Research Log - Week 12

JeffGWood@mavs.uta.edu

August 5, 2016

July 31, 2016

Decided to test *spectral clustering* routines `fnDistance` and `fnSimilarity` from June 5, 2016. Routines work on small images (approximaltey 100 pixels in size), but are bombing out `matlab` on larger images since for an image containing n pixels, the *Laplacian matrix* would be $n \times n$ in size requiring large amounts of memory. Put functions and test scripts in `Wood_Kamangar/StatusReports/StatusReport_12/`

I am looking into other methods of *image segmentation* including *graph-cuts* (described as the "gold-standard").

August 1, 2016

Started reading [Mark1997] [1].

SUMMARY: Paper describes expanded algorithm for *view interpolation* that building on [Chen1993] [2]. Pixels (including z-buffer and color information) in source images (referred to in article as *reference frames*) are transformed to the new new frame (referred to in article as *derived frames*) via *Euclidean*-transformations and *Affine*-transformations.

The paper addresses problems associated with *holes* being proudced in the derived frame, which result from a number of sources. They inlcude pixels *occluded* in the reference frame. Another source are surfaces that are highly incident to the image plane in the refence frames, but more closely parallel to the image plane in the derived images. The occurance of holes can be addressed through the use of a *mesh* for surface reprsenation (similar to that resulting from a *point cloud*). This results in holes of the latter type (surfaces of different angles to the image plane) being filled. Holes of the former type (from occluded pixel areas) occur along a siloutte of the backround/foreground surfaces. Normally the mesh results in a distorted surface connecting that foreground and background surface. The proposed algorithm (referred to in the article as *compositing*) addresses this issue by keeping the surfaces distinct and seperate and filling in the missing pixels with those containing the maximum (farthest) z-value.

August 2, 2016 Finished reading [Mark1997] [1]. Still unclear about some aspects including details calculations in section **4.3 Connectedness Calculation**.

SUMMARY: The compositing algorithm works by transforming the pixels in each *reference* frame to seperate *transformed-reference* frames. Each pixel buffer contain position, z-buffer, and color information. Because a multiple pixels from a single *reference* frame can be mapped to the same pixel buffer in its *transformed-reference* frame, pential new pixel values are compared with those already in the pixel buffer, with those pixels containing the closest (minimum) z-value remaining in the buffer.

Another aspect of the proposed algorithm is the treatment of *rubber surfaces* that occur along the siloutte lines between the foreground and backround segments of the generated mesh surface. This is handled by the notion of *connectedness* of surfaces. Pixels with mesh vertices part of a single object surface are considered to be *highly-connected*, whereas mest trianges covering disjoint and seperate surfaces have *low-connectiveness*.

An additional concept the authors make use of is *confidence value*. The article references the confidence value as the ratio of a pixel's *solid angle* in the reference frame to the *solid angle* in the derived frame. It is essentially a measure of how much a surface is parallel to the image plane. Surfaces, whose normal vector turns *towards* the image plane when transforming from the reference frame to derived frame, result in *pixel holes* and have low confidence values. Sufaces, whos normal vector turns away from the image plane are oversampled and have high confidence values.

When selecting pixels for the *derived* frame a number of scenarios arise: If both pixels have high connectedness, the one with closer Z-value is used in the derived frame. If the Z-values are the same (within a tolerance), the pixel with higher *confidence value* is used. If only one of the pixels has high connectedness, that pixel is selected. If nneither pixel has high connectedness, the pixel with the higher confidence value is used. When dealing mesh triangles of low connectedness, instead of interpolating between the *foreground* and *background* surface textures, the surface texture with the *farthest* Z-value is used to approximate the occluded areas.

Question for Kamangar: I don't understand the explantion given for section **4.3 Connectedness Calculation**.

August 3, 2016 Started reading [Saito2002] [3]. I am trying to understand concept of *cross-ratios* for the article material. I also plan to add section regarding *homographies* to thesis document. I put `MatLab` code in `Wood_Kamangar/StatusReports/StatusReport_12/`

August 4, 2016 Continued reading [Saito2002] [3]. Will need to read [Saito1999] [4] for background on *projective grid space*. Summary of [Saito2002] [3] follows.

SUMMARY: [Saito2002] [3] describes a system used in *virtualized television* and *free viewpoint television*, and specifically with regards to televizing soccer matches. It defines a *projective grid space* (PGS) betweens two images I_1 and I_2. Instead of a cooridinate system where the basis vectors are all *orthoganal*, the PGS defines two of the basis vectors as being along the principal axis of each of the images being interpolated.

There exist *fundamental matrices* from I_1 to I_2 (denoted in the article as $\mathbf{F}_{12}$) and from I_2 to I_1 (denoted as $\mathbf{F}_{21}$) which transforms points in one image to *epipolar lines* in a the other image. Corresponding points $\mathbf{P}_1$ in I_1 and $\mathbf{P}_2$ in I_2 can similarly be transformed to epipolar lines in a mid-view image I_i by the fundamental matrices $\mathbf{F}_{1i}$ and $\mathbf{F}_{2i}$. The intersection of the two generated epipolar lines is the position of the corresponding point in $\mathbf{P}_i$ in I_i.

The viewing angle and position of the interpolated image I_i is confined to the baseline between two images I_1 and I_2 using *linear interpolation*. When a third image I_3 (viewed from a higher angle) is adde, the viewing angle and position are confined to the triangle connecting the optical centers of the 3 image planes using *barycentric interpolations*. The interpolation methods are used for both *pixel position* and *pixel color*.

Scene components are divided into 3 major components including:

- Players and Ball
- Field ground and goal
- Background including stands

pixel operations dependant on the component type to which it belongs. The *players and ball* component is considered to be *dynamic* since it is changing between fames. The *players and ball* components are actually sillouted areas created from extracting the *field ground and goal* components. The pixels in the intermediate views are determined from interpolating pixel values between boundaries (along the epipolar lines) of the sillouetted areas. The *field ground and goal* components are transformed via *homographies* with the planes corresponding to the *field ground* and sides of the *goal post* treated as seperate planes. **Although not explicitly stated, I'm guessing pictures of the field ground (without the goal post or players) may also be taken before hand to account for pixels that might otherwise be occluded (with the inclusion of the players and goal post)**. The remaining area containing the *background and stands* are transformed with *image mosaicing* and a *plane at infinity*.

August 5, 2016 **UPDATE:** Discussed the matter of memory issues related to *spectral clustering* detailed on July 31, 2016 with `levine@uta.edu`. Due to the memory issues related to rendering moderate size images (300x200 pixels) I decided it might be better to:

1. Downsample original image to manageble size on which *spectral clustering* can be performed.
2. Extract edge regions of down sampled areas.
3. Partition original size image into manageble sub areas.
4. Perform *spectral clustering* on sub areas corresponding to edge regions extracted from downsampled image.
5. Join sub areas from image partitioning by mapping possibly non-equal segment labels between sub areas.

References

[1] William R. Mark, Leonard McMillan, and Gary Bishop. Post-rendering 3d warping. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, I3D '97, pages 7–ff., New York, NY, USA, 1997. ACM.

[2] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '93, pages 279–288, New York, NY, USA, 1993. ACM.

[3] Hideo Saito Makoto, Makoto Kimura, Satoshi Yaguchi, and Naho Inamoto. View interpolation of multiple cameras based on projective geometry. In *In: International Workshop on Pattern Recognition and Understanding for Visual Information*, 2002.

[4] H. Saito and T. Kanade. Shape reconstruction in projective grid space from large number of images. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, page 54 Vol. 2, 1999.