The Video Z-buffer: A Concept for Facilitating Monoscopic Image Compression by exploiting the 3-D Stereoscopic Depth map

Sriram Sethuraman[1] and M. W. Siegel[2]

[1]David Sarnoff Research Center, Princeton, NJ 08543-5300

[2]The Robotics Institute/School of Computer Science,
Carnegie Mellon University, Pittsburgh, PA 15213
(E-mail: sriram@earth.sarnoff.com, mws+@ri.cmu.edu)

ABSTRACT

Compression can be achieved by exploiting knowledge both internal and external to a given image or video source. In this paper, we present means for generating and exploiting the specific external knowledge of a 3D stereoscopic depth map of the given scene to compress the given monoscopic source. Several instances in which the depth map can potentially increase compression or provide improved functionality are presented to motivate further work along this line of reasoning.

1. INTRODUCTION

The primary goal of any image (2D) or video (2D+t) compression algorithm is to generate a representation of the source that is smaller than the source's raw bitmap. The ability to compactly represent the source implies knowledge, about the source content, that is in some sense deeper than the raw pixel intensity arrays. In this paper we distinguish between the *internal* and *external* knowledge about the source. Both of them facilitate description, and thus enable compression.

Internal knowledge is knowledge that can be inferred from the source intensity array itself. Conventional waveform-based coding methods use internal knowledge to extract statistical redundancies within the intensity array to achieve a more compact representation of the array. The compression factors achieved by exploiting internal knowledge alone range between one and two orders of magnitude over the raw uncompressed representations.

To achieve higher compression, lower complexity, or the ability to interactively manipulate scene objects, certain *a priori* or external knowledge about the scene is required. This external knowledge is separate from the 2D or 2D+t arrays of luminance and chrominance. For instance, the knowledge that the nature of the scene in a typical videophone sequence is of the *head and shoulders* type against a constant background suggests particular strategies and tactics for achieving

deep yet psychophysically pleasing compression. For that matter, lossy coding to suit the human visual system's properties is in itself employing the external knowledge that the processed source material will be viewed by humans.

In this paper we propose a concept for generating and using a specific type of external knowledge, an auxiliary depth map, that can potentially result in higher compression or simpler implementation, while providing a better handle on scene objects in conjunction with other internal or external knowledge about the scene.

2. THE VIDEO Z-BUFFER

In computer graphics an important example of an overlay of external knowledge is the Z-buffer. It is used to facilitate efficient rendering of a scene that contains multiple potentially occluding facets or objects. The Z-buffer is simply a set of bit planes that parallel the red, green, and blue color bit planes which records the depth of each pixel with respect to the screen. During rendering, a pixel's RGB values are updated only if the depth of the update under consideration is less than the depth of the facet that generated the current values. This ensures that the closest object occludes the farthest, irrespective of the order in which the facets are examined.

Monoscopic images and video cannot unambiguously provide the depth of each visible pixel in an imaged 3D scene. However the relative depths of the objects in the scene can potentially lead to better representation of the monoscopic source, as will be elaborated in the next section. We propose to introduce depth as the external knowledge by employing an inexpensive (perhaps monochrome or lower resolution) camera alongside the studio quality camera used to obtain the monoscopic source under consideration. Through 3D stereoscopic disparity analysis between the two camera sources, the relative depths or disparities of the visible pixels can be computed. We refer to such a depth map as the video Z-buffer. Figure 1 illustrates this concept.

Depending on the way in which the depth map is exploited to represent the monoscopic

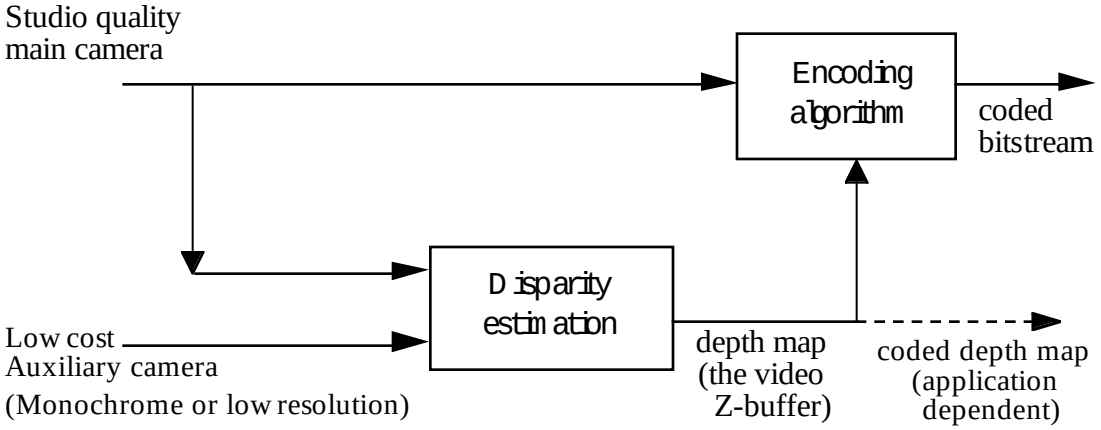


Fig. 1: Schematic of the proposed video Z-buffer based monoscopic image compression scheme

sequence, the depth map may or may not be required at the decoder. If the depth map is transmitted to the decoder, it could also be used to synthesize a second perspective for 3-D stereoscopic viewing. However, in this paper's context, stereoscopic broadcast is a side issue that we do not pursue.

3. POTENTIAL APPLICATIONS OF THE VIDEO Z-BUFFER

In this section, we present several scenarios in which the knowledge about the depth map can be exploited to achieve a more compact representation of the monoscopic source when used in conjunction with other *a priori* knowledge about the scene. For this section, it is assumed that a stereoscopic disparity map of the required accuracy has been estimated given the main and auxiliary video sources. Application-specific issues and increase in complexity due to the proposed setup are discussed in Section 4.

Depth-based bit allocation:

If the depth range of interest for the scene under consideration is known, then the bit allocation during intraframe coding or residual coding can be tailored to depend on the depth of the visible pixel. For example, in a videophone application, the object of interest is much closer to the camera than the background is. Hence for a given bit budget, more bits can be targeted for the nearer areas of interest, and the regions farther from the camera can be coarsely quantized.

Stationary background extraction:

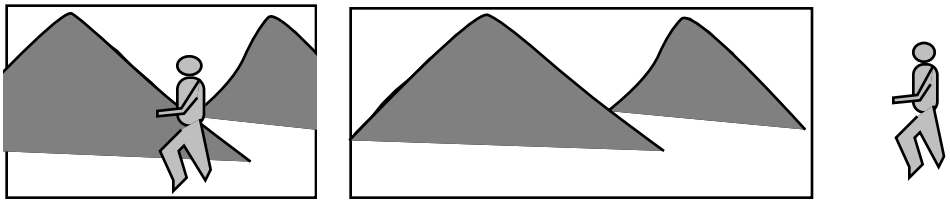
For scenes captured using a fixed camera with only object displacements within the scene, the background is stationary. As the foreground (or mid-ground) objects undergo displacement, they occlude and expose portions of this stationary background. The occluded and exposed regions are irregular in shape, and if coded individually for every frame would require a considerable number of bits. However, if the background regions can be extracted or accumulated over time and coded as a single image, the object displacements can be overlaid on top of the extracted background. Since the background typically covers a large portion of the scene, a significant saving in bit-rate over time is achieved as the background is coded only once.

Knowledge of the depth map renders the construction of the stationary background almost trivial. Motion-based occlusions do not pose a problem if the main and auxiliary camera views are obtained at the same time instant. Using a simple depth-based thresholding scheme, the background region can be segmented out. As the background is stationary, no registration over time is needed. Any pixel with an unassigned value in the background buffer can be updated if the depth is more than a predetermined distance. The foreground objects against this background can be segmented based on their depth, and can then be tracked over time. The background buffer, along with the tracked primitives, provides a very compact representation of the scene. Even if the static background buffer is maintained at a very high spatial resolution, over time its cost is negligible. Thus the knowledge of depth, along with additional knowledge about the nature of the

scene, increases the compression efficiency. However, a delay equal to the time needed to accumulate the background is introduced. Furthermore, if the background changes slowly but significantly on the time scale of a scene's duration, then the issue of updating the buffer in the least unobtrusive fashion will have to be eventually addressed.

Sample frame | Extracted background | Object to be tracked over time and overlaid

Fig. 2: (a) Stationary background extraction using depth



Sample frame | Background mosaic (assuming translational camera motion parallel to the scene) | Extracted object (trajectory over time to be estimated)

Fig. 2: (b) Nonstationary background identification and mosic construction

Nonstationary background reconstruction:

Unlike in the previous case, if there is camera motion also, then the background will not be stationary, and it will also have to be tracked over time. This requires thresholding based on depth to identify the background within a frame, and registration of the background from frame to frame, to construct the background mosaic. Several mosaic construction and camera motion stabilization methods are presented in [2, 4]. Once the background mosaic has been obtained, it can be compressed as a single still image; as in the stationary case, the cost over time of maintaining even a very high resolution mosaic is very small. Once again the knowledge about the depth greatly simplifies the identification of the background. Compared to the layered coding approach reported in [3], the availability of the depth map can considerably reduce the high computational complexity.

Depth based region grouping and applications:

The recent increase in the use of a mix of synthetic and naturally generated imagery and the rapid proliferation in the creation, storage, and network-based retrieval of digital video content has led to a desire to have other functionalities in addition to coding efficiency, such as the ability to

interactively manipulate scene objects, and easy indexing / retrieval capabilities, as reflected by the synthetic/natural hybrid coding (SNHC) proposals called for by the MPEG-4 standardization committee [1]. For instance, the task might be to replace the natural background with a synthetic background or to render the given scene from a different perspective or to retrieve a specified object in the scene from a video database. These added functionalities require an object level understanding of the scene as opposed to a pixel or region level understanding.

Conventional object-oriented coding strategies employ internal features such as intensity, color, and motion to segment regions in the image. The homogeneities based on these attributes typically lead to higher coding efficiencies. However, the segments obtained using these criteria do not usually correspond to entire physical objects in the scene, but to only parts of them. This gives rise to an increase in complexity as the number of primitives that need to be interactively handled is much higher than the actual number of physical objects. However, typically objects of interest in a scene are localized within a predictable depth range. Hence by using the external knowledge of depth, neighboring primitives extracted using other segmentation techniques that lie within a certain depth range can be merged to obtain a collective, larger primitive that corresponds to an entire physical object present in the scene. Hence a more compact representation of the scene is obtained, which reduces the complexity of the subsequent operations (e.g., indexing and retrieval) performed over the primitives.

We present below a potential application for such object-based representation. During motion estimation, global object motion can be estimated for each grouped primitive. Local displacements within an object can be refined for each constituent primitive. The knowledge of the entire object improves the accuracy of the global estimate. Also, removal of the global motion component provides a good coding model for the motion estimates of the constituent primitives, as their variance with respect to the global estimate is typically small. Since the object trajectories over time are estimated, advanced capabilities such as frame interpolation and synthesis become possible.

Depth based disambiguation:

In the scenarios considered so far, a suitable encoding strategy is chosen at the encoder given the depth map and other external information about the scene; the decoder does not require the depth map. Now we consider an application where the depth map is required both at the encoder and the decoder.

Sequence coding by tracking objects over time often provides an efficiently coded representation. When segmented objects in a frame are tracked over time, these objects occlude each other, and the background, due to object displacements. Unless these different objects are reconstructed at the decoder based on their decreasing depth order, regions that ought to be occluded would become visible and vice versa, thus resulting in a perceptually ambiguous picture at the decoder. In this case, the use of the depth map to decide on the region that should be visible is an exact analogue of the Z-buffer in computer graphics. It should, however, be noted that the advantage that segment tracking based coding brings must be evaluated against the increase in the

bit-rate arising from the need for transmitting the depth map. Segment tracking along with depth-based disambiguation was employed in a stereoscopic sequence coding context in [5]. A stereoscopic pair of frames were segmented based on disparity and inter-frame motion using quadtree decomposition and the segments so obtained were tracked from frame to frame in both the sequences over time till the next segmented frame. This scheme typically required a lower bit-rate than a conventional segmentation based block-matching scheme, at low quality settings. At higher quality settings, the savings in bits due to segment tracking were more than offset by the coding overhead needed to represent the background regions (both location and texture) that are uncovered. We predict that when combined with the nonstationary background mosaic creation, segment tracking would significantly outperform block-matching.

Motion parallax:

The availability of a depth map at the decoder could also be used to synthesize views that lie between (and even slightly beyond) the perspective spanned by the main and auxiliary source cameras. This requires suitable scaling of the disparity map, detection of occluded areas and areas with disparity estimation errors, and suitable prediction/interpolation strategies for obtaining a disparity estimate for such areas [5, 6]. A brief review of different synthesis techniques along with a new efficient scheme for synthesis from block-based disparity maps can be found in [6, 7]. With additional hardware and software at the receiver to track a viewer's head movements and to compute the correct perspective view for the head (monoscopic) or eye (stereoscopic) positions, motion parallax information, or the feeling of *look-around*, could be provided to the viewer. Our experiments demonstrate that pleasing simulations of look-around can be created with existing commercial headtracking systems; however, cheaply computing the intermediate views at video frame rate will have to wait for one or two more generations of computer hardware evolution. We routinely demonstrate to visitors to our lab an interface[1] [6] that, given two horizontally separated still image frames, allows the user to select at one extreme a single hyperstereoscopic still image without look-around, at the other extreme a continuum of synthesized monoscopic images that effectively stimulate the perception of look-around motion parallax anywhere between the two given views, and between these extremes trades off between a continuum of stereoscopic image pairs with both less than the extreme stereoscopic disparity and less than the extreme look-around motion range. While our own current hardware requires that we use pre-computed intermediate views for this demonstration, existing top end graphics workstations could do all the required computations on the fly.

4. IMPLEMENTATION AND APPLICATION SPECIFIC ISSUES

In this section we briefly consider certain application / implementation specific issues associated with the proposed setup in Fig.1. First, in addition to the auxiliary camera, the proposed depth map generation requires an additional digitizer and frame store for the auxiliary camera view

[1] The interface was developed primarily by J. S. McVeigh.

at the encoder. The complexity of the disparity estimation module and the associated storage requirements for the depth map depend on whether a dense or a sparse depth map is required by the application. Also, in applications that require transmission of the depth map, a compact representation of the depth map is vital. We have employed a disparity-adaptive segmentation method to create a sparse representation for the disparity (depth) map in [5]. Also it has to be borne in mind that depth values for pixels in the main view that are occluded in the auxiliary view cannot be estimated, which can affect the performance of the algorithm that uses the depth map. The frequency of update of the depth map also depends on the application, and would have an impact on the cost and the nature of the hardware overhead required by the setup.

As a practical matter, the most efficient and economical way to implement this concept might be to generate the disparity map "open loop" with respect to the studio quality camera, using two monochrome low resolution outrigger cameras and PC performance level frame capture and computing hardware, while relying mostly on robust mechanical alignment to assure spatial correspondence between the studio quality main image stream and the depth map stream.

5. CONCLUSIONS

We have presented the concept that external knowledge in the form of depth information obtained by using an auxiliary (possibly inexpensive) camera can be taken advantage of to achieve a more efficient and compact representation of a monoscopic image or video source. Several potential applications of the depth map that can enhance the coding efficiency while providing additional desirable functionalities have been presented. We hope to substantiate our claims with future experiments and simulations.

REFERENCES

[1] ISO/IEC JTC1/SC29/WG11, "MPEG4 proposal package description", no. 937, "Call for proposals", no. 943, March 1995.

[2] M. Irani et al., "Video compression using mosaic representations", Signal Processing: Image Communications, Vol. 7, no. 4-6, pp. 529-552, Nov. 1995.

[3] J. Y. A. Wang and E. H. Adelson, "Representing moving images with layers", IEEE Transactions on Image Processing, vol. 3, no. 5, pp. 625-638, Sep. 1994.

[4] R. S. Jasinschi and J. M. F. Moura, "Content-based video sequence representation", Proc. of Intl. Conf. on Image Proc., Oct. 1995, Vol. 2, pp. 229-232.

[5] S. Sethuraman, "Stereoscopic image sequence compression using multiresolution and quadtree decomposition based disparity- and motion-adaptive segmentation", Ph.D. Thesis, Dept. of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, July 1996.