Fast Panorama Stitching on Mobile Devices

Digital Image Processing - Project

Additya Popli - 20161215
Devansh Gautam - 20161171

Project Overview

The aim of the project is to create high-resolution and high-quality panoramic images on mobile phones, so that a user can capture an image sequence of a wide range of scenes with a camera phone and see a panoramic image created immediately on the phone.

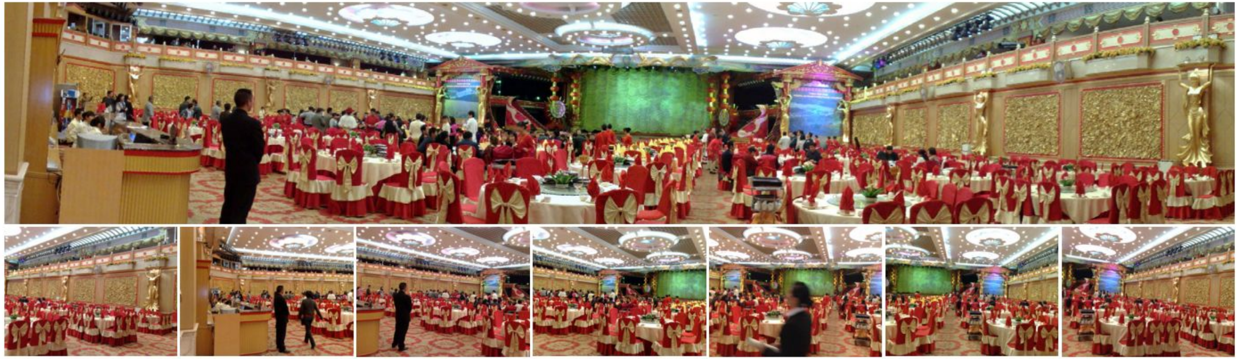


Fig. 12. Panoramic image produced by the fast panorama stitching with 7 source images in an indoor scene with moving objects.

Major Tasks

- Preparing the dataset

  The paper assumes all the images to be stitched are given along with their corresponding offsets from the first image.

- Optimal Seam Finding

  An error surface is constructed with squared differences between overlapping images. A low-cost path is found through the error surface by dynamic programming and used as an optimal seam to create labeling and the overlapping images are merged together along the optimal seam.

- Colour Correction

  All the images to be stitched together have different lighting which needs to be fixed and the images have to be blended properly so that the stitching seam it not visible and it seems like a continuous image.

Tasks Completed

- Preparing the dataset
- Optimal Seam Finding

An error surface is constructed with squared differences between overlapping images. A low-cost path is found through the error surface by dynamic programming and used as an optimal seam to create labeling and the overlapping images are merged together along the optimal seam.

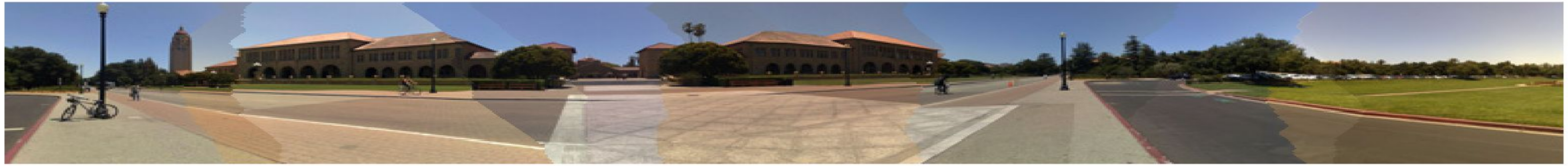
Suppose that *abcd* is the overlapping area between the current composite image I_c and the current source image S_c.

I_i^o and S_i^o are the overlapping images in the area *abcd* of I_c and S_c respectively. We compute squared differences *error* between I_i^o and S_i^o as an error surface, where ***error*** $= (I_c^o - S_c^o)$.

We apply dynamic programming to find a minimal cost path through this surface. The optimal path can be then be traced back to find the actual seam.

Output

This is the output we get without color correction using only optimal seam finding.



Tasks Left

- Colour Correction and Blending
- Integration with the current code