A LIVE PANORAMIC VIEW CREATION ON VIDEOS USING FIRST ORDER STATISTICS

Baskar .A[1], Prabukumar .M[2], Christopher Clement .J[3]
[1]Amrita Vishwa, Vidyapeetham University,India
[2,3]VIT University
E-mail : [1]a_baskar@ettimadai.amrita.edu

Abstract

The automatic construction of panoramic view for video is an active area of research in computer vision and it plays an important role in video surveillance, large aerial and satellite videos. More recent applications include scene stabilization, video indexing and increasing the field of view. In this paper a novel video stitching algorithm has been proposed. This algorithm is tested on three types of datasets. First dataset comprises of overlapping videos having identical lighting conditions. Second dataset consists of different lighting conditions. Third dataset consists of videos of various overlapping conditions. The proposed algorithm gives better results compared to the existing algorithms. It also is capable of handling different lightness conditions and time synchronization of input video sequences. However, under rotation and transformation the algorithm fails to produce a good panoramic view.

Keywords: Erasure decoding, Key Distribution, MDS Codes, Multicast.

I. INTRODUCTION

A panoramic view is the process of creating panoramic video from several partly overlapping digital video pictures. With the increased popularity of digital cameras, this is an interesting concept not only to the professional graphics artist but also to the consumer. Video stitching methods have application plays an important role in other area such as video surveillance, large aerial and satellite videos from sequence of frames. More recent applications include scene stabilization, video indexing and increasing the field of view. In the research literature methods for automatic panoramic view creation fall broadly into two categories approaches that use pixel-to-pixel matching are called direct methods [1, 2]. Approach that use interest points matching are called feature based methods [5, 6].

Direct based methods have an advantage that they use all the available data and hence provide very accurate registration. In these approaches pixel-to-pixel matching is used, and hence they consume time to create a panoramic image. Feature based methods extract distinctive features from each sequence of image. In these approaches, extracting distinctive features take more time. In this work we focus on direct based methods for matching sequence of video frames [3, 4], our algorithm uses mean intensity variation instead of pixel-to-pixel matching to identify the overlap region, and also the algorithm creates good panoramic view, even the sequences of video having different intensity. The rest of our paper is organized as follows. Section 2 reviews related work. Then, in section 3, we present our optimization algorithm in detail. Experimental results in this paper are presented in section 4. We conclude our paper in section 5.

II. A DIRECT BASED APPROACH

A. Error metrics

The simplest way to establish an alignment between two images is to shift one image relative to the other. Given a template image $I_0(x)$ sampled at discrete pixel locations }xi = (xi, yi)…we wish to find where it is located in image $I_1(x)$. A least-squares solution to this problem is to find the minimum of sum of the squared differences (SSD) [1, 2] .It is expressed in Equation (1).

$$E_{SSD}(u) = \sum_i [I_1(x_i + u) - I_0(x_i)]^2 = \sum_i e_i^2 \tag{1}$$

where u= (u, v) is the displacement and ei = I_1 (x_i + u) “ I_0 (x_i) is called the residual error. In general, the displacement u is fractional, so a suitable interpolation function is applied to image I_1(x). In practice, a bilinear interpolation is often used, but bi-cubic interpolation yield slightly better results. Color images are processed by summing differences across all three color channels.

B. Robust error metric

We can make the above error metric more robust to outliers by replacing the squared error terms with a robust function p(ei) [7] to obtain the Equation (2)

$$E_{SRD}(u) = \sum_i \rho(I_1(x_i + u) - I_0(x_i)) = \sum_i \rho(e_i) \tag{2}$$

The robust norm p (e) is a function that grows less quickly than the quadratic penalty associated with least squares. One such function, sometimes used in motion estimation for video coding because of its speed is the sum of absolute differences (SAD) metric [14]; it is shown in the Equation (3)

$$E_{SAD}(u) = \sum_i |I_1(x_i + u) - I_0(x_i)| = \sum_i |e_i| \qquad (3)$$

C. Hierarchical motion estimation

In this approach we have defined an alignment cost function to optimize, how do we find its minimum? The simplest solution is to do a full search over some range of shifts, using either integer or subpixel steps. This is often the approach used for block matching in motion compensated video compression, [7] where a range of possible motions (say Q16 pixels) is explored. To accelerate this search process, hierarchical motion estimation is often used, where an image pyramid is first constructed, and a search over a smaller number of discrete pixels (corresponding to the same range of motion) is first performed at coarser levels. The motion estimate from one level of the pyramid is used to initialize a smaller local search at the next finer level. While this is not guaranteed to produce the same result as full search, it usually works almost well and faster.

More formally, let

$$I_k^{(l)}(x_j) \leftarrow \tilde{I}^{(l-1)}(2x_j) \qquad (4)$$

Is the decimated image at level l obtained by sub sampling (down sampling) a smoothed (prefiltered) version of the image at level *i*-1. At the coarsest level, search for the best displacement $u^{(l)}$ that minimizes the difference between images $I_0^{(l)}$ and $I_1^{(l)}$. This is usually done using a full search over some range of displacements (where S is the desired search range at the finest (original) resolution level).

III. A LIVE PANORAMIC VIEW

Live panoramic view is the process of creating panoramic video from several partly overlapping digital video pictures. A Live panoramic view system is an integration of three modules, namely pre-processing, panoramic algorithm, video stitching and panoramic view generation. In the pre-processing step the input which is the captured sequences of video are pre-processed. These include steps such as color conversion (RGB to HSV and HSV to RGB) and time synchronization information of video frames. In the panoramic algorithm module a direct based method has been used to identify the overlap region between sequences of video. In the final stage the homography matrix is identified and used to merge the overlap region and to create the panoramic image. The input videos are then merged and a panoramic view of video is created from the stitched images.

The entire system has been classified as two stages. In the first stage, it will embrace the entire module present in the system. Here only the first frame has been taken from sequences of video. Suppose, if the video is not time synchronized, the first best match frame has been taken from sequence of video using time synchronization algorithm. For every 15 minutes of the video, this stage will call and identify the overlap regions. In the second stage, the system will embrace the preprocessing module (only getting frames and pixel values), the video stitching algorithm modules and panoramic view of video. The both stages are illustrates in Figure 1 and Figure 2.

A. Pre-processing

Sequences of video having overlap are given input as this pre-processing module. Pre-processing stage includes various modules as follows

- Color conversion module
  - o RGB to HSV color conversion
  - o HSV to RGB color conversion
- Time synchronized module.

1. Color conversion module.

A Live panoramic view is constructed from sequences of video having overlap region. Normally sequences of video will not assure that it having same brightness or lightness condition. Handling Different Brightness or lighting condition video is one of challenges area in video stitching. RGB color model provided misleading results for this kind of problems.

Choosing a suitable color space from the start is crucial to system robustness. Although color information in computer applications is usually kept in the RGB (Red, Green, Blue) color space, this type of description is not the most convenient or intuitive for our purposes. Instead, we choose to convert all color

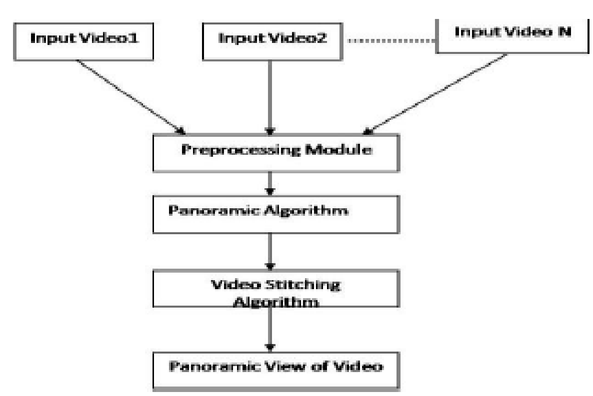

Fig. 1. Overview of the live panoramic view system, first stage

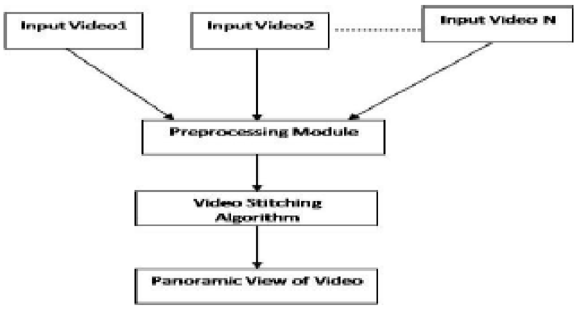

Fig. 2. Overview of the live panoramic view system, second stage

Information to the HSV (Hue, Saturation, Value) color space. In the RGB representation, each color is described by three 8-bit values (three channels), corresponding to the amount of red, green and blue in a pixel. We think of the RGB space as a 3- dimensional cube, with red, green and blue as the axes, and different positions within the cube representing different colors. A drawback to the RGB representation is that the relative location of different shades of the same color (i.e., light blue to dark blue) may or may not be known, or even continuous. Thus, we need a color space where similar color shades cluster together within the color space in a continuous pattern. The HSV color space is a non-linear transformation from RGB, where colors are represented by angular coordinates along a cone, rather than Cartesian coordinates. Conversions are shown in equations (5) and (6).

$$v = \max(r, g, b) \tag{5}$$

$$s = \frac{\max(r,g,b) - \min(r,g,b)}{\max(r,g,b)} \tag{6}$$

The hue channel is the “tint” or “tone” of the color, and is an angle along the circumference of the color cone; the angle distinguishes whether a color is blue, yellow, purple, etc.

The saturation channel is a measure of the amount of white in a color, or its “brightness”, which corresponds to the radial distance from the center axis of the cone. Saturation is the amount of black in a color, or its “intensity”, which is measured along the vertical axis of the cone. Thus, similar shades of the same color cluster together nicely in this color space, since their hue would remain constant, and only the saturation or value channels would vary. This property of the HSV space allows us to ignore changes in lighting intensity on an object, since we choose to put more importance on the hue channel and less importance on the other two channels.

2. Time synchronized module.

It plays very important role in live video panoramic view. To create a panoramic view, it is important to use synchronized input video sequences. Otherwise, the live panoramic module wrongly uses the frames of different time instance to compose the panoramic frame and introduce discontinuity into the resultant panoramic frame. In practical scenario, it is not sure that the sequences of video will start capturing the object at same time instance. In this case applying a video stitching algorithm directly, they have provided misleading output. The time synchronization algorithm has been discussed in detail below.

ALGORITHM

Step 1 : Get the Length from video A and Video B

Step 2 : Compare video A and B length

Step 3 : If length of video A is greater than Video B then

Step 4 : Assumption have made video A starts Capture the Object first.

Step 5 : Increment the video B frame number and Compare with video A.

Step 6 : Repeat the step 5 until the comparison results will go over threshold value

Step 7 : Else length of video A is less than video B Then

Step 8 : Assumption have made video B starts Capture the object first.

Step 9 : Increment the video A frame and compare with video B.

Step10 :Repeat the step 9 until the comparison results will go over threshold value

Step 11 : The time synchronization module will return best Overlapping frames from Video A and video B.

B. A panoramic algorithm

An algorithm is proposed for getting an input as preprocessed video sequences, and the algorithm provide overlap region as an output to the video stitching algorithms. A panoramic algorithm has been includes two different modules as follows

- Similarity measures
- Identify the overlapping region

In this section, a detailed description is provided about every module present in the panoramic algorithm and followed by detailed algorithm description

1. The similarity measure.

The similarity measure function, or cost function, gives an indication of the similarity between two compared video frame regions. The function either is on direct pixel intensity comparisons, or mean intensity variation, or on other geometrical features within the regions. In this work, focus is put on mean intensity variation comparisons. The mean intensity variation is described below.

Mean value= }}(x-1), (y-1)…+ }(x-1), (y)…+ }(x-1), (y+1)…+ }(x), (y-1)…+ }(x), (y)…+ }(X (y + 1)…+ }(x+1), (y-1)…+ }(x+1), (y)…+ }(x+1), (y+1)……/9; (7)

To identify the overlapping region, last three columns from video A frame and first 80% of columns from video B frame, have been considered. Here both the video A and video B input images has been resized into 200x200 image. The similarity measures have been described in the following ways.

The mask has been placed in top right corner of the video A. This mask is used to compute the mean value of frame, from video A. Another mask has been placed in top left corner of the video B. It computes the mean value of frame, from video B. This is illustrated in Figure 3. The mean intensity difference is used to evaluate similarity measures. If both frames from video A and video B mean values are close to similar, it indicate that both input frames are having the same object information. Otherwise both input frames are having different object information.

In first step, the mask from video B has been moved around the horizontal direction upto 80% of columns and computes the mean values and the mean intensity difference with video A frame. This intensity difference returns the similarity measure. This is illustrated in Figure 3. Once the horizontal moves has been completed, in the second step the mask from video A has been moved in vertical directions in every I+3th row. Here „I is referring to the current frame. After every vertical moves, it calls again first step. This is illustrated in Figure 4.

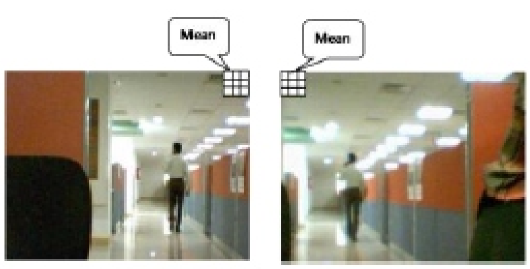

Fig. 3. Finding similarity Measure for overlap input videos, step1

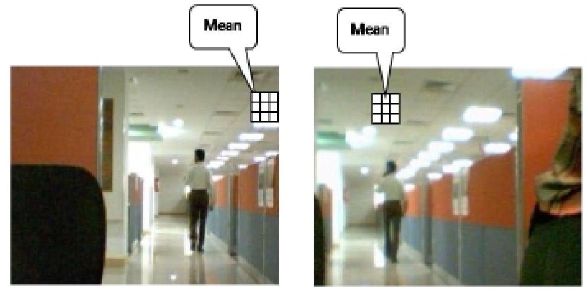

Fig. 4. Finding similarity measure for overlap input videos, step2

3. Identify the overlapping region.

The next stage of the live panoramic algorithm is called as identify the overlapping region. The previous stage of algorithm gives set of column values as an output to every vertical move. Set of these column values given as an input to this stage. The output of this stage is precise overlapping column value. The detail description of this stage provided bellow. Once the set of putative correspondences have been determined, identify the overlapping region is now employed. This stage further classified into two sub stages. In first sub stage, the algorithm amasses set of column values for every vertical move from previous stage called similarity measures is sorted form. In next sub stage, the algorithm is used to count maximum recurring column value. This column value signifies, upto this column value, the second image reiterate information from first image. These regions are called as an overlap region.

ALGORITHM

Step 1 : A mask placed on top right corner of the video A frame and another mask placed on top left corner of the video B frame. Both will return mean values over there.

Step 2 : The mask present in the video B frame moves upto 80% in horizontal directions, and compares with video A frame to identify the similarity.

Step 3 : If the difference is less than threshold value, the corresponding column values will includes in similarity array to find the overlap region

Step 4 : Once the horizontal directions is completed. The mask presents in the video A frame moves in vertical direction and repeat from step 2 to step 4. This process will be continuing for the whole image.

Step 5 : sort the similarity array in ascending order.

Step 6 : count column values present in similarity array.

Step 7 : the algorithm will returns maximum column value as the overlap region.

C. Video stitching and panoramic view generation

The overlap column value where given input to this stage, which got from previous stage. The homography matrix [9] is identified and used to merge the overlap region and to create the panoramic image. The input videos are then merged and a panoramic view of video is created from the stitched images.

IV. RESULTS AND ANALYSIS

Our algorithm is tested on three types of datasets. First dataset comprises of overlapping videos having identical lighting conditions. Second dataset consists of different lighting conditions. Third dataset consists of videos of various overlapping conditions. The proposed algorithm gives better results compared to the existing algorithms. The test videos for the proposed System have been taken using web camera. The two web cameras were used to captures input videos. All these video clips were digitized in AVI format at 30 frames per second. The resolution of video frames is 320x640 pixels encoded as 24 bits/pixel. The system has been implemented using JAVA 1.5 and tested on an Intel Pentium 4 CPU 3.00 GHz, with 1GB of RAM running Windows XP. The video class has been implemented using JAVA 1.5 using the API Java Media Framework 1.2.1.

Table 1. Comparative analysis of mean intensity Vs Histogram based similarity

Algorithm	Video A and Video B having Identical lighting conditions	Video B having more brightness compare with Video A			Video A having more contrast compare with Video B			Video A having more brightness compare with Video B			Video B having less contrast compare with Video A			Video A and B with various overlapping
Brightness and contrast variation	Nil	5 %	10 %	25 %	5 %	10 %	25 %	5 %	10 %	25 %	5 %	10 %	25 %	Nil
No of Input Videos	5	5	5	5	5	5	5	5	5	5	5	5	5	5
Similarity based on Mean intensity	5	5	5	4	5	5	4	5	5	4	5	4	3	5
Similarity based on Histogram	5	3	2	1	4	3	2	3	3	2	2	1	1	5

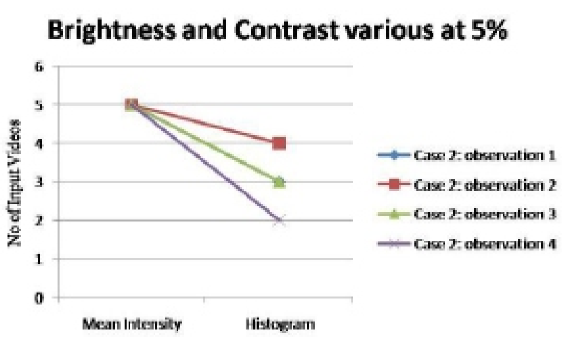

Fig. 5. Analysis chart 1 mean intensity Vs Histogram

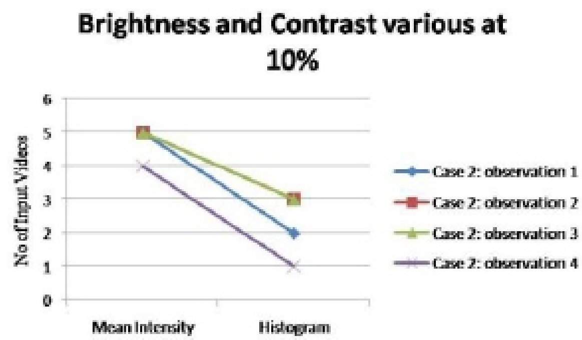

Fig. 6. Analysis chart 2 mean intensity Vs Histogram

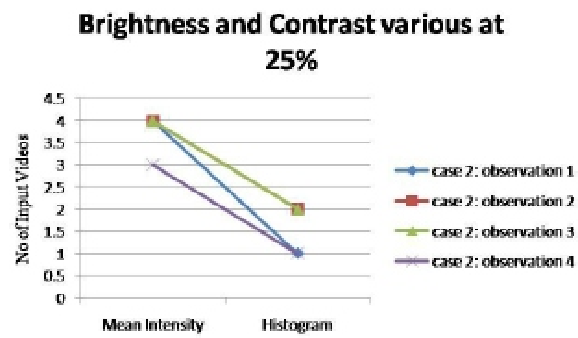

Fig. 7. Analysis chart 3 mean intensity Vs Histogram

A.. Identical lighting conditions

In this case, the sequences of video camera have been capturing the scenes under the same lighting conditions, and sequences of video taken at same time instance.

Fig. 8. (a) and (b) Sequences of overlapped input video capture the scenes at the same time instance

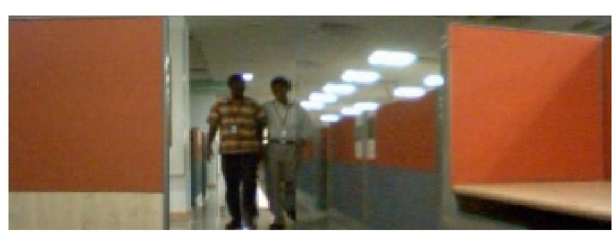

Fig. 9. (a) The panoramic view based on Mean Intensity similarity

Fig. 9. (b) The panoramic view based on Histogram similarity

B. Different lighting conditions

In this case, the sequences of video camera have been capturing the scenes under the divergent lighting conditions.

Observation 1. In first observation, the input videos were taken under different brightness condition. Compare with first input video the second input video added more brightness

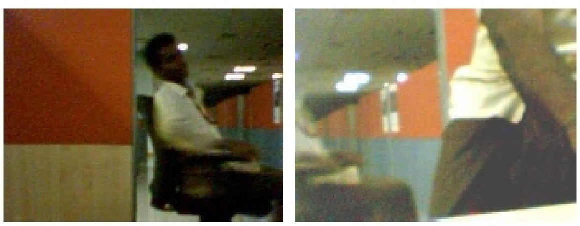

Fig. 10. (a) and (b) The sequences of overlapped input video were in different brightness conditions. The figure (b) having more brightness compare with figure (a).

Fig. 11. (a) The panoramic view based on mean intensity similarity

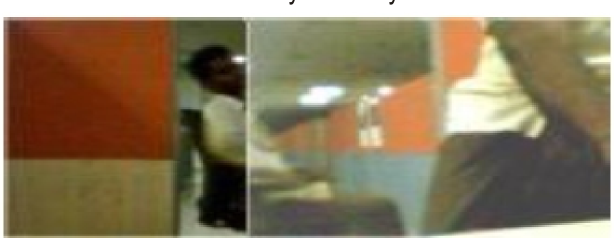

Fig. 11. (b) The panoramic view based on Histogram similarity

Observation 2. In second observation, the input videos were taken under different contrast condition. Compare with second input video the first input video having more contrast.

Fig. 12. (a) and (b) The sequences of overlapped input video were in different contrast conditions. The figure (a) having more contrast compare with figure (b).

Fig. 13. (a) The Panoramic View based on mean intensity similarity

Fig. 13. (b) The Panoramic View based on Histogram similarity

Observation 3. In third observation, the input videos were taken under different brightness condition. Compare with second input video the first input video added more brightness.

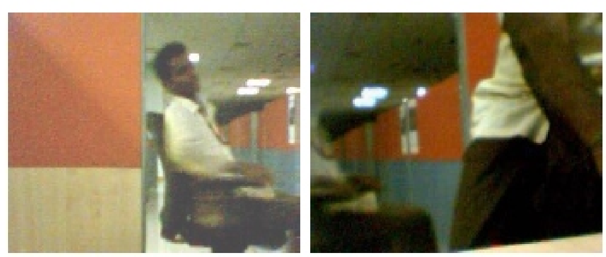

Fig. 14. (a) and (b) The Sequences of Overlapped Input Video Were In Different Brightness Conditions. The Figure (a) Having More Brightness Compare With Figure (b).

Fig. 15. (a) The panoramic view based on mean intensity similarity

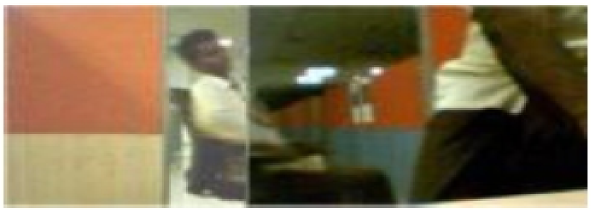

Fig. 15. (b) The panoramic view based on histogram similarity

Observation 4. In fourth observation, the input videos were taken under different contrast condition. Compare with first input video the second input video having less contrast.

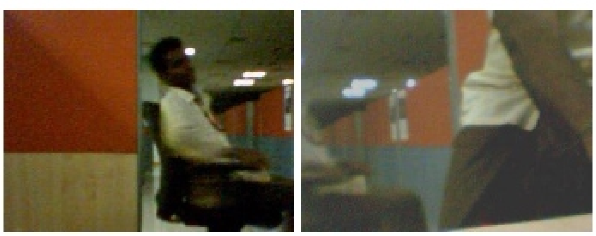

Fig. 16. (a) and (b) The sequences of overlapped input video were in different contrast conditions. The figure (b) having less contrast compare to figure (a).

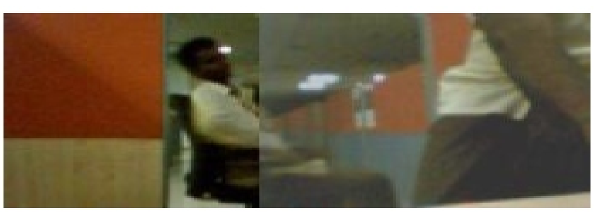

Fig. 17. (a) The panoramic view based on mean intensity similarity

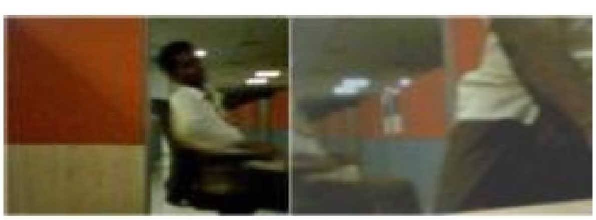

Fig. 17. (b) The panoramic view based on histogram similarity

C. Various overlapping conditions

In this case, the sequences of input video camera have been capturing the scenes under the different overlapping. Analysis performed on sequences of input video includes overlapping between 10 percentages to 80 percentages. In this type of input videos the algorithm performs 100% accuracy.

Fig. 18. (a) and (b) The sequences of input video having 14% overlap.

Fig. 19. (a) The panoramic view based on mean intensity similarity

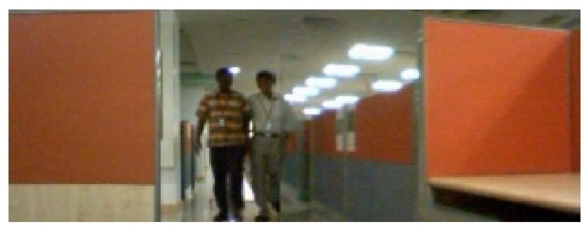

Fig. 19. (b) The panoramic view based on histogram similarity

Fig. 20. (a) and (b) The sequences of input video having 70% overlap

Fig. 21. (a) The Panoramic view based on mean intensity similarity

Fig. 21. (b) The Panoramic view based on histogram similarity

V. CONCLUSIONS

From the literature survey it has been understood that different approaches towards the same problem address different lightness conditions and time synchronization independently and also they are not intended for video. The proposed approach takes care of different lightness condition and time synchronization together. We can infer that this proposed technique gives satisfactory results. However under the conditions of rotation and transformation the algorithm fails to produce a good live panoramic view.

REFERENCES

[1] R. Szeliski and S. Kang, 1995, “Direct methods for visual scene reconstruction”, IEEE Workshop on Representations of Visual Scenes, pp.26– 33, Cambridge, MA.

[2] M. Irani and P. Anandan, 1999, “About direct methods” B. Triggs, A. Zisserman, and R. Szeliski, editors, Vision Algorithms: Theory and Practice, number 1883 in LNCS, pp.267–277, Springer-Verlag, Corfu, Greece.

[3] J. S. Jimmy Li and Sharmil Randhawa. “Improved Video Mosaic Construction by Selecting a Suitable Subset of Video Images” 27th Australasian Computer Science Conference, the University of Otago, Dunedin, New Zealand.

[4] Jonathan Foote and Don Kimber, 2001, “Enhancing Distance Learning with Panoramic Video”, 34th Hawaii International Conference on System Sciences.

[5] Lowe, D.G. “Object recognition from local scale-invariant features”, International Conference on Computer Vision, Corfu, Greece, pp. 1150-1157.

[6] M.A. Fischler and R.C. Bolles,1981, “Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography”,Communications of the ACM, 24(6):381-395.

[7] Richard Szeliski, 2004, “Image Alignment and Stitching: A Tutorial”, Technical Report Microsoft Research.

[8] David G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”,2004,Computer Science Department University of British Columbia Vancouve.

[9] Simon J.D. Prince, Ke Xu, and Adrian David Cheok, 2002, “Augmented Reality Camera Tracking with Homographies”,IEEE Computer Graphics and Applications, pp. 39–45.