Multiview Image Compression using Algebraic Constraints

Chaitanya Kamisetty and C. V. Jawahar
Centre for Visual Information Technology,
International Institute of Information Technology,
Hyderabad, INDIA-500019
jawahar@iiit.net

Abstract—In this paper, we propose a novel method for compression of multiple view images using algebraic constraints. The redundancy present in multiview images is considerably different from that in isolated images or video streams. A three view relationship based on a Trilinear Tensor is employed for view prediction and residual computation. The geometric redundancy in the form of common world structure is exploited during this.

1. INTRODUCTION

A multiview imaging environment consists of an array of cameras which image the world from different positions and viewing angles. This kind of imaging is used in creation of virtual environments, tracking and surveillance applications, 3D reconstruction etc. Multiview imaging is also used for video-conferencing where an immersive and interactive environment is preferred. Video-conferencing using multiple cameras requires transfer of large amount of multiview image/video data. As the number of camera views increases, the size of the dataset increases linearly. Since all the cameras capture the same world scene the images in the dataset have considerable redundancy. The redundancy due to this has to be exploited to encode multiview images together to reduce the size of the dataset.

Traditional image compression aims to exploit the redundancy due to the spatial distribution of pixels or the limitations of the human perception mechanism. Multiview compression needs to incorporate an additional dimension of redundancy due to the overlap in scene structure. Video compression also addresses the problem of compression of multiple images. However, video frames are taken from the same viewpoint and a simple subtraction often removes the redundant information. In the case of multiview imaging, a variable compensation is necessary before subtraction.

The problems we address in this paper and the solutions we employ can be summarized as follows:

Problem 1: When viewed from distinct viewing points, 3D points get displaced depending on their depth. Simple subtraction as in mpeg results in large residuals, even if the world is stationary. We employ a multiview geometric constraint [7], to achieve variable compensations for image points.

Problem 2: When imaged from new viewing angles, the intensity/colour of a 3D point gets modified. Geometric compensation alone is not enough to achieve high compression. We additionally employ a brightness constraint [8] to address this problem. This uses an optical flow constraint for prediction of intensities in the novel view.

Stereo views can be considered to be the simplest multiview dataset. Currently, block based disparity compensated prediction methods are used to code stereoscopic views. In these approaches, the left view is coded independently and the right view is predicted from the left view using disparity compensation methods [4], [5]. The error between the right view and the predicted view is then encoded. When this method is extended to views taken from general positions, disparity needs to be estimated in horizontal and vertical directions, increasing the computational complexity and size of the residual. Attempts have been made to counter this through image rectification (which results in virtual parallel views) [1], [2]. But they require the camera parameters to be known (calibrated cameras). In [3], image alignment was done by estimating the fundamental matrix, but it relies on search techniques for disparity estimation which is again computationally expensive.

All the above mentioned algorithms assume either a parallel camera setup or known calibration. They cannot be used in a general multiview environment. Multiple view geometry is an active area of research in computer vision. Multiple view geometry [6] provides constraints that relate the features in multiview images. These constraints are dependent only on the internal parameters of the camera (like focal length) and their relative positions. They are independent of the scene structure. They take the form of the fundamental matrix for two views and the trilinear tensor for three views. In this paper, we present a compression scheme based on algebraic constraints in multiple views. Residual is defined to be the difference between an already available image and the algebraically predicted image. Base image, residual images and coefficients of the algebraic relationships between features are compressed together in the proposed scheme. Experiments are conducted to verify the applicability of the algorithm. Performance of the compression strategy is presented with the help of the statistics of the residual image. Coding of the residual images considering the geometric distortion is beyond the scope of the present paper.

0-7803-7651-X/03/$17.00

We briefly compare the different compression problems in Section 2, highlighting the special difficulties associated with the Multiview compression. The algorithm to code multiple views is described in Section 3. Use of brightness constraints is described in Section 4. Results of the algorithm are presented in Section 5. We conclude the paper in Section 6.

2. Comparison of Mono, Stereo, Video and Multiview Image Compression

Compression of images is a classical problem, with numerous algorithms being present for different classes of images. Mathematical tools starting from Fourier and Wavelet transforms to Neural Networks and Fractals are popular to solve this problem [9]. Traditionally the focus has been limited to exploit the human visual limitations. Performance of the algorithm may depend on the signal characteristics of the image. The challenge faced in compressing multiple images is considerably different from that of isolated images. Here, one has to take care of the overlap between the images in terms of content of the scene. In video, even if there are multiple frames, images differ due to the motion components of objects. However in stereo and multiview, the disparity/displacement varies across pixels and the traditional motion vector based approaches become insufficient.

Disparity compensated prediction for stereo image coding and motion compensation used in video coding are very similar. We take a cue for coding multiview images from this. However stereo, video and multiview images differ in certain characteristics as described below.

For typical video sequences where low bit rate coding algorithms are employed, background objects do not generally move from one frame to the next and only a small percentage of the scene undergoes motion. The displacement for the moving objects, which may be modeled as purely translational usually, does not exceed a few pixels. By contrast, every object in a stereo pair is displaced and the displacement may be large when compared to video sequences. As a consequence the performance of disparity compensation is lower than motion compensation applied for video coding. In case of multiview images, motion cannot always be modelled as simple translation of cameras.

Standard block matching algorithms such as MSE or MAD assume constant intensity along the displacement trajectory. While this is generally valid for video sequences, it is rarely true for stereo pairs and multiview images. In addition to the geometric errors, the reflected intensity at a point in these cases depends on the object surface properties.

Yet another difference is the source of occlusion. In video sequences, occlusion occurs due to moving objects. In stereo pairs and multiview images, occlusion occurs when some part of the scene can only be captured by one of the cameras due to finite viewing area, referred to as framing error.

3. Compression Procedure

Consider a multiview imaging environment where images of the scene are taken using cameras in different spatial positions. We assume that there is considerable overlap in the scene being imaged by adjacent cameras. Let $\mathbf{I} = \{I_1, I_2 \ ... \ I_n\}$ represent the set of images taken by these cameras. We select two images from $\mathbf{I}$ to be the base views. These views are coded independently using a stereo image compression algorithm. The third view is then predicted from the base views using the trilinear tensor τ [6], [7]. Prediction of I_3 is given by $\hat{I}_3$, which can be expressed as $\hat{I}_3 = f(I_1, I_2, \tau)$. Where τ, the trilinear tensor relates the views I_1, I_2 and I_3. Predicted image may differ from the observed image obtained from the camera, resulting in a residual. The residual is computed as $\text{Res}(I_3) = (I_3 - \hat{I}_3)$. Each of the remaining views is coded similarly using two views in the encoded set as the base views. Thus the basic algorithm needs to handle the compression of triplets of images. The algorithm to code three views I_1, I_2 and I_3 is described in the rest of this section.

Trilinear Constraints

The geometry of multiple view images is analyzed in detail in recent past [6]. It is being shown that the locus of the image of a world point in an image can not be arbitrary given its coordinate in one or more other views. This resulted in epipolar (for two views) and multiview geometry. Coordinates of corresponding points in two views are related by a fundamental matrix [6].

The trilinear tensor for three views plays a role analogous to the fundamental matrix for two views. It encapsulates the (projective) geometric relations between three views that are independent of the scene structure. The tensor only depends on the motion between views and the internal parameters of the cameras. This is defined uniquely by the camera matrices of the views. However, it can be computed from image correspondences alone without requiring knowledge of the motion or calibration [6]. The trilinear tensor τ is a $3 \times 3 \times 3$ representation with 27 elements. Derivation of the trilinear tensor constraint for three views is given in [7].

Let $\mathrm{P}(x, y, 1, \lambda)$ be a point in the 3D space that is projected onto 3 views with image points $p_1(x^1,\ y^1,\ 1), p_2(x^2,\ y^2,\ 1)$ and $p_3(x^3,\ y^3,\ 1)$ respectively. The equations relating these image points to τ can be represented using tensorial contraction as

$$p_1{}^i s_j^\mu r_k^\rho \tau_i^{jk} = 0 \tag{1}$$

Where $i, j, k = 1, 2, 3$ and $\mu, \rho = 1, 2$.

$$s_j^\mu = \begin{bmatrix} -1 & 0 & x^2 \\ 0 & -1 & y^2 \end{bmatrix} \quad r_k^\rho = \begin{bmatrix} -1 & 0 & x^3 \\ 0 & -1 & y^3 \end{bmatrix} \tag{2}$$

The tensor τ relates the points in three views and can be computed using corresponding feature points in three views. If τ is known, using image co-ordinates in two views we can predict the image point in the third view.

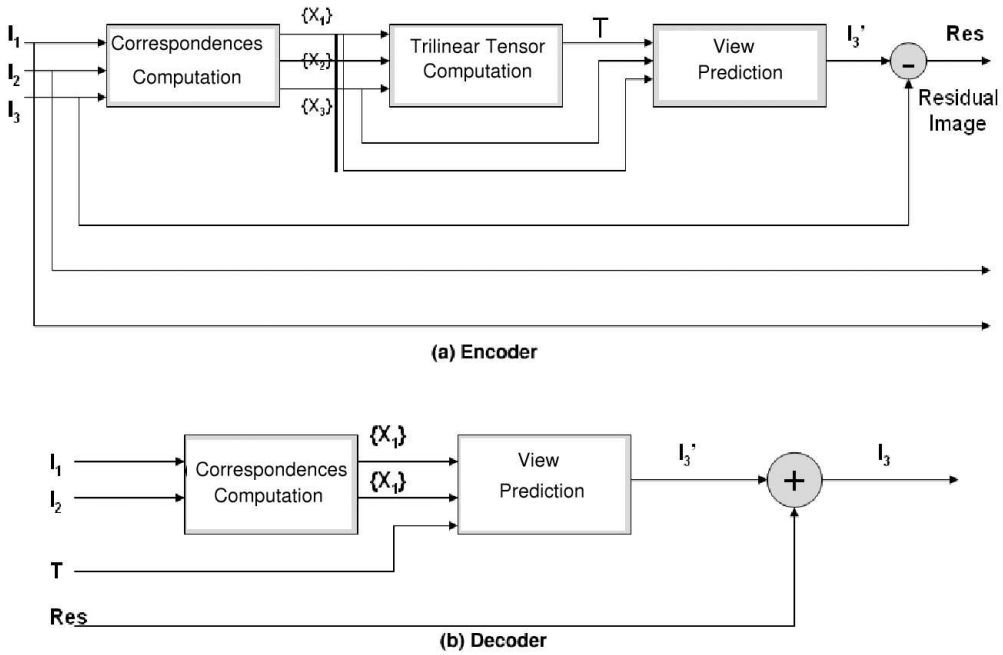

Figure 1. Overview of compression procedure

Prediction

Given two images, one could think of computing the 3D structure and projecting to the third camera for view prediction. However, this needs accurate calibration of cameras, which is impossible in uncalibrated environment. Instead, given only a weak calibration, we can predict the view using the algebraic relationships between corresponding points in these images [6]. We employ the trilinear relationships.

We first compute the corresponding feature points in the three views. Every corresponding triplet contributes four linearly independent equations represented by Eq. 1. We require a minimum of 7 corresponding triplets to determine τ. If we have more than 7 triplets, we get an overdetermined set of equations which can be solved using linear least squares method [6] to obtain the 27 coefficients of the trilinear tensor. The feature correspondences in I_1 and I_2 are projected to I_3 using τ. The feature points in I_1 are divided into non-overlapping triangles using Delanauy triangulation. These triangles in I_1 are then mapped to triangles formed from the predicted points in I_3 and corresponding textures are mapped. This texture mapped image gives $\hat{I}_3$, the prediction of I_3.

Residual Computation

Once the algebraic relations are computed, we represent the triplet of images using the trilinear tensor coefficients, stereo-compressed base images and the residual image.

By analyzing the properties of the residual image, we can conclude the amount of information present in it. The statistics (like mean intensity) of the residual image indicate how well the view was predicted from the base views. Residual image compression based on its frequency characteristics was studied in [4], [5]. However, the residual compression for multiview situation will have to take into consideration the effects on geometric reconstruction.

The encoding and decoding procedure is presented below. A pictorial representation of the three view compression is also given in Figure 1.

Encoding

1. Let I_1 be the base view. Hence it is coded independently using a still image compression algorithm.
2. Using a correspondence computation algorithm, we find a set of corresponding features between I_1 and I_2. These features are tracked to I_3. Using these correspondences, we compute the 27 trilinear tensor coefficients. Let the correspondences be represented as $\{X_i^1\}$ $\{X_i^2\}$ $\{X_i^3\}$
3. Using the tensor equations, for every pair of corresponding points in I_1 and I_2, we can predict a point in I_3. This is done for all the correspondences computed in step 3.
4. A triangulation algorithm is employed to find a triangle mesh of the predicted points in I_3. We use I_1 to texture map these triangles. Thus we obtain a predicted view $\hat{I}_3$ of I_3.
5. Residual of I_3, denoted by Res(I_3), is formed as $Res(I_3) = I_3 - \hat{I}_3$
6. I_2 is coded using a stereo compression algorithm. It is to be noted that the effectiveness of the stereo compression algorithm will depend upon the relative camera geometry of I_1 and I_2. To improve the results, we can encode I_2 similar

to I_3 i.e. find the triangle mesh of points in I_2 for which we know corresponding points in I_1 and texture map I_1 onto it to form the $\hat{I}_2$, the predicted view of I_2. This method in a way models the perspective distortion between the views and is expected to give a better prediction. But this method calls for storage/transfer of point correspondences between I_1 and I_2 as opposed to the motion vectors in the stereo algorithm case. Similar to the previous case: $Res(I_2) = I_2 - \hat{I}_2$

The output of the encoder consists of I_1, Res(I_2), motion vectors of I_2 or correspondences between I_1 and I_2, Res(I_3) and 27 trilinear tensor coefficients.

Decoding

1. Using either the motion vectors or the correspondences, we find the $\hat{I}_2$. Then I_2 is decoded using $I_2 = \hat{I}_2 + Res(I_2)$
2. It is assumed that the algorithm to find and track corresponding points across views is known at the decoder end. Hence we find correspondences in the 3 views as in step 2 of encoding.
3. We find $\hat{I}_3$ by following steps 3 and 4 in the encoding algorithm. I_3 is decoded as $I_3 = \hat{I}_3 + Res(I_3)$

4. Brightness Constraint

In the previous section, we have used the geometric redundancy between views to obtain compression. However, in practice, brightness of corresponding points also vary across views. This results in considerable increase in residuals.

The *optical flow constraint* provides a relationship between a point in one image and a line passing through the corresponding point in the second image. It is given by

$$u' I_x + v' I_y + I'_t = 0$$

where (u', v') are the optical flow values at (x, y) between image 1 and image 2 (i.e. $u' = x' - x$ and $v' = y' - y$). The spatial and temporal derivatives at the coordinates (x, y) are given by I_x,I_y and I'_t . In practice, $I'_t = I_2(x, y) - I_1(x, y)$. The geometric constraint discussed in the previous section can be combined with the optical flow constraints to result in a brightness constraint [8].

If a point p in image 1 corresponds to p' and p'' in images 2 and 3 and s', s'' are the lines passing through them, then the *tensor brightness constraint* is given by

$$s''_k s'_j p^i \tau_i^{jk} = 0 \tag{3}$$

where $s' = (I_x, I_y, I'_t - xI_x - yI_y)^T$ and $s'' = (I_x, I_y, I''_t - xI_x - yI_y)^T$ The derivatives I_x and I_y are computed at p' using the operator $[-1\ 0\ 1]$. For every point in the third view I''_t is computed using the tensor brightness constraint Eq. 3. Its brightness value $I_3(x, y)$ is then computed using

$$I_3(x, y) = I''_t + I_1(x, y)$$

Thus we can also predict the intensity of the points in the third view. The predicted third image is subtracted from the original image and the residual is compressed along with the base images as discussed in the previous section.

5. Results

We have explored the possibility of using algebraic constraints for multiview compression in the previous sections. We conducted some experiments to verify the applicability of the algorithm proposed in Section 3. We here show results on two synthetic datasets which were generated by simulating a multiview environment. The *Face* dataset consists of views of a texture mapped human face. Three of the views are shown in Figure 2 (a),(b),(c). The images are of size 600×528. The foreground pixels cover about 21% of the image. The *Teapot* set consists of views of a teapot obtained by rotating the camera. One of the views of the teapot is shown in Figure 2(g). The images in this dataset are of size 512×512 with around 14% foreground pixels. The teapot is illuminated by ambient and diffuse light and the surface does not contain any texture information.

After applying the compression procedure on the face data set, it was found that the mean gray value of the residual image (Figure2(d)) is only 8.3 where as the original image had a mean value of 151.7. It was also seen that the dynamic range of the residual was nearly halved from 0-255 to 0-135, and nearly 88% of the foreground pixels were in the range 0-14 as can be seen from the histogram shown in Figure 2(e). The statistics were computed only using the foreground pixels. This indicates that selection of an appropriate algorithm that can exploit the characteristics of the residual can lead to a good compression ratio.

In the teapot dataset, the original image had a mean gray level of 217.9 and the residual had a mean value of 9.5. Nearly 93% of foreground pixels in the residual were in the range 0-14. This shows that the predicted view was very close to the original view.

Brightness constraints described in Section 4 were used on the teapot dataset. Dense point to point correspondence of the visible points was used to test the algorithm. Use of brightness constraints resulted in smaller values of residual at most image points (closer to zero). The residual image shows certain bright regions (refer to Figure 2(h). This is because, the intensity value computed at points with large intensity gradients lead to a poor estimate of the brightness value. The use of brightness constraints can be improved by identifying points where its use is valid.

Statistics of the residual obtained in all the cases clearly shows that the predicted view is very close to the original image. This implies that by adopting an appropriate algorithm to code the residual, we can obtain good compression ratios.

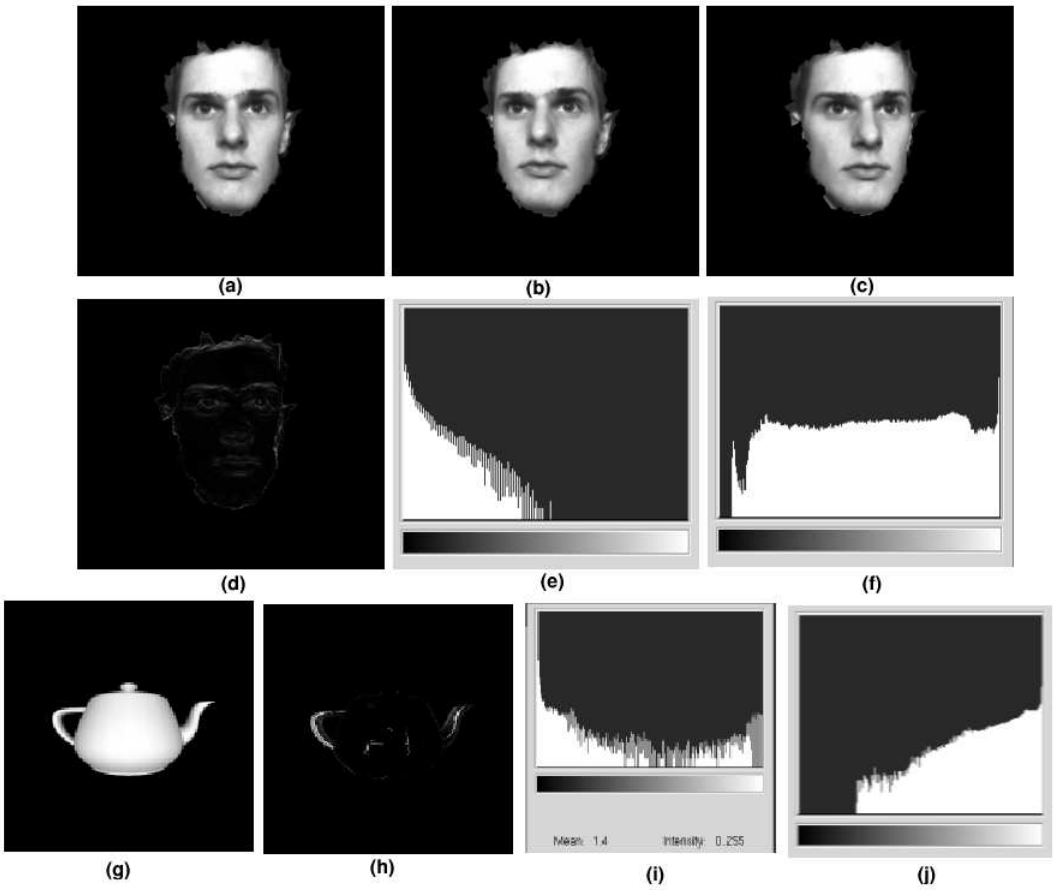

Figure 2. (a),(b),(c) : Three views from the *Face* dataset. (d) : Residual of image (c) obtained without using brightness constraints. (e) : Histogram of (d). (f) : Histogram of (c). (g) : A view from the *Teapot* dataset. (h) : Residual of (g) obtained using brightness constraint. (i) : Histogram of (h). (j) : Histogram of (g)

6. CONCLUSION

The paper describes an approach to compress multiple views using algebraic constraints. The evaluation of the algorithm has been carried out using the statistics of the residual image. It is seen that the residual carries considerably less information than the original image. In a true multiview environment, a more generic multiview constraint may be employed in place of a trilinear one. The set of images may be alternatively analyzed in a joint image space. Future work will focus on coding of the residual image and improving the use of brightness constraints.

REFERENCES

[1] D. Tzovaras, N. Grammalidis and M. Strintzis, "Object-based coding of stereo image sequences using joint 3-D motion/disparity compensation", *IEEE Trans. Circuits Syst. Video Technology*, vol 7, pp. 312-327, Apr. 1997.

[2] L. Falkenhagen, "Block-based depth estimation from image triples with unrestricted camera setup",*Proc. IEEE Workshop Multimedia Signal Processing,* Princeton, 1997

[3] Ru-Shang Wang, Yao Wang, "Multiview Video Sequence Analysis, Compression and Virtual Viewpoint Synthesis", *IEEE Trans. on Circuits and Systems for Video Tech.*, vol 10, April 2000

[4] Moellenhoff S. Mark, Maler W. Mark, "Transform Coding of Stereo Image Residuals", *IEEE Trans. on Image Processing*, June 1998

[5] Woontack Woo, Antonio Ortega, "Stereo Image Compression with Disparity Compensation Using the MRF Model"

[6] R. Hartley, A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, UK, 2000

[7] A Shashua *Trilinear Tensor: The Fundamental Construct of Multiple View Geometry and its Applications* Int. Workshop on AFPAC, 1997

[8] Stein P. Gideon, Shashua Amnon, *Model-Based Brightness Constraints: On Direct Estimation of Structure and Motion* IEEE Transactions on Pattern Analysis and Machine Intelligence, September 2000

[9] Zhang, Y.Q. and Li, W.P. and Liou, M.L., "Special Issue on Advances in Image and Video Compression", *Proc. IEEE*, 83(2),1995, pp 135-138