first understand this formulation, then figure out what is the problem with it, and finally propose new methods from our understanding in the geometry class that could potentially improve these approaches. Other than Hartley & Zisserman book, you may want to check out the following papers to answer the questions in this section.

1. D. Eigen, and R. Fergus. *Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture.* In ICCV 2015. `https://cs.nyu.edu/~deigen/dnl/dnl_iccv15.pdf`.

2. I. Laina, C. Rupprecht, V. Belaigiannis, F. Tombari, and N. Navab. *Deeper Depth Prediction with Fully Convolutional Residual Networks.* In 3DV 2016. `https://arxiv.org/pdf/1606.00373.pdf`.

3. A. Chakrabarti, J. Shao, and G. Shakhnarovich. *Depth from single image by harmonizing overcomplete local network predictions.* In NIPS 2016. `http://www.cse.wustl.edu/~ayan/mdepth/`.

Lets' take a **geometric** perspective of this problem. (**10 Points**).

1. Geometrically, what is wrong with the problem of single image depth estimation?

2. Reason why we cannot get absolute depth from single image, and can only know the relative position in the scene.

We will now explore the **formulation** of this problem. (**10 points**)

1. As mentioned earlier, depth estimation from a single image can be formulated as a simple regression problem. Write mathematically the simplest formulation you could think of.

2. What are the problems with this formulation?

3. Propose a new formulation that could potentially overcome the above problems.

4. What are the problems that the proposed formulation can still not overcome, and why you think there is no way you could have incorporated required constraints in the above formulation to address these issues?

Finally, about adding extra constraints from multiple cameras or motion cues; using the output reliably, and how to evaluate our formulation. (**20 points**)

1. Suppose you have multiple views available, how will you modify the above formulation so that you can get the absolute scale? Recall that to get the depth from a stereo pair requires the knowledge of baseline.

2. Treating single image depth estimation as a regression problem can only provide the relative "depth" values at test time. In class, Martial stressed the importance of knowing the confidence on the prediction/results so that it can be used reliably. Intuitively suggest how one could get the confidence scores from these systems OR how should one modify the current formulation to get the confidence score along with the predicted values.

3. What are the different criteria proposed in the literature to evaluate the output from a trained model? What is it that each one of these criterion can tell us about, and what it could not?

2.2 Surface normal estimation from a single image (20 Points)

Somewhat similar is the formulation for estimating surface normal from a single image. You may want to check out the following papers to answer the question in this section.

1. D. Fouhey, A. Gupta, and M. Hebert. *Data-Driven 3D Primitives for Single Image Understanding.* In ICCV 2013. `https://people.eecs.berkeley.edu/~dfouhey/2013/3dp/index.html`
2. X. Wang, D. Fouhey, and A. Gupta. *Designing Deep Networks for Surface Normal Estimation.* In CVPR 2015. `https://people.eecs.berkeley.edu/~dfouhey/2015/deep3d/deep3d.pdf`.
3. A. Bansal, B. Russell, and A. Gupta. *Marr Revisited: 2D-3D Alignment via Surface Normal Prediction.* In CVPR 2016. `http://www.cs.cmu.edu/~aayushb/marrRevisited/`.

Lets' formulate this problem, and see how can we improve the formulation with our understanding from the class. (**20 points**)

1. Write a simple mathematical formulation for this problem?
2. What are the problems with this formulation?
3. Propose a new formulation that could potentially overcome the above problems.
4. Reason why you think the above formulation is sufficient to get reliable estimates. Is there something you think that could not be captured by your proposed formulation? Why you think it could not be incorporated in above formulation?
5. What are the different criteria proposed in the literature to evaluate the output from a trained model? What is it that each one of these criterion can tell us about, and what it could not?

One last word: Feel free to drop an email to the TA in case you enjoyed working on this problem set (**Q2**), and you think that you want to experiment this formulation and see how it improves over state-of-the-art performance!