Optical Flow for Detection of Transitions in Video, Face and Facial Expression

Jharna Majumdar[1(✉)], M. Aniketh[2], and N. R. Giridhar[2]

[1] Department of Computer Science and Engineering, Head Center for Robotics Research, Bangalore, India
jharna.majumdar@nmit.ac.in
[2] Department of Computer Science and Engineering, Nitte Meenakshi Institute of Technology, Bangalore 560 064, India

Abstract. Optical Flow is the apparent motion of pixels that is generated when there is relative motion between an observer and a scene. Optical flow is used in many areas of research for object detection, motion estimation, navigation and tracking applications. In this paper, we have proposed two novel applications where Optical Flow has been used for Determining Shot Transitions in a video sequence and Human-Face Expression Detection in a video. For Video Shot Detection, local invariant feature points using SIFT (Scale Invariant Feature Transform) corner detectors is calculated and Optical Flow is computed with the detected feature points to determine shot changes in the video. Quality Parameters like Recall, Precision and F-measure is used to determine the quality of the algorithm. Whereas for detecting of human faces, we begin by first performing skin segmentation to obtain probable regions where human face is present. Within the probable region Optical Flow is used to eliminate the background and other objects having human skin color. From the isolated face, Optical Flow is used for identifying expressions.

Keywords: Scale Invariant Feature Transform (SIFT)
Shot transition in a video · Optical flow · Skin segmentation
Expression detection

1 Introduction

With the increase in multimedia technologies, storage has become a major concern. When it comes to digital media, shot change detection is proven to be the basic step to achieve temporal segmentation. Hard cut (abrupt change) is easy to locate when to compare to dissolve, fade and wipe (gradual transitions) in a video.

To locate hard cut many methods have been proposed. For example, Nourani-Vatani et al. [2] exploits optical flow for scene change detection. It accounts for how flow vectors are generated and used to detect discontinuity and appearance change at key locations. Similarly, Majumdar, et al. [4] used local invariant feature points extracted from KLT and SIFT, developed by Lowe [5], corner detectors to locate scene breaks and for computing the corner point's optical flow was used.

K. Arai et al. (Eds.): SAI 2018, AISC 858, pp. 511–523, 2019.
https://doi.org/10.1007/978-3-030-01174-1_38

Optical flow has proven to have many applications in the field of image and video processing. It is mainly used in areas concerning tracking applications. Liu et al. [3] in their research work has focused on object tracking and has made use of Multi-scale Harris corner point features. To locate these points in a video sequence optical flow has been used. Computation of Optical Flow imposes constraints. So Mohammed et al. [1] proposed an optical flow computation method based on local features called nearest flow which is about estimating the distance ratio of two nearest features to find the best match for a feature point.

Kroger et al. [8] in their application of industrial food inspection has shown the performance of Hough Transform and Random Sample Consensus (RANSAC) algorithms in estimating the corner point position. SIFT show their limitations when there is a drastic change in images. To overcome this limitation, Beckouche et al. [9] proposed a new algorithm called Affine Parameters Estimation by Random Sampling (APERS) which detects the outliers in a given matched points.

Face detection is a computer technology used to identify human faces in digital images or videos. Face detection is a difficult task in image analysis which has each day more and more applications. Many algorithms have been proposed to detect human faces. The algorithm proposed by Viola and Jones [10] is a robust technique which makes use of Haar Cascades, Integral Image and Ada Boost Filtering to detect objects. This method is very popular and widely used. Dhavalikar et al. [11] used YCbCr color model for skin segmentation and Active Appearance Method (AAM) to extract facial features.

A facial expression is one or more motions or positions of the muscles beneath the skin of the face. Detecting such expressions has attracted increasing attention from researchers. Dhavalikar et al. [11] used Euclidian distance and Artificial Neuro-Fuzzy Interface System to detect expressions. Kulkarni et al. [12] have made used of filters like Gabor and log Gabor to extract facial features. Log Gabor filter increases the accuracy of expression recognition. Happy et al. [13] made use of appearance features of selected facial patches for expression recognition. These patches are further processed to obtain discriminate features for classifying each pair of expression.

Optical Flow has been mainly used for object tracking. In this paper we have made use of Optical Flow for video shot detection, using corner detectors (Sect. 2), face detection (Sect. 3) and facial expression detection (Sect. 4).

2 Optical Flow Using Corner Detectors for Video Shot Detection

Differential based and feature based are the two commonly used methods for optical flow computation. As mentioned earlier, we make use of optical flow for detecting shot changes in video. However, determining optical flow using differential technique is computationally intensive and imposes additional constraints in the derivation of optical flow equation. Further it is a challenge for optical flow to handle large displacements and deformations. We can overcome this challenge using feature matching which has the ability to capture large displacements. In the proposed work, feature based i.e. corner detector based optical flow is used for detecting shot transitions in videos. We have developed the SIFT algorithm and we make use Hough Transform for removing false matches. Optical flow is then used to determine shot changes. Finally quality parameters like Recall, Precision and F-measure are used to determine the quality of the algorithm.

2.1 Scale Invariant Feature Transform (SIFT)

As mentioned earlier, we have developed the SIFT algorithm using C++ [5, 6]. The feature points generated using SIFT algorithm are invariant to scaling, rotation, illumination and partially invariant to affine transformations. Figure 1(a) and (b) shows the input and corresponding output of the developed algorithm.

(a) (b)

Fig. 1. Input Image, (b) Output of SIFT algorithm (343 key-points).

2.2 Optical Flow Using Local Features

After extracting the local feature points, optical flow is computed using feature matching. The key-points in the two frames are matched by finding the nearest neighbor for each point using Euclidian distance measure. Instead of matching all the points it is necessary that we match only "similar" key-points. This is done by taking ratio of Euclidian distances of 1^{st} and 2^{nd} nearest neighbor [5]. If this ratio is above certain threshold, then matching is neglected. Figure 2 shows the results of key-point matching between two frames of a video.

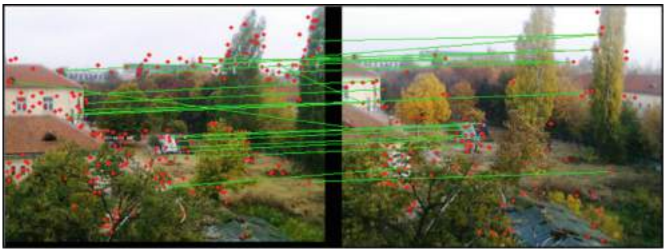

Fig. 2. Optical Flow between two frames using Key-point matching.

2.3 Outlier Detection Using Hough Transform

Feature based optical flow is computed by matching the key-points between the two successive frames. Hence it is essential to remove any outliers to increase the accuracy and robustness of the algorithm. The problem of matching efficiency of the SIFT algorithm is addressed by a generalized Hough transform. According to the work proposed in [7], Hough Transform produces more accurate and stable results for experimental data when compared to RANSAC which produces good results for synthetic data. Further, RANSAC tends to be inaccurate in the presence of Gaussian noise. In this work, we have therefore made use of Hough Transform for outlier detection.

In order to identify subsets of matching key points that agree on location, scale, and orientation, the algorithm detects straight lines using the slope-intercept y = mx + b. For each pixel at (x, y) and its neighborhood, the Hough transform algorithm determines if there is enough evidence of a straight line at that pixel. Points are incremented for every possible line (m, b) that runs through that point. Maximal values correspond to the most prominent feature points in the image. The output of outlier detection algorithm is shown below in Fig. 3.

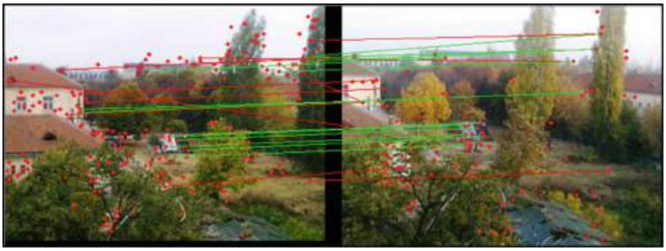

Fig. 3. Outlier Detection using Hough Transform – Red lines indicate false matches identified using Hough Transform.

2.4 Performance Measure

For determining the quality of the proposed algorithm, quality parameters like Recall, Precision and F-measure is used.

- Recall is the ratio of correct detections over all true detections.

$$\textbf{Recall rate} = \frac{No.\ of\ Changes\ Identified}{Total\ No.\ of\ Exsisting\ Changes}$$

- Precision is the ratio of correct detection over all detections.

$$\textbf{Precision rate} = \frac{No.\,of\ Correct\ Changes\ Identified}{\text{No. of Changes Identified}}$$

- F-measure combines precision and recall, is the harmonic mean of precision and recall.

$$F_1 = 2 * \frac{\text{Precision} * \text{recall}}{\text{Precision} + \text{recall}}$$

The results for the algorithm proposed above are shown in Part I of Results Section.

3 Face Detection

One of the most popular and widely used face detection algorithm is the Viola-Jones algorithm, which makes use of Haar Cascade filters [10]. Many methods have been proposed for detecting human face [11]. In the current work, we have proposed segmentation and optical flow based face detection algorithm. This algorithm has four key stages: (1) Skin Segmentation, (2) Noise Removal, (3) Clustering (4) and Elimination of False Objects.

3.1 Skin Segmentation

The objective of skin segmentation is to recognize skin pixels in the given image. The algorithm not only considers individual ranges of the three color parameters RGB, but also takes into account combinational ranges of each color parameter to provide greater accuracy in recognizing the skin area in a given image. Following function is used for skin segmentation:

$$\text{If } p_{i,j} \begin{cases} R > 50\ \&\ G > 30\ \&\ B > 20 & O_{i,j} = 255 \\ \&\ R > G\ \&\ R > B\ \&\ |G - R| > 15 & \\ \text{Otherwise} & O_{i,j} = 0 \end{cases}$$

$P_{i,j}$ = Input image (24 bit)
$O_{i,j}$ = Output image (8 bit)

$0 \leq i \leq$ height of an image, $0 \leq j \leq$ width of an image

The threshold image ($O_{i,j}$) contains pixels which belong to skin region. After segmentation, along with the human face other objects having the color of human skin is detected, is shown in Fig. 4.

3.2 Noise Removal

Noise in this case is a cluster of pixels which have a very small area when compared with that of the image. For every cluster of pixels which are connected to each other, a boundary is drawn from which size of the cluster is estimated. If the cluster size is very less, the entire cluster of pixels is erased and the same is shown in Fig. 5.

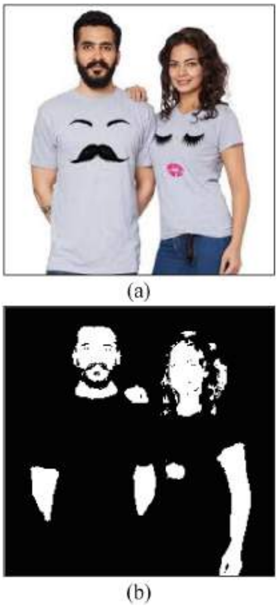

Fig. 4. (a) Input image, (b) Output obtained after applying the skin segmentation.

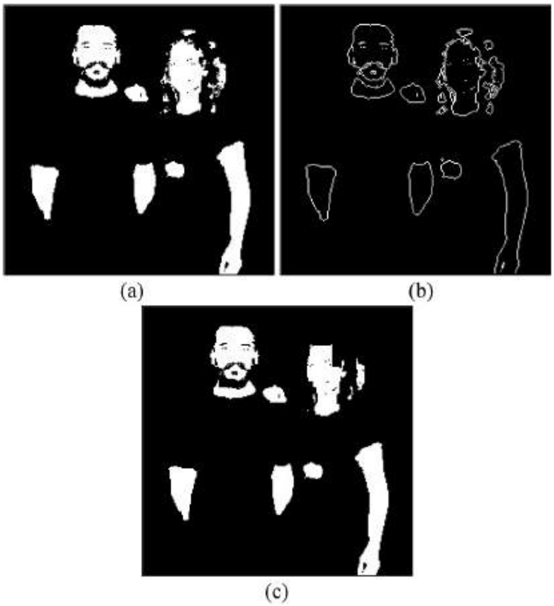

Fig. 5. (a) Segmented image, (b) Boundary of each group of pixels which are connected to each other, (c) After eliminating the group which has less area when compared to that of dimension of the image.

3.3 Clustering

Clusters which are close to one-another, having similar pixel range is grouped together to form a larger cluster. This step helps in eliminating the discontinuities between two related clusters. A bounding box is drawn for identifying the new clusters. The size of the largest cluster is determined. Clusters whose size is very small compared to size of the largest cluster is also erased. Figure 6 shows the bounding box being drawn around the cluster of pixels.

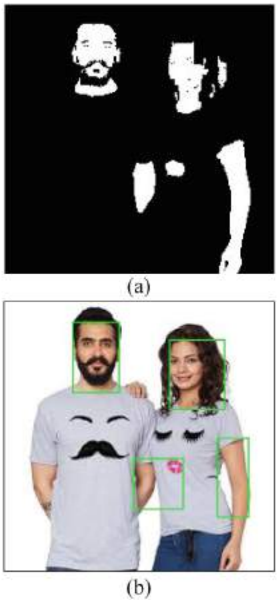

Fig. 6. (a) Threshold image after elimination of minor clusters, (b) Bounding box showing new clusters.

3.4 Elimination of False Objects

The aspect ratio of each bounding box is calculated by taking the ratio of box height to box width. Ideally, the aspect ratio of a human face lies between 1.1 and 1.8. Any bounding box whose aspect ratio exceeds this range is eliminated. There is still a possibility of a non-face object to be present along with the human face, if that object has color and aspect ratio of a human face. Optical Flow is applied to eliminate such objects i.e. background objects have relatively less motion between an observer and a scene, hence show minimal flow. The same is represented in Fig. 7.

The results for the algorithm proposed above are shown in Part II of Results Section.

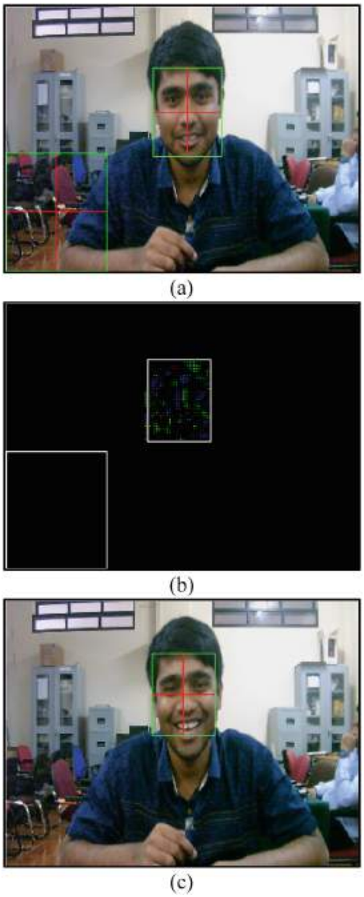

Fig. 7. (a) Frame with background object, (b) Optical Flow of the detected objects, (c) After removing objects with zero flow.

4 Identification of Facial Expression

To identify human expressions many novel methods have been proposed [11–13]. In the previous section, we have obtained the frame in which human face is isolated. We make use of the obtained face and divided into four quadrants so as to isolate facial muscles involved in making an expression. Optical flow within each sub region is calculated. For each pixel, optical flow corresponds to a vector having magnitude and orientation. Dominant orientation for each sub region is required to identify the facial expression. To do so, we make use of 36 bin histogram. The peak of the histogram gives the dominant orientation. Other bins whose magnitude is 80% of the peak is also considered. The dominent orientation obtained from each sub region is used for identifying an expression (Fig. 8).

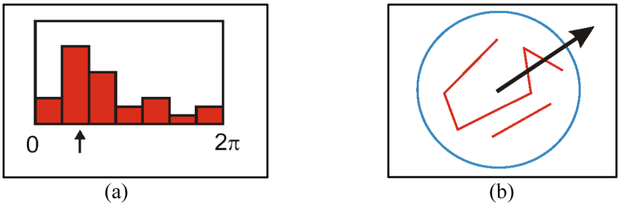

Fig. 8. (a) Histogram of local gradient directions to compute dominant orienation within each sub-region. (b) Dominant Orientaion of the sub-region.

The results for the algorithm proposed above are shown in Part III of Results Section.

5 Results

In this section, we have presented the results in the following three parts:

5.1 Part I – Optical Flow Using Corner Detector for Video Shot Detection

The algorithm discussed in Sect. 2 is implemented for three different video sequences. In Figs. 9, 10 and 11(a), (b), (c) shows the video transition and corresponding flow obtained during this transition is shown in (d), (e) and (f), respectively. During shot transition, the scene changes and the corresponding flow during this transition is zero. Quality parameters calculated for the three video sequences, are shown in Table 1. In Table 1, we see that the algorithm gives very good result for all the three quality parameters for all the three video sequences.

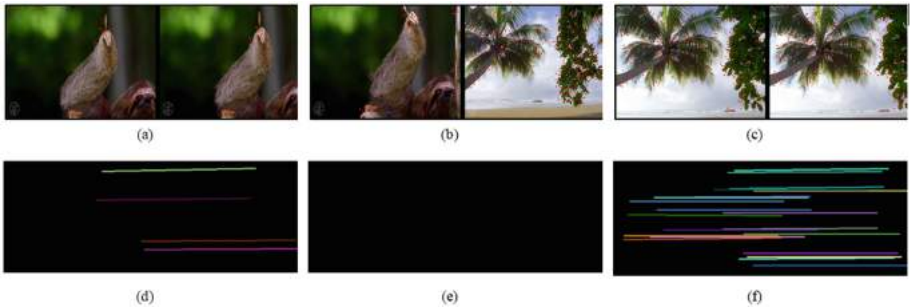

Fig. 9. Video Shot Detection Using Optical Flow – Clip 1 (a) Scene 1-Frame No. 17-18 (b) Shot-Frame No. 18-19 (c) Scene 2-Frame No. 19-20 (d) Optical Flow corresponding to Scene 1- Frame No. 17-18 (e) Optical Flow corresponding to Shot- Frame No. 18-19 (f) Optical Flow corresponding to Scene 2- Frame No. 19-20.

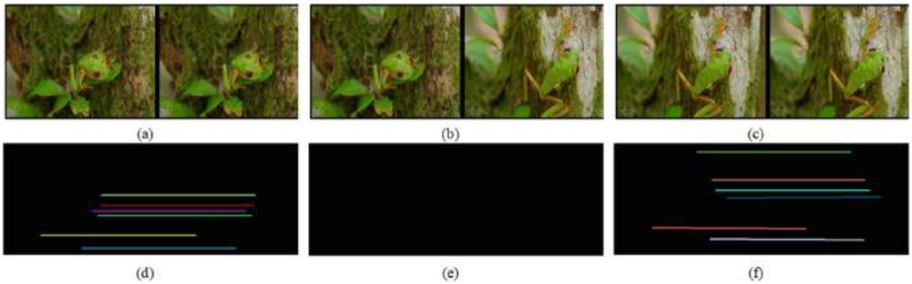

Fig. 10. Video Shot Detection Using Optical Flow – Clip 2 (a) Scene 1-Frame No. 57-58 (b) Shot-Frame No. 58-59 (c) Scene 2-Frame No. 59-60 (d) Optical Flow corresponding to Scene 1- Frame No. 57-58 (e) Optical Flow corresponding to Shot- Frame No. 58-59 (f) Optical Flow corresponding to Scene 2- Frame No. 59-60.

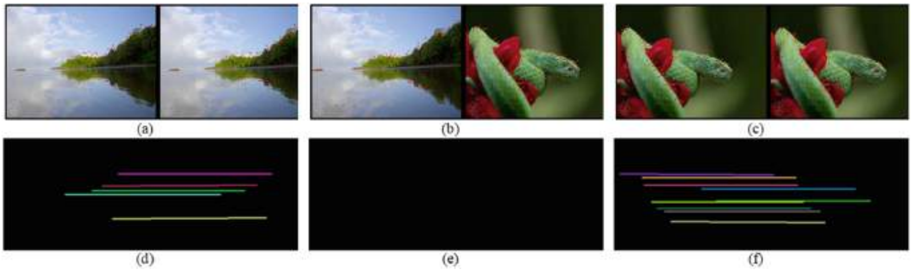

Fig. 11. Video Shot Detection Using Optical Flow – Clip 3 (a) Scene 1-Frame No. 87-88 (b) Shot-Frame No. 88-89 (c) Scene 2-Frame No. 89-90 (d) Optical Flow corresponding to Scene 1- Frame No. 87-88 (e) Optical Flow corresponding to Shot- Frame No. 88-89 (f) Optical Flow corresponding to Scene 2- Frame No. 89-90

Table 1. Quality parameter values of three different video sequences

Quality parameter vs input	Clip-1	Clip-2	Clip-3
Recall	98.23%	99.31%	98.42%
Precision	100%	100%	100%
F-Measure	99.11%	99.65%	99.20%

5.2 Part II – Face Detection Using Skin Segmentation and Optical Flow

The face detection algorithm described in Sect. 4 is run for two different video sequences. The results of face detection using skin segmentation and optical flow is shown in Figs. 12 and 13. In Fig. 12(a), (b), (c), (d) are input video frames of video sequence 1 and (e), (f), (g), (h) are the corresponding face detection outputs. Similar results are shown in Fig. 13 for video sequence 2.

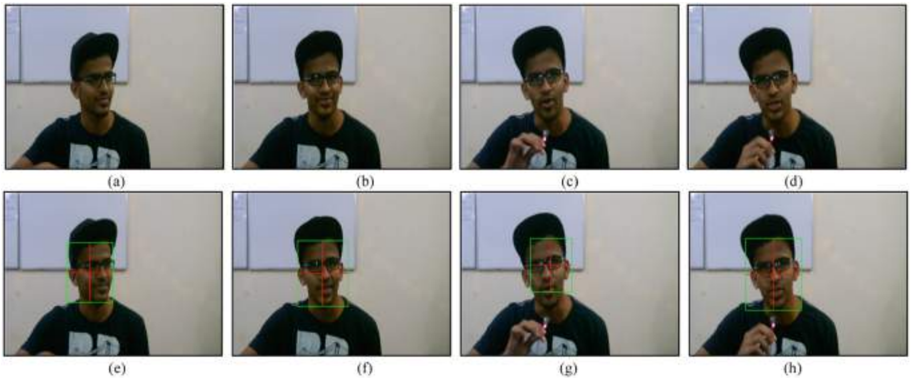

Fig. 12. Face Detection using Skin Segmentation and Optical Flow – Video Sequence 1 (a) Frame No. 23 (b) Frame No. 24 (c) Frame No. 25 (d) Frame No. 26 (e) Face detection corressponding to Frame No. 23 (f) Face detection corressponding to Frame No. 24 (g) Face detection corressponding to Frame No. 25 (h) Face detection corressponding to Frame No. 26.

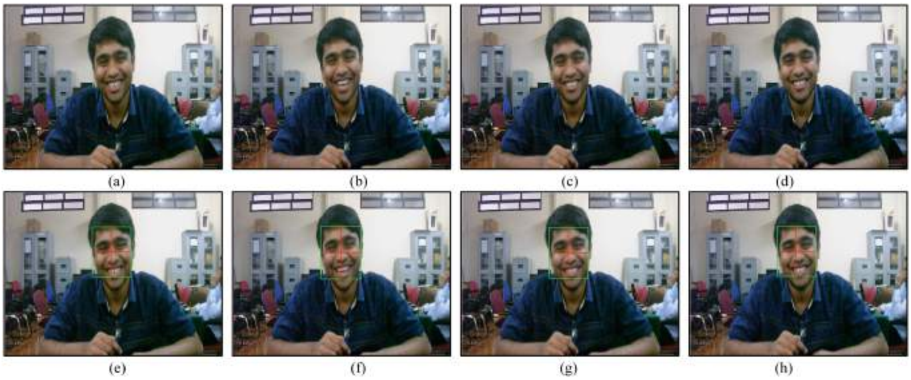

Fig. 13. Face Detection using Skin Segmentation and Optical Flow – Video Sequence 2 (a) Frame No. 63 (b) Frame No. 64 (c) Frame No. 65 (d) Frame No. 66 (e) Face detection corressponding to Frame No. 63 (f) Face detection corressponding to Frame No. 64 (g) Face detection corressponding to Frame No. 65 (h) Face detection corressponding to Frame No. 66.

5.3 Part III – Identification of Facial Expression Using Optical Flow

Facial expression identification algorithm described in Sect. 3 is run for two different video sequences each comprising images of faces having different facial expressions. Figure 14(a), (b), (e), (f) shows different facial expressions in video sequence 1. The corresponding flow between (a) - (b) and (e) - (f) is shown in (c) and (g) respectively. The dominant orientation seen in (c) and (g) is shown in (d) and (h). Similar results are show in Fig. 15 for video sequence 2.

To identify the facial expressions, the dominant orientations within each sub region are analyzed. For Smile expression, sub regions 3 and 4, shows the dominant

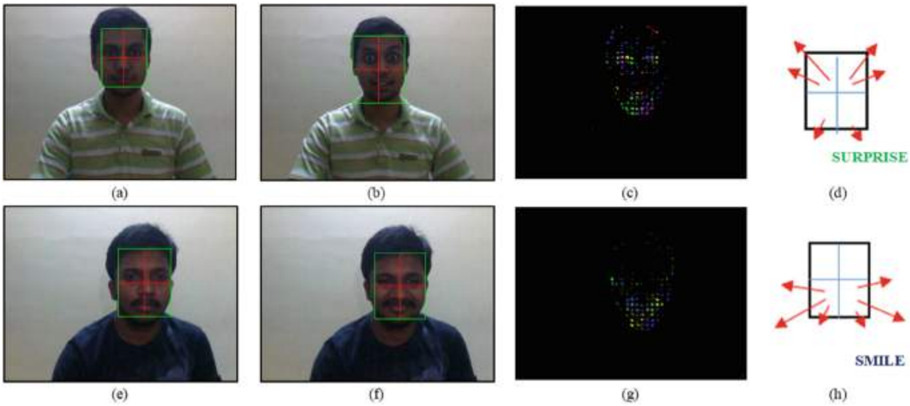

Fig. 14. Identification of Facial Expression using Optical Flow (a) Frame no. 31 (b) Frame No. 32 (c) Optical Flow Corresponding to Frame no. 31 and 32 (d) Corresponding Dominant Orientation of Sub regions (e) Frame No. 61 (f) Frame no. 62 (g) Optical Flow Corresponding to Frame no. 61 and 62 (5) Corresponding Dominant Orientation of Sub regions

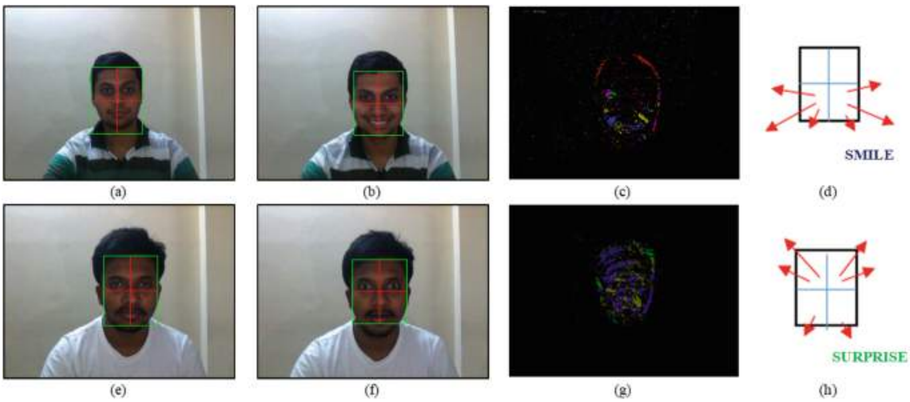

Fig. 15. Identification of Facial Expression using Optical Flow (a) Frame no. 81 (b) Frame No. 82 (c) Optical Flow Corresponding to Frame no. 81 and 82 (d) Corresponding Dominant Orientation of Sub regions (e) Frame No. 111 (f) Frame no. 112 (g) Optical Flow Corresponding to Frame no. 111 and 112 (5) Corresponding Dominant Orientation of Sub regions

orientation in the range of 180^0–270^0 and 270^0–360^0 respectively, as seen in Figs. 14 (h) and 15(d). For identification Surprise expression, all four sub-regions show dominant orientation. Regions 1, 2, 3, 4 shows dominant orientation in the range of 0^0–90^0, 90^0–180^0, 180^0–270^0, 270^0–360^0, respectively, as seen in Figs. 14(d) and 15(h).

6 Conclusion

In the present paper, we have proposed Optical Flow using SIFT Corner Detector. The code of SIFT is written in C++ using the method proposed by David Lowe [5]. The purpose of not using existing OpenCV code is to be able to introduce variations in the algorithm, which is otherwise not possible by making OpenCV function calls. We have used Optical Flow for three different applications: (1) to finds the CUT Transition in Video sequence for the purpose of Video Shot Detection; (2) Face Detection; (3) Determining Facial Expression. The future way forward are the following: (1) Integrating the Face and Facial Expression Detection to the Humanoid Robot developed in-house in our Robotics Research Centre. (2) Porting the entire Software Code in Embedded Board UDOO (with 4 Processors) and perform Hardware Optimization for real time Exploitation. (3) Extend this concept to other Facial Expressions like Sad, Disgust and Anger.

Acknowledgment. The authors express their sincere gratitude to Prof. N.R. Shetty, Advisor and Dr. H.C. Nagaraj, Principal, Nitte Meenakshi Institute of Technology for giving constant encouragement and support to carry out research at NMIT. The authors extend their thanks and gratitude to the Vision Group on Science and Technology (VGST), Government of Karnataka to acknowledge their research and providing financial support to setup the infrastructure required to carry out the research.

References

1. Mohammed, A.D., Morris, T.: Optical Flow Estimation using Local Features
2. Nourani-Vatani, N., Borges, P.V.K., Roberts, J.M., Srinivasan, M.V.: On the use of Optical Flow for Scene Change Detection and Description
3. Liu, M., Wu, C., Zhang, Y.: Multi-resolution Optical Flow tracking algorithm based on Multi-scale Harris Corner points feature
4. Majumdar, J., Venkatesh, G.M., Kamath, A.: Video Shot Detection using Corner Detectors and Optical Low
5. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints
6. Copyright 10718/2017-CO/ SW, SIFT corner detector and descriptor
7. Jacobs, L., Weiss, J., Dolan, D.: Object Tracking in Noisy Radar Data: Comparison of Hough Transform and RANSAC
8. Kroger, M., Sauer-Greff, W., Urbansky, R., Lorang, M., Seigrist, M.: Performance evaluation on contour extraction using Hough Transform and RANSAC for multi-sensor data fusion applications in industrial food inspection
9. Beckouche, S., Leprince, S., Ayoub, S.F.: Robust Outliers Detection in Image Point Matching
10. Viola, P., Robust, M.J.: Real-Time Object Detection
11. Dhavalikar, A.S., Kulkarni, R.K.: Face Detection and Facial Expression Recognition System
12. Ketki, R., Sahebrao, K., Bagal, B.: Facial Expression Recognition
13. Happy, S.L., Routray, A.: Automatic Facial Expression Recognition using Features of Salient Facial Patches. IEEE