

Creation and Implementation of Adaptive Algorithms For Visualization of Stereoscopic Face Mask

Maneet Kaur

Assistant Professor, School of Computer Science and Engineering,

Lovely Professional University, Phagwara, Punjab, India. Pin Code:144411

maneet.rainu@yahoo.com

Mangena Venu Madhavan

Research Scholar, School of Computer Science and Engineering,

Lovely Professional University, Phagwara, Punjab, India. Pin Code: 144411

madhavan.11601828@lpu.in

ABSTRACT: *Computer vision and Image processing are one of the most important technical aspects in the field of machine learning. In this paper, we are mainly discussing the face masking of selfie image using computer vision and image processing. In this, we also discuss adaptive algorithms we implemented for multiple image input (MIIO3D) and single image input (SIIO3D) implementation for the construction of an efficient 3D face mask. We will likewise talk about the constraints and future viewpoints which can be attainable to the present technical world.*

Keywords: *Computer Vision, Image Processing, Feature Extraction, Feature Detection, Feature Matching, point cloud, Sfm (Structure from Motion), Triangulation method, Fundamental Matrix, camera parameters, SAD (Sum of Absolute Differences), MATLAB, Face Mask, MIIO3D, SIIO3D.*

I. INTRODUCTION

Given a single image of an everyday object, a sculpture can recreate its 3D shape, even if the object has never been seen before. Presumably, it is familiarity with shapes of similar 3D objects and how they appear in images, which enables the artist to estimate its shape. This might not be the exact shape of the object, but it is often a good enough estimate for many purposes. As the technology improved, everyone is more interested in taking selfies, images of architectural buildings and monuments and these images in a 2dimensional format. Even if we take the

images, we are missing certain information on the originality due to missing depth and height within the object in 2D images. If we can make a 3D model it can be useful in many aspects such as reconstruction of architectural buildings, protecting the old architecture in the form of 3D

of 3D images, 3D modeling of biological organs, and entertainment. These examples motivated us to form a novel framework for motion Selfie.

Why only Selfie?

We can go for some architectural designs, or to the level of 3d modeling biological organs. But in those cases, we need a heavy setup which will be more expensive also. We are making this project in a sense of prototype for all the other categories coming under this from an economic perspective. On the top of it, selfie will be very easily reachable to the audience so that we can grab their attention in order to explain the importance of 3D reconstruction from multiple 2D images.

A. Computer Vision

Computer Vision takes images and videos as an input to produce the desired result in real-time. Image processing is a subset of Computer vision, which gives the qualitative and quantitative information of the input. Computer Vision systems find the patterns in images and can spot the different types of objects. It produces the output as an image and the information about the shape, number, size, and color of the image. Computer vision systems can generate 3D models of the images like road and can predict

crashes. In the case of a self-driving car, it can protect passengers by deploying the airbags.

B. Image Processing

Image Processing is for the usage and application of mathematical functions and transformations over the images. Dense reconstruction involves a point cloud construction in order to obtain a 3-d image.

and columns of pixels, on which algorithms were implemented on the image like smoothing, contrasting, and sharpening, stretching, etc. Some algorithms adjust the exposure, contrast, shadows, highlights and many other properties of an image. For 1-dimensional signals, input images will be filtered with different LPF (Low Pass Filters), HPF (High Pass Filters), etc. HPF filter helps in finding the edges of an image and LPF helps in blurring or removing the noise of an image.

II. REVIEW OF LITERATURE

When we are converting 2D to 3D edge information is one of the crucial information[1]. Estimating the depth of an image which is a challenging task to convert the 2d image to 3d. The process of estimating depth is divided into various steps. This first step is block-based region grouping. In this process, pixels are grouped into blocks and pixels in these blocks are assumed to have some depth. Depth estimation requires the application of minimum spanning tree algorithms. The second step is to find depth from prior information: hypothesized depth gradient of each block is derived from the linear perspective information. The higher is the depth gradient the closer is the block is closer to the observer. The third step is Bilateral filtering: This step smoothing of the image takes place. In this step, the edges are detected and the blocky artifacts of the edges because of the previous step is removed. The last step is depth-based rendering: This step is further subdivided into other steps: pre-processing of depth, 3D image wrapping, Hole filling.

In 3D conversion, we have a lot of applications, particularly in security aspects. Suppose an army intended to know about a new place with enemies moving around then they can get the images by using Unmanned Aerial Vehicles (UAV) and can convert them into 3D form in order to get the proper information [2]. The first step is data acquisition which is done using UAVs installed with pre-calibrated camera. The next step is 3-dimensional building reconstruction; includes steps like feature description,

feature matching, loop closure, path estimation, dense reconstruction, surface fitting. Feature description and matching step involves locating key points of a figure and matching those key points with other figures. Loop closure is used to similar images in this method variance of two images are used to find the similarity between the two images. Dense reconstruction involves a point cloud construction in order to obtain a 3-d image.

Proposed methodology to convert a 2d image to 3d image. The basic idea to develop a 3d image is to construct the left and right view of an object using a stereo pair of images. Various mathematical concepts of interpolation and vanishing points are used in this method. The limitation of this algorithm was that it does not give accurate results for the round-shaped images [3].

While converting a 2D image to 3D, computer vision plays an important role [4]. It involves multiple 2D images and then converting them into 3D images and different steps involved in them.

Nowadays, smartphones become very common essential good for everyone. That is the only reason a lot companies are concentrating on building apps rather than building websites [5]. Methodology to construct a 3d image from multiple 2d images using a smartphone and the methodology to do that. This will provide us a motivation to bring 3D reconstruction at smartphone level besides having many constraints.

Structure from Motion (SfM) plays a vital role in building of 3D image from multiple 2d images. For doing this we should be having more clarity in terms of algorithm and its implementation. The major applications mentioned in this paper are: image-based 3D modeling, hand-eye calibration, augmented reality, autonomous navigation guidance, and motion capture [6].

After SfM, Sum of Absolute Differences (SAD) plays an important role in the building of a 3D image. It is very useful in analyzing the depth of the 2D image. In this scenario, we usually use stereo images due to they will be having more information which is essential for 3D image building than the monocular image. To this, the methodology for estimating the depth from 2D image is very crucial [7].

An optimum way to estimate the depth using pre-calculated features in the database. The steps include calculating the depth from a single image by using different light intensities of the image, considering a group of connected pixels as a window of fixed size and using its minimum intensity. First, the depth is taken as guess depth

and later iteratively the refined depth is calculated [8]. To preserve the global structure of the object the pattern is preserved using the comparison not only based on intensities but also on depths. This algorithm can be similarly used to construct back of an object.

Again, in the security perspective surveillance cameras play a crucial role. By using the images from a surveillance camera we can recreate the scene in 3D so that

we will be having more clarity at the object level [9]. The same case has been discussed in terms of the multiple view calibration and point cloud creation. The camera posture estimation focuses to find the relative camera position. This involves the setup of three cameras which are 60 degrees apart. This helps in posture estimation of the camera. This involves the calculation of dense point cloud and a mesh of the image. After the image extraction, the PMVS (Patch Multi-view stereo) is done with the SIFT algorithm for reconstruction from the image feature stored.

III. PROPOSED APPROACH

A. SfM

The procedure of assessing the three-dimensional structure of an object in a scenario in the form of two-dimensional views or images is known as Structure from Motion (SfM). There are different factors that affect the way we approach the problem like the ordering of images, camera used type and number of them are used. In case a single calibrated camera, we will be using *up to scale* method which refers to the rescaling of the composition and the value of the camera movement and preserve the observations obtained. It is actually intended to identify whether camera close to an object or not [6,10]. We are actually interested in calculating the original size of the object in global units, and then we need the extra knowledge about

- The object size which is used in the worked scenario
- Having knowledge from different sensors such as an odometer

In the SfM algorithm, our assumption is that the camera at the origin and its optical axis lies on the z-axis. The major steps in SfM can be given as follows:

1. SfM needs point correspondences between pictures. To do this, we will be using Kanade-Lucas-Tomasi (KLT) algorithm. It works well once the cameras are close. As cameras move additionally apart, the KLT algorithm breaks down, and feature matching is often used instead.

To find out the pose of the second camera relative to the primary camera, you need to calculate the fundamental matrix. Use the corresponding points found in the previous step for the computation. The fundamental matrix describes the epipolar geometry of the two cameras. It relates a point in one camera to an epipolar line in the other camera.

3. For calculation of relative camera positions, we need to give the input as a fundamental matrix. This will return the alignment and also gives the second camera position in the coordinate system of the first camera.
4. Find the matched points of 3D positions using the triangulation method. As the pose is up to the scale, when we calculate the structure, it will have the exact shape but not the original size.
5. After getting the matched points, we can reconstruct the image in 3D format.

SfM Working Model

If we generalize the scenario of two views to multiple views the process of SfM (Structure from Motion) will follow the different steps as mentioned in the following flowchart.

SfM (Structure From Motion) includes different steps such as identifying and matching features of the images, fundamental matrix estimation, select the pair of images which are not much distorted to reconstruct, use those images and apply triangulation method to triangulate the points, and finally adjust the bundle of points. In the Fig. 1 working model of SfM, describes the flow chart for SfM (Structure from Motion).

In the primary step of identification and matching features, the system is able to extract, identify the features in every image and matches the features depending on the 2D coordinate system points.

The fundamental matrix represents the relation between any pair of images at the pixel level of the same scenario. These matrix constraints the estimation of points in the pixel level from the scenario can occur in both images. Given the estimation of a scenario point into one of the images that equivalent points in the other image is controlled to a line which will help in the search, and also allow us for the detection of wrong equivalent image points which the fundamental matrix represents in referred to as epipolar constraint, matching constraint, discrete matching constraint, or incidence relation.

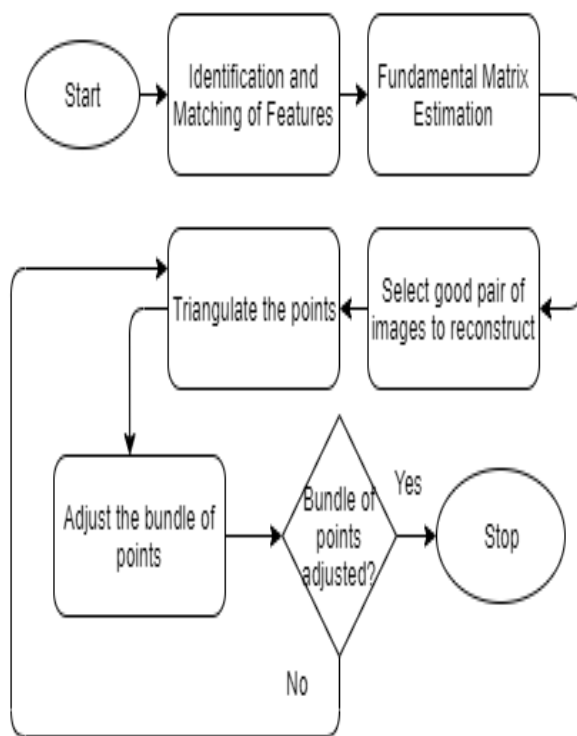


Fig. 1. SfM working model

Depending on the distortion of images, it will segregate the images into two divisions such as one division with less distortion or no distortion and the other having high distortion. Then, select the pair of images from the first section in order to use these images to build the 3D model.

Triangulate the points by using the triangulation method. Triangulation method is one of the crucial methods in which explains the way of verifying a point in three-dimensional space given its estimates onto a pair, or more, images. The solution for this problem, it is essential to know the variables of the camera projection function from three-dimensional to two-dimensional for the cameras engaged. Triangulation sometimes referred to as reconstruction. It generates the world coordinates for the existing images which will help to reconstruct the images.

Once the 2D image points been triangulated then we will try to adjust the bundle of points and again it goes to triangulating the points and the readjusts the bundle of points and this process will continue till there is not much difference in adjusting bundle of points.

B. SUM OF ABSOLUTE DIFFERENCE (SAD)

A sum of absolute difference (SAD) is a simple video quality metric used for block comparison and for

moving vectors calculations. Each frame is divided into small blocks (i.e. 8×8 pixels) and for every block, in one frame the most similar (minimum SAD) block in the next frame is found. This minimum sum of absolute differences is assigned as the SAD for each block in each frame (up to then-1 frame). Then all the SAD values are averaged for each frame and for all the frames in the clip, and divided by the block area, for normalization. Depth Estimation Analysis is done by using a Sum of Absolute Difference (SAD)[7,11].

C. Point Facial Landmark Algorithm

This algorithm is used for feature extraction. This will help to identify the landmarks on the face and numbered from 1 to 68[12]. If we are using this algorithm, then we need to identify in such a way that all the 68 points need to be identified. Otherwise, it might get landing with an error. For the elimination of this error, we will be considering the high definition images instead of low-quality images. In Fig.2, it is a simple representation of 68 – Point Facial Landmark Algorithm.

From the figure on the next page, we can see the 68 points on the face, which determines the face structure. The numbering of points varies accordingly as the face perspective of the input image. By using these facial landmarks, we can apply various Computer vision techniques, such as:

- Face Alignment
- Face part extraction
- Head pose estimation
- Blink detection
- Face swapping and etc.

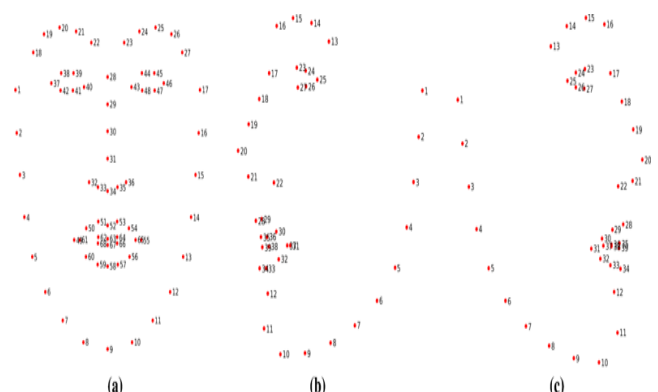


Fig. 2. (a) straight Face, (b) left side of the face, (c) right side of the face

D. KLT (Kanade - Lucas – Tomasi) Algorithm

KLT algorithm, majorly, we used for single input image and output 3D. In computer vision, for extracting the facial features, we use an approach of Kanade-Lucas-Tomasi (KLT) algorithm. It detects the face using feature points and tracks the face in real-time or from video. In real-time, it collects the facial features only for the first time and then starts tracking the face whenever a person moves [13]. For example, even if a person tilts the face, it will show the face features accordingly without calculating them, has at the first time it collected the features .

E. SIFT (Scale – Invariant Feature Transform) Algorithm

SIFT algorithm we used for the single image input and output 3D model. In general, this algorithm majorly can divide into four categorical steps. They are

1. Feature point or Keypoint detection
2. Feature point localization
3. Orientation assignment
4. Feature description generation

In the first step of Feature point or keypoint detection is to identify the regions of an image which are having unique features or information like blobs and corners[14].

In the second step of Feature point localization is a building block of computer vision. This forms a compact vector representation of the local

neighborhood. Using local features enables to better handle scale changes, rotation, and occlusion. The benefits of local features can be explained primarily in the following two ways.

1. For image stitching or three-dimensional construction, we need to focus on anchor points.
2. Without the necessity of image segmentation, we can represent the image information for the purpose of detection or classification.

Orientation assignment will be helpful in making a single image to get the dimensions get orient for a certain angle as we mention during the implementation and scale-invariant also gets applied in parallel to the orientation.

Feature description generator will generate descriptors that identify the linking among the pixels existing in a digital image. Majorly, there are two types of descriptors as follows:

1. *General information descriptors:* These descriptors are considered to be low-level descriptors. These will give the information about the color, shape, regions, textures, and motion.

2. *Specific domain information descriptors:* These provide the details of the events and objects of the scenario. A strong example of this is face recognition.

F. RANSAC Algorithm

RANSAC refers to Random Sample Consensus. While working on data with outliers the end result generated is ambiguous and doubtful. RANSAC [15] is a mathematical method which involves deriving a mathematical model from a dataset that involves an outlier. MATLAB computer vision toolbox provides a function for the RANSAC algorithm. This algorithm calculates the final result from the data which is not containing the outliers. The steps of this algorithm are:

1. The subsets of the larger dataset is randomly selected.
2. A model is made and fitted to the subset that is selected.
3. Identifying and removing a number of outliers.

G. KNN algorithm

Knn [16] algorithm known to majority as the classification technique under supervised learning. It can identify and segregate the points depending on the neighborhood. In the case of image processing, by using the basic flavor of KNN algorithm to utilize for clustering of points depending on the features and for the identification and removal of the outliers.

H. Harris corner detection algorithm

Harris corner detection algorithm is used for the corners detection and it is a very common algorithm in computer vision. It distinguishes very effectively between corners and edges. This algorithm mainly works to extract the corners and gives the information about the feature in the image [17]. When we use this algorithm, there is no need of using shifting patches for some angle, yet it proved efficient one. This algorithm majorly includes five steps as follows.

1. Changing the color image to grayscale.
2. Calculation of spatial derivative
3. Construction of structure tensor setup.
4. Compute eigenvalue of the structure tensor in order to calculate harris response.
5. Identify the optimal local points in order to identify the corners.

IV. MODEL EVALUATION

Adaptive Algorithms

We will be discussing the algorithms which are followed in two different scenarios separately. The first one is the multiple-image input and its 3D model output and we named it to be MIIO3D (Multiple Input Images Output 3D). The second is the single image input and its 3D model output and we named it to be SIIO3D (Single Input Image Output 3D). The algorithms are as follows

A. MIIO3D Algorithm for reconstruction of 3D image from multiple images

This particular we named it to be MIIO3D which abbreviates to be Multiple Input Images Output 3D. The name indicates the process of scenario follows.

Input: A series of images or views of an object and these often referred to as right and left views of an object.

Output: Three-dimensional reconstructed image of the two-dimensional view of the object.

1. Two-dimensional images of the object should be transformed into a greyscale.
2. Important camera variables are saved into the program.
3. The disturbance in the image should be removed and the undisturbed image is to be acquired.
4. The features should be extracted by using different views of an object for a single view.
5. Till all the images or views get finishes repeat step 4.
6. Matching points should be recognized.
7. Calculate the camera position by considering the previous image or view.
8. The camera position should be related to the current image or view to convert into a three-dimensional coordinate system.
9. Triangulation method to be for the initial three-dimensional global point.
10. Obtained three-dimensional global points and camera positions should be refined.
11. Obtained three-dimensional points should be displayed and location to be specified.
12. Calculate and display the dense image of three – dimensional model.

B. SIIO3D Algorithm for reconstruction of 3D image from a single image

This particular we named it to be SIIO3D which abbreviates to be Single Input Images Output 3D. The name indicates the process of scenario follows.

Input: A single image of the object in any direction

Output: Reconstructed three-dimensional image of the input two-dimensional image of the object.

1. The input 2D image is converted into a greyscale.
2. The camera parameters are loaded.
3. The features should be extracted by using different views of an object for a single view.
4. Matching points should be recognized.
5. Calculate the camera position by considering the previous image or view.
6. Repeat steps 3, 4, and 5 until all the angles are completed according to the user.
7. The camera position should be related to the current image or view to convert into a three-dimensional coordinate system.
8. Triangulation method to be for the initial three-dimensional global point.
9. Obtained three-dimensional global points and camera positions should be refined.
10. Obtained three-dimensional points should be displayed and location to be specified.
11. Calculate and display the dense image of three – dimensional model.

Flowchart for reconstruction of 3D image from multiple images

In this scenario, the work will be starting with an object which is the face of a person and then the multiple images of a face. In Fig. 3, it is the working model which is a representation of the algorithm MIIO3D.

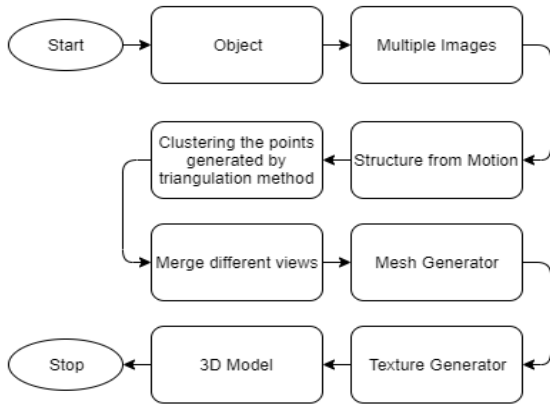


Fig. 3. Working model for MIIO3D

Flowchart for reconstruction of 3D image from multiple images

In this scenario, the work will be starting with an object which is the face of a person and then the single image of the face. In Fig. 4, it is the working model which is a representation of the algorithm SIIO3D.

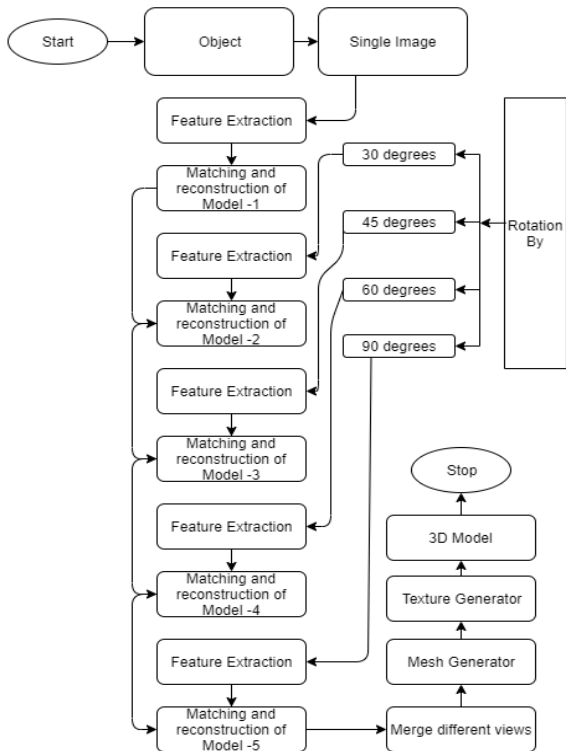


Fig. 4. Flowchart for SIIO3D

Output for reconstruction of 3D image from multiple images

INPUT:



Fig. 5. Sample Input Images

OUTPUT:

3D POINT CLOUD:

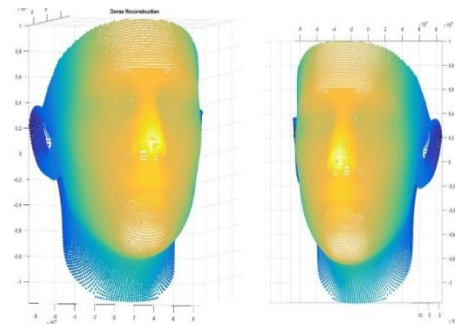


Fig. 6. 3D Point Cloud

FACE MESH:



Fig. 7. Face Mask

STEREOSCOPIC FACE MASK:



Fig. 8. 3D Face Model

Output for reconstruction of 3D image from a single image

INPUT:



Fig. 9. Sample Input Images

OUTPUT:

3D POINT CLOUD:

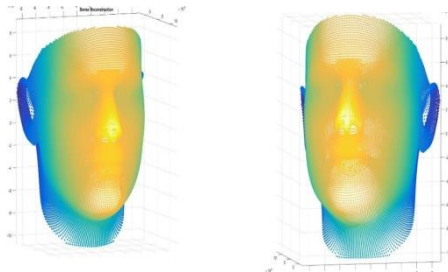


Fig. 10. 3D Point Cloud

FACE MESH:



Fig. 11. Face Mask

STEREOSCOPIC FACE MASK:



Fig. 12. 3D Face Mask

V. RESULT AND DISCUSSION

Usually, the performance of a program or a system measured on the basis of CPU time, memory used (primary memory) by the program and the database utilization. In this, mainly we concentrated on CPU time and memory usage. The main parameters we considered are the type of processor and file size. Processors we used are i3 (8th generation), i5 (6th generation), i7 (6th generation).

A. Performance of Program based on CPU Clock Time

From the Fig. 13, we can identify that multiple image input and single image input i3 (8th generation) working better when compared to i5 and i7. If we compare between i5 and i7, i7 performing better in multiple-input scenario whereas in single input those two processors are performing almost equally. Overall, multiple-input scenario performing much better than the single input scenario.

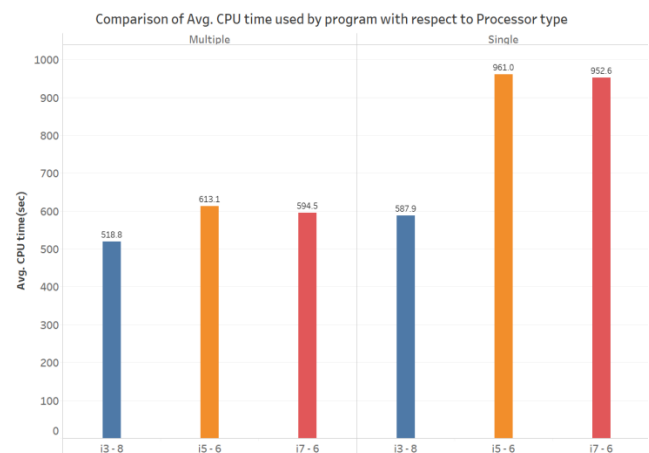


Fig. 13. For single image input vs multiple image input

B. Performance of Program based on Average Memory Used

From the Fig. 14, we can observe, in a multiple-input scenario, i3 and i7 utilization of RAM almost the same and the efficiency also better whereas in a single input scenario, i3 using highest RAM usage but i7 still showing the better performance. In multiple image input and single image input i3 (8th generation) working better when compared to i5 and i7. If we compare both the scenarios, we are unable to find a single pattern.

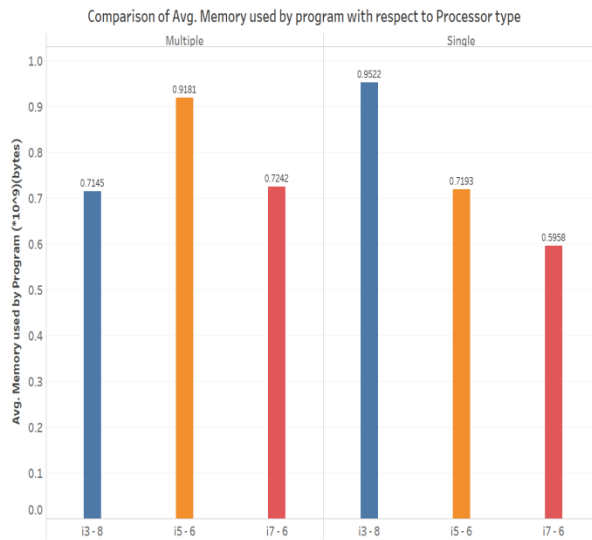


Fig. 14. For single image input vs multiple image input

C. Performance of Program based on Average CPU Time with File Size

From the Fig.15 we can notice, in a multiple-input image scenario, as the size of images increasing the average CPU time taken is getting decrease whereas in single input image as the size of the image is increasing, the average CPU time is getting increased.

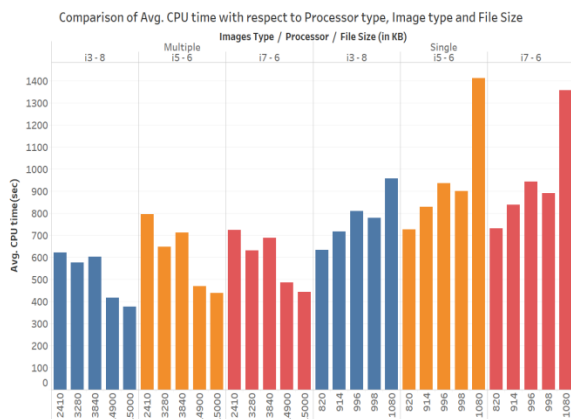


Fig. 15. Comparison of avg. CPU time with file size

D. Performance of Program based on Average Memory Utilized with File Size

From the Fig.16, we can identify that we are unable to identify any pattern that might be due to the PCs used are multiprogram PC. So, parallel processes might be running in the background that might be affected in the data taken. Overall, still i3 and i7 comparably working efficiently.

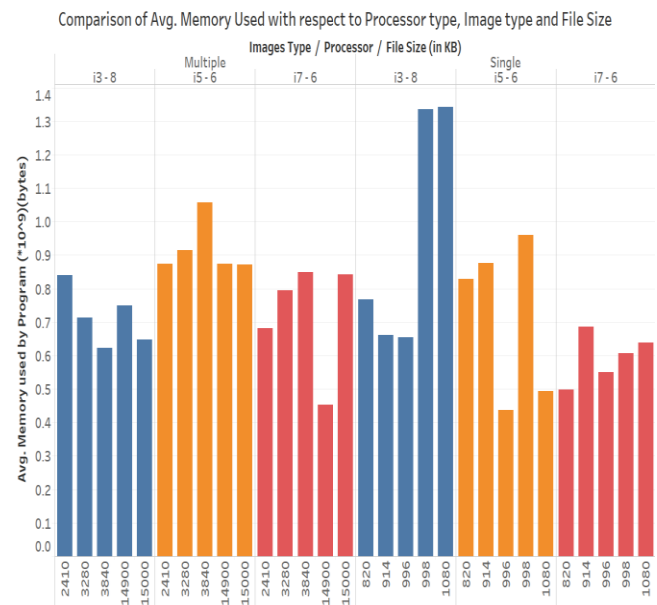


Fig. 16. Comparison of avg. memory utilized with file size

VI.CONCLUSION AND FUTURESOCPE

On a conclusive note, we able to frame the algorithms for multiple input images and single input images such as MII03D and SII03D respectively. We successfully implemented these algorithms and we able to generate the stereoscopic face mask. On top of it, we given the performance of our program on the basis of CPU time and memory used (primary memory). We further looking to make a better performance algorithm so that it can play an important role in this technical world. The applications of this scenario can be found in plastic surgery of the face, for investigation purposes like crime scene recreation, in education and especially in augmented reality.

REFERENCES

- [1] S. Bharathi, "2D-To-3D Conversion of Images using Edge Information," no. Icon3c, pp. 27–32, 2012.
- [2] C. Wefelscheid, R. Hänsch, and O. Hellwich, "Three-Dimensional Building Reconstruction Using Images Obtained By Unmanned Aerial Vehicles," *ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. XXXVIII-1/, no. September, pp. 183–188, 2012.
- [3] D. Donatsch, N. Färber, and M. Zwicker, "3D conversion using vanishing points and image warping," *3DTV-Conference*, pp. 0–3, 2013.

- [4] R. M. Swarna Priya, S. L. Aarthy, C. Gunavathi, P. Venkatesh, S. Koppu, and X. Z. Gao, "3Dreconstruction of ascene from multiple 2D images," *Int. J. Civ. Eng. Technol.*, vol. 8,no. 12, pp. 324–331, 2017.
- [5] O. Muratov, Y. Slynko, V. Chernov, M. Lyubimtseva, A. Shamsuarov, and V. Bucha, "3DCapture: 3D Reconstruction for a Smartphone," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, pp. 893–900, 2016.
- [6] A. Shalaby, M. Elmogy, and A. Abo El-Fetouh, "Algorithms and Applications of Structure from Motion (SFM): A Survey," *Int. J. Comput. Inf. Technol.*, vol. 06, no. 06, p. 358, 2017.
- [7] C. S. Panchal and A. B. Upadhyay, "Depth Estimation Analysis Using Sum of Absolute Difference Algorithm," *Int. J. Adv. Res. Electr.*, vol. 3, no. 1, pp. 6761–6767, 2014.
- [8] T. Hassner and R. Basri, "Example-based 3D reconstruction from single 2D images," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2006, no. July 2006, 2006.
- [9] N. Vincent *et al.*, "RECONSTRUCTION OF 3D MODEL FROM 2D SURVEILLANCE," vol. 3, no. 5, pp. 735–744, 2015.
- [10] S. Ullman, "The Interpretation of Structure from Motion Author (s): S. Ullman Source: Proceedings of the Royal Society of London. Series B, Biological Sciences, Vol. 203, 1979.
- [11] J. Vanne, E. Aho, and K. Kuusilinna, "for Motion Estimation," vol. 16, no. 7, pp. 876–883, 2006.
- [12] W. Abdalimageedet *al.*, "Face recognition using deep multi-pose representations," *2016 IEEE Winter Conf. Appl. Comput. Vision, WACV 2016*, no. 1, pp. 1–9, 2016.
- [13] N. Tsapatsoulis, V. Altxopoulos, and S. Kollias, "A vector-based approximation of KLT and its application to face recognition," *Eur. Signal Process. Conf.*, vol. 1998-Janua, no. 2, pp. 1–4, 1998.
- [14] M. Bicego, A. Lagorio, E. Grosso, and M. Tistarelli, "On the use of SIFT features for face authentication," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2006, 2006.
- [15] F. Tarsha-Kurdi, T. Landes, and P. Grussenmeyer, "Hough-Transform and Extended Ransac Algorithms for Automatic Detection of 3D Building Roof Planes From Lidar Data," *ISPRS Work. Laser Scanning 2007 SilviLaser 2007*, vol. XXXVI, no. 1, pp. 407–412, 2007.
- [16] B. Douillard, A. Quadros, P. Morton, J. P. Underwood, and M. De Deuge, "A 3D classifier trained without field samples," *2012 12th Int. Conf. Control. Autom. Robot. Vision, ICARCV 2012*, vol. 2012, no. December, pp. 805–810, 2012.
- [17] L. H. Zou, C. Jie, Z. Juan, and L. H. Dou, "The comparison of two typical corner detection algorithms," *Proc. - 2008 2nd Int. Symp. Intell. Inf. Technol. Appl. IITA 2008*, vol. 2, pp. 211–215, 2008.