

## 7. Conclusions

So far we have looked at what morphing is and how it can be used. We examined a number of techniques for morphing, noting trends in expression and flexibility. We saw examples of various morphing programs and noted their differing approaches to similar problems, particularly with regards to the keyframe layout, the approach to feature specification on the images, how correspondence is determined, and how modal their interfaces were, noting that giving something a state makes it more complicated to work with.

With the knowledge gleaned from Chapter 2 then went on to look at the requirements for the tool and used various literatures to identify key problems, ideals to work towards for desirable morphs, and possible directions to follow in order to meet these requirements for both manual and automatic morphing. These plans were cemented in the design chapter where we decided to use Java to implement a field morphing technique with frequent releases and Client feedback.

After careful consideration of the techniques to use we started to build a product that was largely suitable for the Client's purposes, producing results that were mostly acceptable, albeit after a significant delay. The product was relatively stable despite it not being a final release. @@@ Subsequent versions moved towards a semi-automated feature detection system @@@

This project has realised its aim of yielding a morphing tool, though it has failed to significantly expand on work already widely available. In particular, the morphing tool offers little improvement over most manual morphing programs and doesn't add significantly to work on automating the process of feature specification.

Through reading this report we hope you have developed a greater understanding of morphing techniques and an appreciation of this aspect of computer graphics.