15-463: Computational Photography

Final Project: View Morphing

Authors

Eugenio Dominguez edomingu@andrew.cmu.edu
Anna Michalska amichals@andrew.cmu.edu

Overview

For our final project we have decided to implement the View Morphing paper by Steven M. Seitz and Charles R. Dyer.

Motivation

Standard morphing techniques do not perform well if we apply them to different views of an object, producing unnatural results that generally do not maintain the 3D structure of the object we are morphing. To overcome this we can use the view morphing approach that interpolates a fluid camera motion between different views of a scene.

Project details

The view morphing algorithm consists of three main parts. Firstly, in order to align the image planes, the two chosen images are prewarped. Secondly, by linearly interpolating positions and colors of corresponding points, the two prewarped images are morphed. Thirdly, to transform the image plane of the new view to the desired one, the images are postwarped.

View morphing movie:
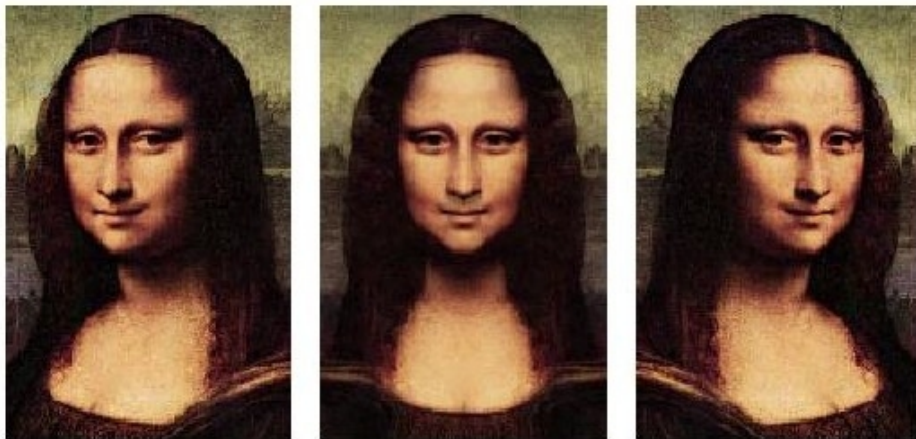
Another view morphing movie:
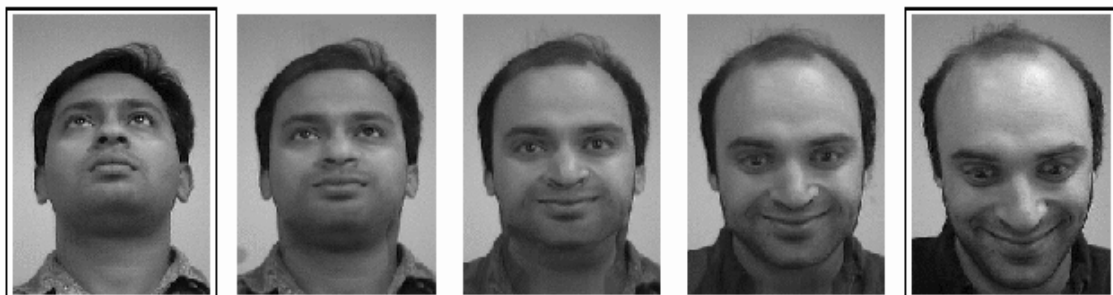
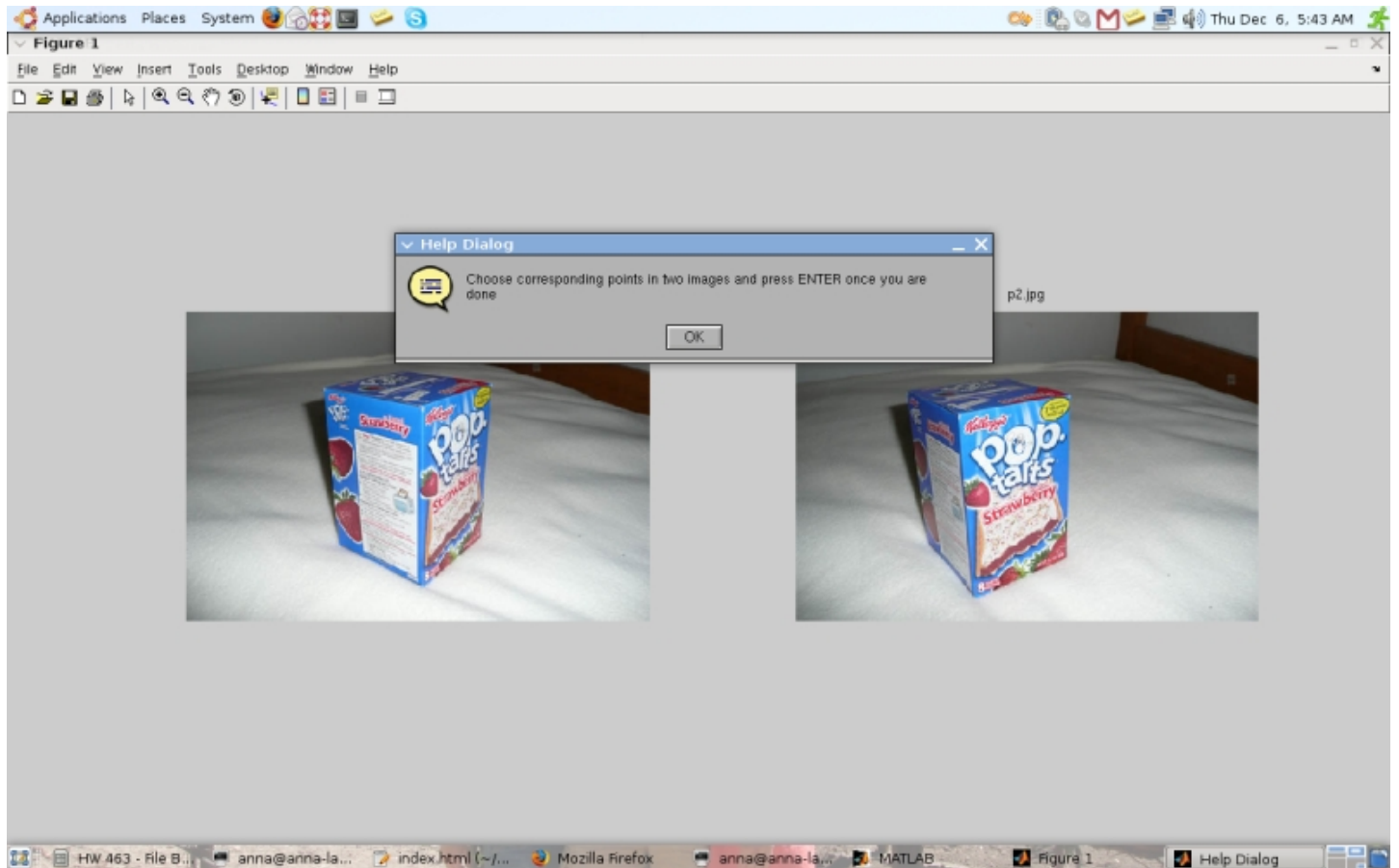### Timeline

1.   Read the paper extremely carefully
2.   Start coding
3.   Finish coding
4.   Create cool view morphing movies

Results

We made a GUI to get the 2D positions of 8 or more corresponding points in the two images I_0 and I_1.

Let us denote $u' \leftrightarrow u$ any pair of matching points in the two images I_0 and I_1.

Given sufficiently many point matches $u' \leftrightarrow u$ (at least 8) we moved on to automatically computing the image prewarping transforms.

We used the idea from the "In defence of 8-point Algorithm" paper by Richard L. Hartley to compute the fundamental matrix F of the two views defined by equation:

$u'^T F u = 0$.

We preceded the "8-point Algorithm" with a very simple normalization (translation and scaling) of coordinates of all matched input points. We also enforced that this fundamental matrix is singular, and of rank 2.

Once we obtained the fundamental matrix of the pair of pictures that we wanted to morph, we prewarped both images I_0 and

I_1 to make them parallel views of the scene: $I_0\hat{}$ and $I_1\hat{}$.
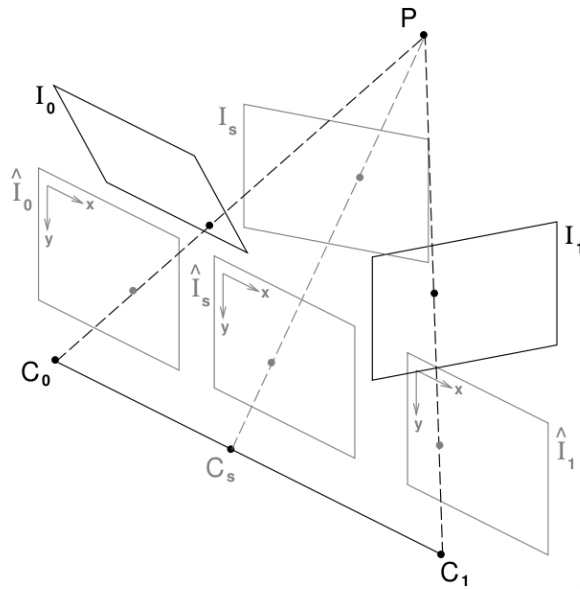
Example of prewarping:

Original pictures:

After prewarping

Then we applied a 2D morph between the two prewarped pictures by linearly interpolating positions and colors of the corresponding points in $I_0^\wedge$ and $I_1^\wedge$.

Here are our results:

Code

getcoords.m - the user clicks on corresponding points in two images and this function returns the coordinates of the corresponding points

findfundmat.m - calculates a fundamental matrix for the two specified images

normalize_points.m - computes normalization (translation and scaling) of the coordinates of points

computeH1H2.m - calculates the 3D rotation that we need to produce parallel views.

viewMorph.m - makes a view morphing movie.

References

- http://www.cs.washington.edu/homes/seitz/vmorph/vmorph.htm