

# Research Log - Week 12

JeffGWood@mavs.uta.edu

August 3, 2016

---

July 31, 2016      Decided to test *spectral clustering* routines `fnDistance` and `fnSimilarity` from June 5, 2016. Routines work on small images (approximate 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 produced in the derived frame, which result from a number of sources. They include pixels *occluded* in the reference frame. Another source are surfaces that are highly incident to the image plane in the reference frames, but more closely parallel to the image plane in the derived images. The occurrence of holes can be addressed through the use of a *mesh* for surface representation (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 silhouette of the background/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 separate 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 separate *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, potential 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 silhouette lines between the foreground and background 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 mesh triangles covering disjoint and separate 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. Surfaces, whose 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 neither 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 explanation 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/

## 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.