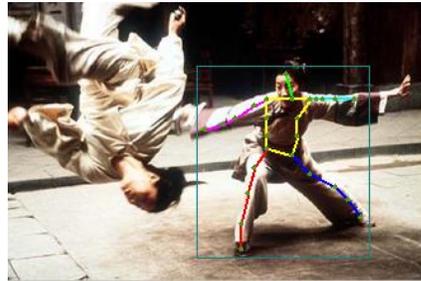


VisionGL: Towards an API for Integrating Vision and Graphics

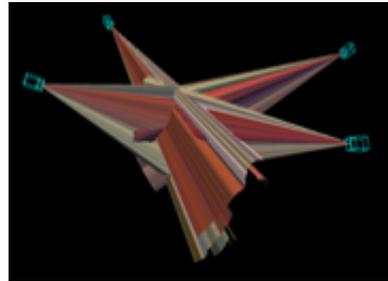
Gregor Miller and Sidney Fels*
Human Communication Technologies Laboratory
University of British Columbia, Vancouver, Canada



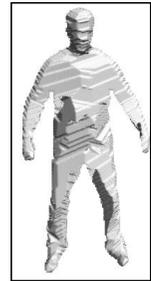
(a) Blending



(b) Motion/Appearance Capture



(c) Reconstruction



(d) Meshing

Graphics and Vision are inverse problems and yet frequently work together in many applications, such as: (a) image stitching, where vision finds position and property transforms and graphics blends images based on these; (b) motion/appearance capture and modelling of pose; (c) 3D reconstruction, such as from multiple views; and (d) meshing of objects recovered from images (e.g. motion flow or estimated shape).

Computer Vision and Computer Graphics have a long history of intersecting research, methods and applications. In general, it is the job of vision to extract a model from one or more images (and viewpoints) to represent some real-world object or scene. Graphics then takes this model and interprets it based on some context to render the model for visualization and interaction. There are many examples of this, such as avatar creation from Kinect™ (extract depth and pose of person, mesh and render), panorama stitching in cameras (extract transforms among images and blend) and performance/appearance capture. Recently we introduced *OpenVL* [Miller and Fels 2013], an abstraction for computer vision at a level such that general developers and researchers in other fields can apply sophisticated vision methods in their work. Other vision frameworks generally present APIs to developers as lists of specific computer vision techniques e.g. detection using Haar cascades. Application of these methods under real-world conditions requires significant algorithmic knowledge and a steep learning curve. *OpenVL* hides the details of algorithms behind an interface flexible enough to provide solutions to a wide variety of vision problems. Given this abstraction of vision, we would like to formulate an abstraction to perform graphics operations on the result of *OpenVL*.

We propose a new abstraction called *VisionGL* which can take defined computer vision outputs and render them as required by the developer for visualization, debugging and application interaction. *OpenVL* models image contents as *segments*: regions within the image which are distinct from their surroundings. The definition of distinct is provided by the developer through *properties* such as colour, texture, intensity, blur and so on. *VisionGL* can take these segments and re-render them to an image to provide an overview of their location (such as outlines), average colour (to indicate the area covered and the distinctiveness of the colour) or using their original colour information. Various segments can also be extracted

based on conditions (e.g. a certain colour or texture) and can be composited with other images or results (e.g. from other segmentations). Applications could then employ *OpenVL* for chroma keying or background subtraction and use *VisionGL* to render a composite.

Image stitching is accomplished by providing a description of the variances between segments in a set of images (such as position, intensity and blur) with a request for the type of transform (e.g. affine or projective). *VisionGL* takes the original images (optionally with the segmentation) along with the transforms and can be instructed to blend the images together, using either an image-based or a segment-based approach. *OpenVL* is also capable of pose estimation from colour images, which can be used for markerless motion capture, with applications such as avatar creation (although the accuracy of these methods is not currently at the same level as depth-based or marker-based approaches). We propose a modelling approach in *VisionGL* which would use a human model to match the skeleton returned by *OpenVL* and approximate a pose and allow extraction of the colour information in the image to match the appearance of the person. The final aspect of the *VisionGL* framework is a meshing approach: this can be used to create motion fields produced by optical flow approximation to alter a mesh across time, or to mesh various results from vision processes, such as 3D reconstruction (multi-view stereo, visual hull) or depth approximation (two-view stereo reconstruction), taking into account various complex issues such as depth discontinuities, contiguous surfaces and noise which can be controlled by the developer.

Our *VisionGL* framework is part of an on-going project to create a simpler ecosystem for researchers and developers use vision and graphics methods without requiring expert knowledge. We believe the full integration of *OpenVL*, *VisionGL* and *OpenGL* will provide a full toolset to engage the developer community and enable the creation of new applications.

*e-mail: {gregor,ssfels}@ece.ubc.ca

References

- MILLER, G., AND FELLS, S. 2013. *OpenVL: A task-based abstraction for developer-friendly computer vision*. In *Proceedings of the 13th IEEE Workshop on the Applications of Computer Vision (WACV), WVM'13*, IEEE, 288–295.