INTERNATIONAL ORGANIZATION FOR STANDARDIZATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO-IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND ASSOCIATED AUDIO

ISO-IEC/JTC1/SC29/WG11
MPEG 95/
July 1995

TITLE: A Multi-Viewpoint Digital Video Codec utilizing Computer Graphics Tools & Image Warping Tech. based on Perspective Constructions of Intermediate Viewpoint Images

PURPOSE: An Efficient MPEG-2 Compatible Coding Scheme for Multiview Videos

SOURCE: Belle L. Tseng and Dimitris Anastassiou, Columbia University

1 MULTIVIEW PROFILE PROPOSAL

A multiview video is a 3D extension of the traditional movie sequence, in that there are multiple perspectives of the same scene at any time instance. Comparable to a movie made by a sequence of holograms, a multiview video offers the similar look-around capability. An ideal multiview system allows any user to watch a true 3D stereoscopic sequence from any perspective the viewer chooses. Such a system have practical uses in interactive applications, medical surgery technologies, educational and training demonstrations, remote sensing developments, and a step towards virtual reality.

We propose an efficient video coding scheme to accommodate transmission of multiple viewpoint images at any one instance in time. With the goal of compression and speed, a novel approach is presented incorporating a variety of existing tools and techniques. Construction of each viewpoint image is predicted using a combination of perspective projection of 3D models, 3D texture mapping, and image warping. Foreseeable uses for this coding technique include all applications requiring multi-viewpoint video transmission on bandwidth-limited channels.

With the development of digital video technology, an ISO standarization of video compression codec (MPEG-2) has recently been achieved. The coding standard is specified for one sequence of video, but has also recently been shown to be applicable to two sequences of stereoscopic signals. Extending the number of viewpoint videos beyond two views thus requires an intelligent and novel extension to the current MPEG-2 specification.

Under the block-based constraint of MPEG-2, one disparity vector corresponding to each block of an image can be incorporated and transmitted as a motion vector. However for acceptable predictions of multi-viewpoint images, accurate disparity values for every pixel is required for a more continuous interpolation of intermediate viewpoint images. Thus a disparity/depth map is required for every pixel, whose values can be transmitted as the gray-level intensity of a second "image". Also since the majority of the depth image is quite flat, the depth map can be considerably compressed. Finally a third signal is transmitted consisting of the residual prediction error used for improving the construction of a selected viewpoint image.

The multiview profile will supplement the main profile of MPEG-2, thus one viewpoint is transmitted on the main profile bitstream. In our case, the central viewpoint sequence is transmitted on the main profile. Let N = number of viewpoint sequences, where the minimum number of viewpoints is equal to 3, $N >= 3$. The number of viewpoints must be extendable in the future, especially since the number of viewpoints is an issue in itself, thus accomodation of additional viewpoints should be incorporated as an essential feature.

One central image is designated to be the principal view in which the other multiviews are predicted from. The central image is denoted by I_{Center} from viewpoint V_{Center}. The central viewpoint, usually the middle viewpoint, is chosen so that its image has the highest collection of overlapping objects with each of the other viewpoint images. In this manner, the central viewpoint image can be used to interpolate and predict the most of the other multiviews.

The other multiview images are designed by I_X from viewpoints V_X. The minimum number of viewpoints for the proposed system is three. An example of four additional viewpoint positions is illustrated, where $V_X = V_{Left}, V_{Right}, V_{Top}, V_{Bottom}$ with respective images $I_X = I_{Left}, I_{Right}, I_{Top}, I_{Bottom}$. These camera-captured images available at the encoder are referred to as *real* multiviews, whereas images not directly taken by a camera, but derived by prediction or interpolation methods, are called *virtual* multiviews. Virtual pictures also consist of those viewpoint images in between two real cameras, thus having the virtual constructions allow the viewer to see a smoother video transition between two real views.

The image coordinate system corresponding to each viewpoint is defined as (X_i, Y_i), where viewpoint index $i = Center, Left, Right, Top, Bottom$. Let the global rectangular coordinate system (X, Y, Z) be defined as corresponding to the image coordinate system (X_C, Y_C) of the central viewpoint, where Z defines the orthogonal axis from the central image plane.

The viewpoint vector for some view index i is denoted as $V_i = [vx_i, vy_i, vz_i, va_i, vb_i, vc_i]$. The camera position is represented by the coordinates (vx_i, vy_i, vz_i), and the camera zooming parameter is described by va_i. In addition, the camera rotations are given by the horizontal panning angle vb_i and the vertical tilting angle vc_i. Thus the central viewpoint $V_C = [0, 0, 0, 1, 0, 0]$.

Starting with the central viewpoint image $I_C^t(X_C, Y_C)$ at time t, a depth value is calculated for every pixel point, thus forming a depth map image, named D_C^t, corresponding to the central image. A number of methods can be used to obtain the depth values, including depth calculation from defocused images, disparity estimation using the gradient approach between other viewpoint images, block correspondence matching between multiviews, etc., ... Consequently for every image point $I_C^t(x_C, y_C)$, there is a corresponding depth value $z_C = D_C^t(x_C, y_C)$. The set of 3D coordinate information (x_C, y_C, z_C) spanning the central image is thus similar to a 3D geometrical surface model and a corresponding texture map represented by the intensity values of $I_C^t(x_C, y_C)$.

Following, a generic mesh representation is obtained whose vertices correspond to the pixels of the central image, thus generating a 3D surface model of the scene. This geometric representation offers ease in interpolating different viewpoints. Similar to rendering approaches in computer graphics, given a 3D geometrical model and its associated texture intensity, every viewpoint can be effectively and efficiently constructed by simple geometric transformations followed by 3D texture mapping. Furthermore, interpolating virtual viewpoint images is facilitated and made feasible.

Utilizing computer graphics capabilities, currently available in hardware for speed purposes, texture mapping is the most appropriate and efficient method for rendering an image of a 3D surface structure. In texture mapping, textured images are mapped onto corresponding geometric meshed surfaces. In addition, texture mapping incorporates the fore-shortening effect of surface curvatures. The texture is a 3D function of its position in the object, and therefore the central viewpoint image represents a 3D texture function.

Furthermore, other tools from computer graphics can be incorporated for a better overall construction. For example, if the light source can be extracted from the available viewpoint images, illumination and shading can be included to improve the predictions.

As the channel bandwidth increases, the quality of the decoded multiview videos must also improve. Our approach is to increment the number of transmitted prediction error images at any one time t, thus associated with an additional non-central viewpoint image construction. Consequently, perfect reconstruction is always achievable with unlimited bandwidth.

2 STEPS for ENCODER

1. Find the central viewpoint, achieving the best predictions for the other viewpoint images. Usually the central viewpoint is the middle camera viewpoint position. The central image is denote I_C and the non-central images are designated by I_X, corresponding to central viewpoint V_C and non-central viewpoints V_X respectively.

2. Encode the central image I_C^t in the main profile of MPEG-2.

3. Transmit the encoded bitstream for I_C^t.

4. Decode the bitstream for I_C^t to determine the received central image, $\hat{I}_C^t$.

5. Calculate the depth map, named D_C^t, for the central viewpoint image I_C^t. One such depth estimation technique is depth from defocused images

6. Encode the depth map D_C^t.

7. Transmit the encoded bitstream for D_C^t.

8. Decode the bitstream for D_C^t to determine the received depth map, $\hat{D}_C^t$.

9. Obtain a 3D mesh/wireframe representation, called M^t, for the central image I_C^t of time t by associating each coordinates pair (x_C, y_C) with its corresponding depth value $D_C^t(x_C, y_C)$. Consequently, the 3D surface texture is derived from the 2D image, and a graphical model of the image scene is obtained.

10. Determine a non-central viewpoint, designated as V_X^t, from the set of original real viewpoints, in which the best construction of viewpoint image I_X^t is desired for time t. The selection of non-central viewpoint V_X^t is based on a round-robin schedule, where every viewpoint is alternatively selected at a period of $N - 1$, see Figures.

11. Transmit the selected viewpoint vector V_X^t for time t.

12. Predict the selected non-central viewpoint images, referred to as PI_X^t, by rendering the 3D mesh model M^t in the specified viewpoint V_X^t. First, construct the wireframe image from the mesh representation by simple geometric transformations with perspective projections. Then texture map corresponding areas of the central image onto the appropriate 3D wireframe substructures.

13. Calculate the prediction errors PE^t required for the final reconstruction of the selected viewpoint V_X^t, by examining the difference between the original image I_X^t and the predicted image PI_X^t.

14. Encode the prediction errors PE^t.

15. Transmit the prediction errors PE^t associated with the chosen viewpoint V_X^t.

16. Determine if the allocated bit rate allows transmission of an additional viewpoint prediction error. If bandwidth permits, goto step (8) and chose another non-central viewpoint. Otherwise, start all over from the beginning for the next frame at step (2).

3 STEPS for DECODER

1. Decode the bitstream of the central image, denoted $\hat{I}_C^t$, from the main profile of MPEG-2.

2. Store the decoded central image $\hat{I}_C^t$ in memory.

3. Decode the bitstream of the depth map, thereafter referred to as $\hat{D}_C^t$.

4. Obtain a 3D mesh/wireframe representation, called M^t, for the central image I_C^t by associating each coordinates pair (x_C, y_C) with its corresponding depth value $D_C^t(x_C, y_C)$. Consequently, the 3D surface texture is derived from the 2D image, and a graphical model of the image scene is obtained.

5. Store the 3D mesh model M^t for time t in memory.

6. Decode the selected viewpoint vector V_X^t for time t.

7. Store the viewpoint vector V_X^t selected for time t in memory.

8. Decode the prediction errors, denoted $\hat{PE}^t$, associated with the selected viewpoint V_X^t.

9. Determine the requested viewpoint vector from the user at the decoding system, assigned V_U^t.

10. Predict the user-requested viewpoint images (whether real or virtual), referred to as PI_U^t where index U designates the appropriate viewpoint V_U^t. The predicted images PI_U^t are constructed by rendering the 3D mesh model M^t in the specified viewpoint V_U^t. First, construct the wireframe image from the mesh representation by simple geometric transformations with perspective projections. Then texture map corresponding areas of the central image onto the appropriate wireframe substructures. This module is similar to that of the encoder.

11. Determine if the encoder selected non-central viewpoint V_X^t is similar to the user-requested viewpoint V_U^t.

12. If equal, $V_U^t = V_X^t$,

    - Reconstruction of the final image $\hat{I}_X^t$ is obtained by combining the prediction errors $\hat{PE}^t$ with the predicted images PI_X^t. This kind of reconstruction is defined as *Type I Prediction.*
    - Store the reconstructed non-central image $\hat{I}_X^t$ in memory.
    - Start at the next frame by returning to step (1).

13. If not similar, $V_U^t \neq V_X^t$, the following steps are carried out.

    - Given the user-requested viewpoint V_U^t, retrive from memory the image $\hat{I}_U^{t-f}$ corresponding to the nearest past image $\hat{I}_U$ reconstructed by a Type I Prediction.
    - Similarly, given the user-requested viewpoint V_U^t, retrive from memory the image $\hat{I}_U^{t-b}$ corresponding to the nearest future image $\hat{I}_U$ reconstructed by a Type I Prediction.
    - Form 3D mesh model M^{t-f} created at time $t - f$ using the positional information from D_C^{t-f} and I_U^{t-f}.
    - In parallel, form 3D mesh M^{t+b} created at time $t + b$ using the information from D_C^{t+b} and I_U^{t+b}.

- Generate the front mesh image MI_U^{t-f} for the image $\hat{I}_U^{t-f}$ by locating a grid of fixed points.
- Also, generate the back mesh image MI_U^{t+b} for the image $\hat{I}_U^{t+b}$ by locating a grid of fixed points.
- Warping between the front mesh M^{t-f} and the back mesh M^{t+b}, generate the intermediate mesh-predicted image MPI_U^t for time t.
- Finally, obtain a construction for $\hat{I}_U^t$ by combining the intermediate mesh-predicted image MPI_U^t with the predicted image PI_U^t. The method for this final combination is left to the decoding system or any image fusion techniques may be adopted here, eg. XOR operator, average, ...
- Construction of this final image $\hat{I}_U^t$ is termed as *Type II Prediction.*
- If other viewpoint images are requested by the user, the steps may be repeated again. Otherwise start at the beginning of step (1) to decode the next time frame.

4 ADVANTAGES

1. Tracking of image points are facilitated by knowing the real world 3D coordinates of each point.
2. Prediction of non-transmitted virtual viewpoints is possible and efficient using the proposed combination of computer graphics tool.
3. View Interpolation is fast and efficient because of hardware speed and hardware rendering capabilities.
4. A possible solution to object-based approaches for MPEG-4 is suggested by this method. Some MPEG-4 proposals have insinuated the construction of 3D object models from the image sources.
5. Takes advantage of many existing graphical tools and animation facilities.
6. Combining image processing with computer graphics capabilities.
7. Incorporating the fore-shorting effect of 3D surfaces is automatic by using 3D structures with image texture data.
8. Quality of multi-viewpoint video construction is proportional to transmission bandwidth.
9. Freedom to improve individual coding modules in accordance with advance technologies. Similar to MPEG-2's main profile, this codec does not prevent creativity and inventive spirit.
10. The flexibility of multi-viewpoint parameters supports a wide range of multiview video applications.

5 MULTIVIEW VIDEO PARAMETERS

- Picture Width
- Picture Height
- Frame Rate
- Bit Rate
- Buffer Size
- Number of Viewpoints
- Viewpoint Vectors

6 APPROACHES / TECHNIQUES / TOOLS

- MPEG-2 Video Coding Standard
- Determine Camera Viewpoints
- Depth from Defocus
- 3D Mesh Representation
- 3D Viewpoint Rendering Methods
- Texture Mapping
- Digital Image Warping