SOFTWARE FOR 3D-TV AND 3D-STEREOSCOPIC WORKSTATIONS

M. W. Siegel and V. S. Grinberg, A. G. Jordan, J. S. McVeigh, G. W. Podnar, S. Safier, Sriram S.

The Robotics Institute, Carnegie Mellon University
Pittsburgh PA 15213 USA
phone: 412 268 8802, *fax:* 412 268 5569
e-mail: mws+@cmu.edu

ABSTRACT

In anticipation of high quality 3D-TV and 3D-stereoscopic computer workstations soon being "enabled" by new developments in HDTV and high resolution flat panel displays, we are studying and developing some of the software that will be needed. This paper surveys three software topics that span the breadth of our work: issues relating to geometry, to compression, and to an infrastructure for 3D-stereoscopy in "windows"-based operating system environments. In the concluding section some general issues are presented with the aim of stimulating discussion and consensus in the 3D-stereoscopy community.

1. INTRODUCTION

Grafting high quality 3D-stereoscopic viewing onto existing and anticipated near future high definition flat panel TV and computer video displays depends to some extent on being able to perfect several existing commercial technologies that will make the hardware behave stereoscopically. In contrast, it depends critically on being able to define and build appropriate supporting software, which is at present much less mature than the hardware. We are addressing several software issues that we believe are essential to making this transition. In this paper I summarize our work in three of these areas:

1) geometry algorithms for computing 3D-stereoscopic views of three dimensional world models,
2) compression algorithms for 3D-stereoscopic (and more general) images and image streams,
3) approaches to presenting 3D-stereoscopic imagery in a window on a screen with multiple windows.

The three following sections summarize our work in 3D-stereoscopic geometry, compression, and presentation software; the references therein point to articles in which we have described these efforts in detail. The final section presents conclusions, and raises some philosophical and architectural issues about software that it might be mutually beneficial for the 3D-community to address cooperatively.

2. GEOMETRY

We consider explicitly the 3D-display paradigm in which left and right eye perspectives are both rendered within the confines of a single window on a single screen that is suitably equipped to direct the left eye's perspective to the left eye, and the right eye's perspective to the right eye, of one or more viewers who will all see the same two perspectives. Although I do not demonstrate it here, the outcome is actually more general than this tightly constrained problem statement makes it seem: in fact the result applies, interpreted but unmodified, to single viewer alternatives such as head mounted displays, and to multiple viewer alternatives in which more than two perspectives are rendered in a single window on a single screen equipped to direct the appropriate perspective to the appropriate eye of each viewer.

We address at this time only geometry, ignoring all details of how it is arranged (time multiplexing, polarization multiplexing, angular multiplexing, etc) that each eye will see only the intended view. In the following three subsections we discuss three increasingly general (and thus increasingly complicated) optical system scenarios for 3D imaging and image generation by computer graphics methods:

This work was supported by the Advanced Research Projects Agency under ARPA Grant No. MDA 972-92-J-1010

1) "naked eye reality", the issues relating to creating and displaying imagery that is geometrically indistinguishable from reality as seen by unaided normal human eyes,

2) "augmented eye reality", the issues relating to creating and displaying imagery that, when seen by unaided normal human eyes, is geometrically indistinguishable from reality as seen through a binocular optical instrument, e.g., a pair of binoculars or a stereomicroscope, and

3) "non-projective optics", the issues relating to creating and presenting 3D-stereoscopic imagery that is geometrically indistinguishable from the the real imagery produced by real lenses (which, unlike ideal lenses, cannot be modelled by projection through a point).

2.1. Naked Eye Reality

The geometry of "naked eye reality"[1] has been well known since well before the present electronic and computational imaging age[2, 3]: to avoid geometrical artifacts, especially to avoid vertical parallax in the screen corners and occlusion of corresponding imagery at the screen edges, it is necessary that the real or computer modelling camera axes be parallel, that the camera sensors be horizontally offset so that left and right fields of view cover the screen identically, and that in all other respects the cameras be identical. Furthermore, for the imagery to be geometrically indistinguishable from reality, the camera axes must be separated exactly by the eventual viewer's interpupillary distance. Getting all these details right requires knowing, at image capture or image generation time, the parameters of both the viewer and the theater or display device. These parameters explicitly include the viewing screen size, and its distance and angular orientation relative to the viewer. We note that the implications for purveyors of purported "virtual reality" are indeed severe!

Despite the fact that these requirements are well known, for reasons of expedience they are regrettably frequently honored in the breach. This is especially the case for real cameras, where the practical temptation to converge the cameras (so as to improve the overlap of their fields of view) is often very strong. The rules are often broken by computer-generated imagery as well, for example the programmer succumbs to the temptation, analogous to converging the cameras, to approximate camera translation with scene rotation. The consequences, at viewing time, vary from a nearly imperceptible degree of annoyance to, under some circumstances, complete loss of stereopsis as the viewer becomes unable to fuse the left and right eyes' images. Our experience, which is driven by a video application (remote inspection of aircraft skin for small defects) in which converging the cameras is the only *easy* way to achieve overlapping fields of view, led us to build a geometrically correct 3D camera, with parallel axes, and with the image sensors remounted on precise left-right translation mechanisms.

[The CCD translation adjustment is at present manual, but in principle it could be mechanically coupled rigidly to range via the lens focus adjustment, or flexibly via image understanding software. The cameras send two synchronized NTSC streams to a display unit that allows them to be viewed time-multiplexed at 640 pixels by 480 lines by 60 left-right pairs per second on a 120 Hz VGA monitor.]

Working with this 3D-stereoscopic camera, the targeted users of the application (commercial aircraft inspectors) tell us that its image quality and viewing comfort are comparable to their everyday "naked eye reality". In contrast, when the cameras were converged they reported in some cases vague discomfort, and in others, difficulty or inability to fuse the images. The positive outcome of these *video* experiments bolsters our confidence in the wisdom of the decision we made to restrict our 3D-stereoscopic *computer graphics* rendering options to the venerable parallel axes camera model.

It is useful to point out explicitly, as this fact will be applied in the next section, that to render "naked eye reality" it is not necessary to have a complete model of human eye, with all its aberrations, distortions, and other defects. It is only necessary to present, to the real eyes, artificial imagery formed by projections of the rendered scene through the centers of the eyes' pupils onto the display screen; ignoring accommodation discrepancies, which are manageable by, e.g., simple reading glasses that remove the screen to infinity, the eyes cannot geometrically distinguish this rendering from reality.

2.2. Augmented Eye Reality

The desire to display "augmented eye reality"[4], i.e., to permit the naked eye to perceive, on a 3D-display screen, the world (or a world) as it would be naturally perceived through a binocular optical device, requires addressing certain "philosophical" as well as engineering issues. Human culture appears to have resolved implicitly the issue of how to draw flat images of things that are too small or too big to view comfortably at arm's length: the artist, or the programmer, imagines a scale model that fills the field of view at arm's length, and draws the model. Because the absolute sizes of the objects of interest are hidden by this scaling, it is common practice in technical drawing to insert in the drawing a bar labelled with the length it represents, e.g., "0.1 mm" or "1 parsec". We call this *convention* "uniform" scaling, since length, width, and depth are all scaled by the same factor. However the scaling. i.e., magnification, that is performed by optical systems is *not* uniform: an optical system (more-or-less by definition) with transverse (length and width) magnification m has longitudinal (depth) magnification -*m**2*. Thus pictures taken through long focal length (telephoto) lenses look "squashed" (foreshortened in depth), and pictures taken through short focal length (wide angle) lenses look "stretched" (exaggerated in depth); we call this *reality* "optical scaling".

Computer graphics programmers typically adopt the uniform scaling convention for portraying objects that are too small or too large to portray usefully from a naked-eye perspective. They typically reserve *optical* scaling (or some artistic approximation to it, much as artists reserve it in still and cinematic photography) for imagery that is intended to allude to moods that are commonly described with words like "close" and "distant".

These comfortable and apparently natural viewing conventions for flat imagery do not transfer straightforwardly to 3D-stereoscopic imagery: seen stereoscopically, uniformly scaled scenes look unnatural, like the models in museum showcases. In these uniformly scaled 3D-stereoscopic renderings people look like puppets, dinosaurs look like children's toys, micro-organisms and insects look not like small things viewed up close via the stereo-microscope, but rather like cat and dog sized creatures viewed at arm's length. To make these objects and scenes look right stereoscopically, it is effective to scale them optically, making them look the way they would look through an appropriately selected and adjusted pair of binoculars, or a stereo-microscope, depending on the scaling direction required.

We call the rendering, on a 3D-stereoscopic display, of views of large objects as they would be perceived at a great distance through a pair of binoculars, and of small objects as they would be perceived at a small distance through a stereo-microscope, "augmented eye reality".

For simple (paraxial) models of optical systems, rendering "augmented eye reality" is not difficult. Recalling that rendering "naked eye reality" does not require a sophisticated model of the real human eye, only projection through the center of the eyes' lenses onto the display screen, it follows that rendering "augmented eye reality" requires only projecting onto the display screen the image created by the optical system.

Because the lens equation that describes imaging by a simple optical system is itself (with appropriate interpretation of its sign conventions) a projective transformation, it can be concatenated, by simple matrix multiplication, with the projective transformation that describes the imaging by the naked eye. The result of the concatenation is another projective transformation for "augmented eye reality". The new transformation replaces, entirely mechanically, the initial "naked eye reality" transformation in all subsequent rendering operations.

We have integrated the two common optical accessories, pairs of binoculars and stereomicroscopes, and several elementary devices, such as plane mirrors, into our 3D-stereoscopic rendering system. To aid us in empirically selecting the viewing magnification that works best when we want to render arbitrary uniformly scaled 3D data files (e.g., those we find in various repositories on the World Wide Web), we have developed a user interface in which trial values of the optical and physical parameters are entered, and are typically optimized by the user in few cycles of numerical experiments.

2.3. Non-Projective Optics

As stated implicitly in the previous subsections, perspective viewing, that is projection of a three dimensional space through a point onto a surface (typically but not necessarily a plane), is an instance of a projective transformation. Perspective viewing is the basis of the entire field of "3D computer graphics". At first glance it seems like a good model of reality: it describes the plane mirror exactly, the pinhole camera exactly, and cameras with simple lens to a good approximation. However it departs substantially from reality for certain optical systems that ought to go hand-in-hand with 3D-stereoscopic viewing: wide angle and fish-eye lens cameras[5]. The advantage of simultaneous wide field and 3D-stereoscopy is apparent in diverse applications, for example endoscopic microsurgery at one end of the distance scale, and teleoperation of remote planetary exploration vehicles at the other. But wide angle and fish-eye lenses manifestly do not implement projective transformations: projective transformations (more-or-less by definition) map straight lines into straight lines, whereas the signature of a wide angle or fish-eye lens image is its barrel distortion, the mapping of straight lines into curves radiating from the center of the picture.

Thus to correctly draw wide angle imagery, in essence to create a precise model of distortion, it is necessary to depart from simple perspective viewing. The optical theory that models the distortion inherent in real lenses is an expansion (essentially a series expansion of sines that, in the limit of ideal lenses, are set equal to their respective arguments) in terms of the five Seidel-Schwartzschild aberration coefficients for astigmatism, spherical aberration, curvature of field, coma, and distortion. The first four can be assumed to be well corrected in any modern camera lens. Collectively correcting them assures a *sharp* image at the inevitable price of uncorrectable distortion. The effect is most apparent with wide angle and fish-eye lenses because (obviously) these lenses *must* handle rays that are far from paraxial.

While the reader probably remembers seeing distorted imagery in computer graphics, we doubt that these images are routinely rendered by raytracing or precise modelling of actual wide angle optical systems; we suspect that they are rather typically rendered by the standard perspective drawing algorithms, then empirically warped in 2D to create a suitably pleasing illusion[6, 7]. We began recently began to address in detail formal modelling of binocular optical systems with distortion. The design interface discussed in the previous section is now being extended to allow rendering "augmented eye reality" with real (vs ideal) optical components specified in terms of the five aberration coefficients.

3. COMPRESSION

It has been pointed out many times that the large correlation between left and right eye perspective views can be exploited to encode both views in only slightly more bits than are needed to encode either view alone[8, 9]. With motion sequences it is possible furthermore to exploit the even higher correlation that may exist between left and right eye perspective views that are suitably offset in time[10]. Also, the similarity between motion disparity and perspective disparity makes it possible for 3D-stereoscopic encoding to take advantage of algorithms, e.g., MPEG, and their hardware implementations, which have already been carefully crafted for efficiency (albeit optimized for encoding single perspective motion sequences).

The usual approach in this field is to encode one stream using a conventional motion sequence compression algorithm (like MPEG), to compute a disparity map stream between temporally corresponding frames of the two streams, to encode the disparity map stream (using perhaps a different compression algorithm), and at the receiving end to estimate the second stream from the decoded first stream and the independently decoded disparity map stream.

Most of our current work in 3D-stereoscopic compression emphasizes essentially this conventional approach, looking for its innovation at what we expect will be an especially efficient multiresolution approach to computing and encoding the disparity map. This multiresolution-based compression work is discussed in the following subsection. Recently we have begun to address the similar but more general area of compression of families of image streams, e.g., from a grid of cameras, potentially not as precisely matched and aligned as in an ideal 3D-stereoscopic camera pair. We call this field "multivariable-based compression"; we discuss our early progress in this area in the subsection following the next one. Either approach, or both, may eventually prove valuable in implementing a completely symmetrical approach that will scalably preserve equal image quality in both streams.

3.1. Multiresolution and Segmentation Based Compression

The multiresolution approach appears to be particularly appropriate for this class of problem because it has significant relative advantages for both key elements of the problem: the disparity map calculation per se, and the map's eventual encoding. Initially calculating the disparity map at the bottom of the multiresolution tree is efficient because the resolution is low; it is also robust, because disparities computed at low resolution are relatively immune to errors caused by noise, occlusions, and aliasing. Refinement of the disparity map to higher resolution further benefits from being able to share some of the effort associated with calculating the higher resolution (higher spatial frequency) subimages.

Recently we described a disparity-based multiresolutional segmentation scheme for stereoscopic image compression that significantly reduces the number of segment disparities needed to represent one image of a stereoscopic *pair*, given the other, by efficiently partitioning it based on the disparity content present [11]. We are now exploring several alternative means of employing this segmentation in a stereoscopic *sequence* compression framework [12].

Several groups have suggested such joint coding of the two streams [13, 9, 14, 15, 8]. Building on this work, we are addressing the specific problem of coding stereoscopic image sequences with low excess bandwidths over the bandwidth required for monocular sequences, without unduly sacrificing the perceived stereoscopic image quality [16, 17, 11]. We have now succeeded in showing that we can achieve significant reduction in average overall bit-rate by shifting to content based adaptive coding strategies.

An adaptive block size disparity-based segmentation and the subsequent transmission of these segment disparities to achieve very low stereoscopic coding overhead compared to a single (monoscopic) image compression was discussed in [11]. This scheme adapts the overhead to the disparity detail present in a given stereoscopic image pair, unlike a fixed block size based scheme. Also, by segmenting at object boundaries, it reduces the number of spurious matches while preserving disparity discontinuities.

Our current work [12] treats the system level integration needed to employ this segmentation in the compression of stereoscopic sequences. The problem is complicated by the fact that quadtree based segmentation is not invariant under perspective (or even affine) transformations. Thus the most natural schemes need not provide the most visually pleasing results at low bit rates. Several possible temporal extensions, ending with the particular scheme that provides the best quality stereoscopic stream at high compression rates (albeit with higher computational complexity) are presented in the following paragraphs.

3.1.1. Disparity Based Segmentation

Typical stereoscopic images contain large regions of almost constant binocular disparity arising from the scene backgrounds and large objects at a fixed depth. Fixed block size (FBS) based disparity estimation schemes divide these regions into smaller blocks, thus requiring more block disparities to be coded than is necessary. Also, matching small featureless blocks lead to spurious matches that affect the smoothness of the estimated disparity map, which prevents an efficient predictive coding of the block disparities. Also, block based disparity estimation fails for blocks containing portions of two objects at different depths. To overcome these drawbacks, and to achieve very low overhead for coding one image of the stereoscopic pair given the other, we first implemented a multiresolution and disparity based segmentation (DBS) scheme [11].

In our scheme the stereoscopic image pair is subjected to multiresolutional decomposition to get progressively lower resolution images. The segmentation starts at the coarsest resolution level and is recursively updated as we reach the original resolution level. At each level, a quadtree decomposition is performed with the disparity compensated error as the splitting criterion. The splitting locations are obtained using a simple "dominant edge selection" algorithm (defined in [11]) from the block's intensity values, thus making the segments align with object boundaries (which usually occur at an intensity discontinuity). Each segment is recursively partitioned at the next higher level of resolution. Thus, this scheme adapts the size of the segments according to the disparity detail present in the stereoscopic pair. This significantly reduces the number of bits needed to represent one image of the stereoscopic pair given the other, even after taking into account the overhead for the representation of the segmentation.

3.1.2. Segmentation Based Stereoscopic Sequence Compression

This segmentation scheme can be incorporated in a stereoscopic sequence compression framework in several different ways. We are guided by the overall goal of building a system level scheme that gives an overall minimum bit rate for the jointly coded stereoscopic stream at reasonably high stereoscopic image qualities, while maintaining a moderate computational complexity at the encoder and low computational complexity at the receiver.

The method we call "reversal of prediction direction" has proven to be an invaluable tool in this process. Conventional motion estimation schemes partition the target image into non-overlapping blocks and then find the best match for each target image segment in the reference image. By "reversal of prediction direction" we mean the reverse case, partitioning the reference image into non-overlapping segments, and finding the best matching segments in the target image. This method is particularly useful for efficiently identifying and coping with occlusions.

Within this framework, we have investigated the pros and cons of three coding schemes:

1) Encode one image sequence independent of the other stream.

2) Track disparity-based segments over several frames in both streams, encoding only motion and the details of newly unoccluded regions, until the scene content changes significantly.

3) Noting that (2) is very natural with respect to content, but it cannot be coded efficiently, modify (1) so that both streams are efficiently coded based on segmentation, and only one segmentation is performed per stereoscopic pair of frames. Both streams are coded efficiently by adapting the number of segments, first to the disparity detail,

then to the motion detail. However, the frequency of segmentation used increases the computational complexity significantly.

By replacing the disparity based segmentation with homogeneity-based segmentation at the coarser resolution levels much of this complexity can be eliminated, since variance within the block is an adequate and simple to compute homogeneity criterion.

Quantitative measures of performance are presented and explained in detail in [12], with a comprehensive summary in Table I therein. To summarize briefly, with encoding parameters that, at 30 frames per second, result in no observable artifacts, depth perception that is very close to that seen in the original sequences, and excellent subjective quality, compression factors are typically the order of 60 for the main stream and 150 for the auxiliary stream, including all overheads.

We are currently studying the extent of graceful degradation in quality while decreasing the bit rate. Also, we are considering different ways to reduce the frequency of segmentation and thus the associated computational complexity. We intend to explore the extensibility of this scheme to a symmetric extension of the binocular imaging setup, in which three cameras are used. The middle (cyclopean [18]) camera will generate a high definition monoscopic stream, whereas the left and right (possibly low definition cameras) would generate the left and right disparity streams.

3.2. Multivariable-Based Compression

We look toward a future time in which in a variety of applications it may be considered desirable to efficiently code more than two related video streams, e.g., to provide for look around, multiple viewers each seeing the perspectives appropriate to his or her position relative to the display screen, etc. The concept generalizes straightforwardly to include families of temporally parallel video streams related to each other via one or more arbitrary smoothly varying parameters. When the parameter is horizontal perspective and the number of streams is exactly two the general concept reduces to ordinary binocular stereoscopy. We call these families of video streams "generalized", or "multi-view" video signals[19].

To be practical, multi-view signals will have to be compressed efficiently to overcome the increase, linear in the number of streams in the family, in the bit rate needed to transmit the signal. If the signal is to be encoded using predictive techniques, the reference frame most "similar" to the frame to be coded has to be selected dynamically.

Prediction similarity depends on the structure of the scene, the camera configuration, and the motion of both the scene objects and the cameras. Although at least some of these quantities will most likely vary considerably as the scene evolves and changes, in all known prior work[20, 21, 22, 23, 24], the reference frames used in the prediction process were fixed and heuristically chosen. Since these static reference frames do not necessarily yield the optimum predictions, compression performance suffers.

In preparation for an anticipated ongoing effort we will make to develop multi-variable based compression methods, we have begun breaking ground in three areas:

1) the quantification of prediction similarity,

2) the estimation of similarity from the variance of composite displacement vector maps, and

3) the application of this estimate to the adaptive selection of the best possible reference frame. In the rest of this section we summarize the gist of the concept of generalized video signals, and examine the potential gains in terms of prediction and compression that can achieved through dynamic reference frame selection.

Our current choice of reference frame selection method is based on estimated prediction similarity for all candidate reference frames, where similarity is measured by the absence of occlusion. Block-based motion, or displacement, estimation is performed once for each frame in the video signal. The reference frames used for these single-step displacement estimation procedures are specified to ensure prediction relationships between all frames. The resulting displacement vector maps are processed to reduce the occurrence of erroneous estimates caused by occluded regions. Composite displacement vector maps are generated for each possible reference frame through simple vector addition of the processed vector maps. The amount of occlusion between two frames is estimated by the relative variance of the composite vector map, and the reference frame with estimated minimum occlusion (maximum similarity) is selected for the final prediction process.

We have obtained prediction and compression performance results for two generalized video signal sequences using both this scheme and conventional schemes where the reference frames were pre-selected. Our experiments with the fixed reference frame schemes demonstrate that the relative location of the optimum reference frame varies considerably. This validates the underlying hypothesis that the reference frames should be adaptively selected to achieve optimum performance. The adaptive scheme produced significant average prediction PSNR gains (0.79 to 1.75 dB) over the non-adaptive schemes. When the predicted frames were used in a modified MPEG encoder simulation, the stream compressed using the adaptively selected reference frames required, on average, over 10 percent fewer bits to encode than the nonadaptive techniques. For individual frames, the reduction in bits was sometimes as high as 80 percent.

These results provide encouragement about the eventual practicality of being able to transmit, alongside a reference stream, a multi-variable gradient description stream that will permit predicting, at the receiver, related imagery in potentially useful domains such as horizontal perspective (yielding on-demand look around and binocular stereoscopy), vertical perspective (yielding on-demand look over and look under), and optically correct synthesis of camera zoom.

4. 3D-STEREOSCOPY IN WINDOWS-BASED INTERFACE ENVIRONMENTS

It appears certain that the interfaces exemplified by Microsoft Windows*(TM)*, the Apple Macintosh*(TM)* O/S, and Unix*(TM)*'s X-Windows will be the display paradigm for the foreseeable future. It is thus imperative, if 3D-stereoscopy is to become an integral feature of future computer workstations, that it fit naturally into this paradigm. Thus 3D-stereoscopic vs flat ought to be just another set of declarable properties of a window, on the same footing as window property sets like color vs monochrome vs black-and-white, text vs graphics, and still vs motion. In contrast, with only rare exceptions 3D-stereoscopy is today implemented in hardware and software that take over the full screen, negating the multiple windows paradigm.

We believe that in the long run it will be necessary to incorporate the assumption that displays will be 3D-stereoscopic into the lowest levels of computer workstation hardware, operating system, human interface, and graphics programming library design. Only by doing so will mundane but frustrating and expensive-to-work-around impediments be overcome. A simple example is cursor management, which in the flat environment is handled efficiently and invisibly by dedicated low level hardware and software. When 3D-stereoscopy is grafted onto a workstation whose underlying architecture is monoscopic, then creating, manipulating, and interpreting a cursor with depth in its appearance and its motion becomes a major programming effort at application building time, and a major consumer of processor cycles at application run time. To aid us in cataloging and analyzing the many issues in this area, as well as to give us some practical alternative interim working solutions for building our own applications, we have implemented two high level approaches to 3D-stereoscopy in windows under the X-Windows system on Silicon Graphics*(TM)* workstations. The first approach, using an above/below screen doubling format, is described in the following subsection. The second approach, using double or quadruple buffering for individual windows, is described in the next subsection.

4.1. Screen Content Doubling by Above/Below Splitting

Our first implementation of stereo-in-a-window[25] is built within the X-Windows environment by making it cognizant of the "StereoGraphics*(TM)*" 3D-implementation trick that is available on Silicon Graphics*(TM)* and Sun*(TM)* workstations, and (with additional internal or external hardware) in PCs and Macintoshes*(TM)*. [In this mode the left and right eye views are squeezed vertically by a factor of two and stacked above and below each other in the active screen buffer. With an appropriate monitor, an "extra" vertical synchronization pulse interpolated in the middle of the screen causes an "extra" retrace, overlaying the two squeezed images; rescaling by the monitor's vertical amplifier stretches both images to fill the screen, restoring them to their normal heights. The initial above/below spatial multiplexing is thus converted to a frame sequential (or field, or even subfield sequential, depending on the exact implementation) time-multiplexed mode. The display screen is then viewed through suitably synchronized shutter goggles.] Every high level call to create or update an abstract window generates two lower level calls to create or update the actual above (left eye) and the actual below (right eye) windows. If the abstract window has the 3D-stereoscopic property declared then its actual above and actual below versions will be drawn appropriately differently; otherwise the actual above and actual below window contents will be identical.

This mode works well except for one nearly insurmountable problem: as far as the X-Windows system knows, the two actual windows corresponding to left eye and right eye views are independent; thus when a feature such as a pull-down a menu is accessed, the pull-down will appear in only one of the window pairs, e.g., the left eye's window. Similarly, the system cursor (which is restricted to the upper half of the normal screen) is visible only to one eye. The perceptual effect is, to say the least, annoying.

The programming effort that would be needed to overcome this limitation within the existing X-Windows system is so enormous that it leads unavoidably to the conclusion that achieving stereo-in-a-window using the X-Windows system at a high level is at best an interim solution.

4.2. Window Content Doubling by Multiple Buffering

The second approach we have implemented for stereo-in-a-window[26] within the X-Windows system uses the double or quadruple buffering capability of the Silicon Graphics*(TM)* graphics hardware. Double buffers, designateded "front" and "back", are provided at all levels of the product line; they are intended for animation applications, where it is desirable to display a completed frame in the front buffer while drawing the next frame in the back buffer.

The top member of the graphics engine line, the Reality Engine*(TM)*, is quadruple buffered, left/right and front/back. These machines are inherently capable of 3D-stereoscopic display in individual windows: the completed left-front/right-front pair (in a window that is designated both stereoscopic and active) are time-multiplexed at the screen refresh rate while the left-back/right-back pair are being drawn. The hardware is of course cognizant of whether a left or right buffer is being displayed, so it correspondingly toggles a built-in signal that controls the LCD shutter goggles. If drawing the back buffer is slow (because the image is large or extremely detailed), the motion may become jerky (because there are fewer than the desirable number of *new* frames per second), but it never flickers (because the selected buffer is always being redisplayed at the guaranteed fast-enough screen refresh rate.

Only the high end machines are quadruple buffered; the low- and mid-range machines are double buffered. While there is no explicit software support for stereoscopic display in individual windows using the double buffered machines, left and right images can nevertheless be alternately drawn and displayed in the front and back buffers and viewed through shutter goggles. Since there is no hardware support for stereo-in-a-window, the application software needs also to toggle some hardware device to which the shutter goggles can be synchronized; we use one bit of the parallel printer port, toggled at appropriate points in the drawing cycle by the application program. This method works well for small windows with simple content; however if the window is large or if its content is overly detailed, the drawing cannot keep up with the display, and flicker (not just jerky motion) results.

It is thus clear, in principle and as the outcome of our experiments, that quadruple buffering is the desirable approach to achieving stereo-in-a-window. We expect that as demand for 3D-stereoscopic viewing capability grows, in parallel with the continued increase in performance versus cost in computer hardware, this capability become routinely available on all serious graphics and video workstations. The limitations and inconveniences of the current quadruple buffering scheme implementation, which include a restricted screen resolution choice and a clumsy sequence of operations to turn on the feature (including an obligatory logout and login by the user!) will presumably fade away as user demand stimulates manufacturers to provide increased support for 3D-stereoscopy.

5. CONCLUSIONS

We have described the motivation, content, results, and plans for our work in three areas relating to software for 3D-TV and 3D-stereoscopic workstations: geometry, compression, and presentation of 3D in windows. Under *geometry* we have considered "naked eye reality", "augmented eye reality", and "non-projective optics" (lens distortion) issues. Under *compression* we have considered multiresolution approaches that use variable block size disparity calculation to segment images by content, and, looking forward to the future need to compress multiple temporally parallel streams, techniques for the dynamic selection of the optimal reference stream. Under *presentation* we have considered a high level approach that draws the screen twice in an above/below format, and a low level approach that uses double or quadruple buffering hardware to multiplex the content of individual windows.

How best to approach these (and most other) issues relating to software for 3D-stereoscopic television and computer graphics, depends in large measure on how 3D-displays will fit into future working and living environments. Stereoscopy has long been a field in which enthusiasts have predicted that very soon *all* optical imaging, drawing, and graphic representation of data will be in 3D; in fact, we may now be on the threshold, technologically, of being able to make this prediction a reality. Others (ourselves included) believe that in the predictable future 3D image capture and display technologies will remain sufficiently intrusive, relative to the flat alternatives, that 3D will for a long time be used primarily in applications that demand it, e.g., for resolving serious visual ambiguities in situations where it is important to resolve them. If the enthusiasts are right, then 3D-stereoscopy should be designed into the lowest levels of the software (and hardware) infrastructures. If the marginalists are right, then 3D-stereoscopy will for the foreseeable future continue to "piggy back" on the flat infrastructures, so developers will have to continue in the mode of inventing "tricks" that make the infrastructure do more than it was designed to do. Of course, this is an oversimplified argument: the enthusiasts can validly argue that if we pursue stereoscopy with a technological dedication to making it pervasive, then the intrusiveness of the technology will soon be so diminished that it will become pervasive. We would like to stimulate ongoing discussion and refinement of these issues in the 3D community.

6. REFERENCES

[1] V. S. Grinberg, G. W. Podnar, and M. W. Siegel. Geometry of binocular imaging. In *Stereoscopic Displays and Applications V*, pages 56–65, Bellingham WA, February 1994. SPIE/IS+T.

[2] A. C. Hardy and F. H. Perrin. *The principles of optics*, volume xiii of *International series in physics*. McGraw-Hill, New York, 1932.

[3] Takanori Okoshi. *Three-Dimensional Imaging Techniques*. Academic Press, New York, 1976.

[4] V. S. Grinberg, G. W. Podnar, and M. W. Siegel. Geometry of binocular imaging ii: The augmented eye. In *Stereoscopic Displays and Applications VI*, pages 142–9, Bellingham WA, February 1995. SPIE/I+ST.

[5] V. S. Grinberg and M. W. Siegel. Geometry of binocular imaging iii: Wide-angle and fish-eye lenses. In *Stereoscopic Displays and Applications VII*, page TBD, Bellingham WA, January 1996. SPIE/IS+T.

[6] Ned Greene and Paul Heckbert. Creating raster omnimax images from multiple perspective views using the elliptical weighted average filter. *IEEE Computer Graphics and Applications*, 6(6):21–27, June 1986.

[7] Paul Heckbert. Private communication, July 1995.

[8] I. Dinstein et al. Compression of stereo images and the evaluation of its effects on 3-d perception. *SPIE Applications of Digital Image Processing XII*, 1153:522–9, 1989.

[9] M.G. Perkins. Data compression of stereopairs. *IEEE Trans. on Communications*, 40(4):684–96, April 1992.

[10] M. W. Siegel, Priyan Gunatilake, Sriram Sethuraman, and A. G. Jordan. Compression of stereo image pairs and streams. In *Stereoscopic Displays and Applications V*, pages 258–68. SPIE/IS+T, February 1994.

[11] S. Sethuraman, M. W. Siegel, and A. G. Jordan. A multiresolution region based segmentation scheme for stereoscopic image sequence compression. In *Proceedings of the 1995 SPIE/IS+T Conference (San Jose)*, page pages TBD, Bellingham WA, February 1995. SPIE/IS+T.

[12] Sriram Sethuraman, M. W. Siegel, and Angel G. Jordan. Segmentation based coding of stereoscopic image sequences. In *Proceedings of the 1996 SPIE/IS+T Conference (San Jose)*, page TBD, Bellingham WA, January 1996. SPIE/IS+T, SPIE/IS+T.

[13] D. Tzovaras, M.G. Strintzis, and H. Sahinoglou. Evaluation of multiresolution block matching techniques for motion and disparity estimation. *Signal Processing: Image Communication*, 6:59–67, 1994.

[14] A. Tamtaoui and C. Labit. Constrained disparity and motion estimators for 3dtv image sequence coding. *Signal Processing: Image Communication*, 4:45–54, 1991.

[15] A. Tamtaoui and C. Labit. Coherent disparity and motion compensation in 3dtv image sequence coding schemes. *Proc. of ICASSP '91*, IV:2845–8, 1991.

[16] Sriram Sethuraman, A. G. Jordan, and M. W. Siegel. Multiresolution based hierarchical disparity estimation for stereo image pair compression. In A N Akansu, editor, *Applications of SubBands and Wavelets*, page TBD, NJIT ECE Dept, University Heights, NJ 07102, March 1994. IEEE, IEEE.

[17] Sriram Sethuraman, M. W. Siegel, and A. G. Jordan. A multiresolution framework for stereoscopic image sequence compression. In *Proceedings of the 1994 IEEE International Conference on Image Processing*, volume II, pages 361–5. ICIP'94, IEEE Computer Society Press, November 1994.

[18] B. Julesz. *Foundations of cyclopean perception.* University of Chicago Press, 1971.

[19] Jeffrey S. McVeigh, M. W. Siegel, and Angel G. Jordan. Adaptive reference frame selection for the predictive coding of generalized video signals. In *Proceedings of the 1996 SPIE/IS+T Conference (San Jose)*, page TBD, Bellingham WA, January 1996. SPIE/IS+T, SPIE/IS+T.

[20] R. Chassaing, B. Choquet, and D. Pele. A stereoscopic television system (3d-tv) and compatible transmission on a mac channel (3d-mac). *Signal Processing: Image Communication*, 4(1):33–43, November 1991.

[21] M. E. Lukacs. Predictive coding of multi-viewpoint image sets. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume I, pages 521–4, Tokyo, 1986. IEEE.

[22] A. Puri, R. V. Kollarits, and B. G. Haskell. Stereoscopic video compression using temporal scalability. In *Proceedings of the SPIE International Conference on Visual Communications and Image Processing*, volume 2501, pages 745–56, Taipei, Taiwan, May 1995. SPIE.

[23] A. Schertz. Source coding of stereoscopic television pictures. In *Proceedings of the IEE International Conference on Image Processing and its Applications*, pages 462–4, Maastricht, Netherlands, April 1992.

[24] H. Aydinoglu and M. H. Hayes. Compression of multi-view images. In *Proceedings of the IEEE International Conference on Image Processing*, volume 2, pages 385–9, Austin TX, November 1994.

[25] S. Safier and M. W. Siegel. 3d-stereoscopic x windows. In *Proceedings of the 1995 SPIE/IS+T Conference (San Jose)*, pages 160–7, Bellingham WA, February 1995. SPIE/IS+T, SPIE/IS+T.

[26] J. McVeigh, V. S. Grinberg, and M. W. Siegel. A double buffering technique for binocular imaging in a window. In *Proceedings of the 1995 SPIE/IS+T Conference (San Jose)*, pages 168–75, Bellingham WA, February 1995. SPIE/IS+T, SPIE/IS+T.