CHAPTER 5

SELF OCCLUDED IMAGE MODEL

5.1 INTRODUCTION

Person heads are considered as three-dimensional objects in 3D space with only difference in the spot and in its composition. As a result, 3D face modeling is mostly admitted in face recognition application for stubborn subjects [14]. Structure from motion (SfM) and 3D face reconstruction system models a 3D shape of face using several 2D image series. Taking into consideration, this method is vulnerable to point correspondence error and dropping its performance in case of self-occluded 2D face image [15]. To remove point correspondence error, a matrix called Shape Conversion Matrix (SCM) is considered to get the true place of self-occluded Facial Feature Points (FFPs). SfM-based method guesses a 3D facial shape and a projection matrix from the corresponding 2D FFPs from many facial images [54]. It is used to get the geometry of 3D face shape from 2D images. The images are reconstructed in two ways as given below.

- To use the previous information about the sight and decrease the degree of freedom. For example, parallelism and co-planarity limitation can be employed to restructure simple geometric shapes like projected positions of line segments and planar polygons in own view.
The use of equivalent image spot in many visions. In the given face image, (Figure 5.1) using triangulation method 3D point may be reconstructed from two or more views. A significant prerequisite is the determination of calibration of the camera and the pose, which can be uttered by a projection matrix.

SfM-based methods can be classified into dense correspondence-based and sparse correspondence-based techniques respect to the density of the related 2D FFPs.

Dense correspondence-based techniques discover dense concerning 2D FFPs and recreate a dense 3D facial shape from them. If dense equivalent 2D FFPs are accurately established, then these methods can recreate a complete 3D facial shape. On the other hand, it is tedious to discover dense equivalent 2D FFPs as the cheek and forehead facial regions have no outstanding texture patterns. Sparse correspondence-based techniques discover a prearranged amount of sparse and significant equivalent 2D FFPs and recreate a sparse 3D facial shape from them. A dense 3D mean face is then tailored to the recreated sparse 3D facial figure.
A method for 3D face modeling is SfM. The 3D outline of an object is constructed with many images of the object with varied direction. The shape is computed in this technique through self-occlusion. A discrepancy matrix (SCM) computed from the superlative 3D shape is used in the existing system. This 3D shape is reconstructed using SfM. An image set is captured with varied orientations for a subject.

In the system planned adopts a new SfM technique called Multi-Stage Linear approach (MSL)[26]. The system is build by a new face alignment algorithm called Robust Alignment by Sparse and Low-rank Decomposition (RASL). An added creative feature localization method called simultaneous inverse compositional algorithm is adapted. A generalized poly cube tri-variant spline-based 3D dense mean model adaptation is incorporated. A capable structure for robust 3D face rebuilding to self-occlusion (FRMSOI) is formulated using these methods and this chapter explains the proposed model in detail.

5.2 FRMSOI MODEL

A new capable framework for 3D face reconstruction is generated and Figure 5.2 depicts the framework growth. FFPs are the 2D points that point to end point, curves and ridges of the face jointly representing the outstanding features like eyes, nose and facial outlines.
Figure 5.2 Architecture of FRMSOI Model

The 3D geometric shape is estimated by SfM and true FFPs are located by SCM. However, the system is not sufficiently efficient. Therefore, a new framework is planned using efficient algorithms and advanced techniques. Using these techniques, the FRMSOI Model is developed. This model enhances the efficiency of reconstruction of a face by overlooking the flaws in the current system.

5.2.1 Self Occlusion Elimination

The Shape Conversion Matrix (SCM) is estimated by the training phase [45] for self-occlusion elimination. The training face images are associated using robust arrangement by Sparse and Low-rank Decomposition (RASL) method. The matrix of the changed image can be divided by summing up the sparse matrix of errors and low-rank matrix of improved aligned images. The optimal set of transformations for image domain is sought through this method. It does not need pre-filtering or matching and it perform directly on input images. It is well-
mannered, scalable and concurrently aligns hundreds of images in fewer times by solving a series of convex optimization problems.

A 2D image with varied orientation is then projected. By observing FFPs and estimating the truth, the facial features are estimated. From this 2D facial shapes are formed. For obtaining the ground truth and shape of the face, the 2D model and 2D facial shape is computed. This association is employed to obtain a discrepancy matrix called SCM. It is used to locate the self-occluded FFPs in the rebuilt 3D face shape. The input experiential shape parameter is multiplied with SCM to get the true location of FFPs.

The SCM [54] is constructed by using training ground truth Facial Feature Points (FFPs) and examines FFPs from several training face images with diverse views. Let a 3D facial shape consisting of 3D FFPs in the \( k^{th} \) face check be \( S_{k}^{3D} \), then it can be written as

\[
S_{k}^{3D} = \begin{bmatrix}
x_{k1} & \cdots & x_{kn} \\
y_{k1} & \cdots & y_{kn} \\
z_{k1} & \cdots & z_{kn}
\end{bmatrix}
\]

Where \( N \) is the number FFPs and \( x_{kn}, y_{kn}, \) and \( z_{kn} \) refer to \( x, y \) and \( z \) position of \( n^{th} \) FFP in the \( k^{th} \) 3D face checks from different subjects are not well united. The 3D alignment procedure is as follows.

- Choose a \( S_{1}^{3D} \) as a reference 3D facial outline.
- The finest match is found in similar and the 3D facial shapes are aligned to the reference 3D facial outline by RASL.
• A fresh referred 3D facial outline is selected from the mean of the united 3D facial image.

• Till the residual of the 3D facial shape and the referred 3D facial image is a smaller amount than the pre-determined threshold, steps 2 and 3 are repeated.

• The 3D face checks are aligned by using the final similar transforms from step 2.

The aligned 3D facial outline is rotated to obtain the 2D facial outline in the training ground truth. Then the 2D facial shape is projected on the 2D image plane. The FFPs are manually annotated to get the observed facial outline. The ground truth 2D facial shape $G^{2D}$ is represented by the following formulation

$$G^{2D} = s_lP_lR_c(R_\theta S^{3D}_k + t_c)$$

Here $R_\theta$ is the rotation matrix. It rotates $S^{3D}_k$, $R_c$ and $t_c$ is the rotation and translation matrices correspondingly that aligns the world coordinate to the camera coordinate. The parameters $s_l$, $P_l$ are the scaling factor. The orthographic projection maps a point in the camera coordinate to the image coordinate.

The ground truth and observed 2D facial shape of the $v^{th}$ projected image is represented by the linear grouping of mean and the shape variation by means of the training ground truth 2D shapes and the Principle Component Analysis (PCA) as

$$G^{2D}_v = A_0 + \sum_{i=1}^{n} \alpha_{vi}A_i$$
\[ O_v^{2D} = B_0 + \sum_{i=1}^{m} \beta_{vi} B_i \]

Where \( \beta_v = [\beta_{v1}, ..., \beta_{vm}]^T \) observed shape parameter of the \( v^{th} \) is projected image and \( m \) is the dimension parameter.

### 5.2.2 Face Alignment

The Face Alignment is handled using Robust Alignment by Sparse and Low-rank Decomposition (RASL). The matrix rank and the \( \ell^0 \)-norm are considered as discrete valued tasks; the solution specified by the existing method is unstable if there are no sparse errors in the images. From sparse errors, recover low ranking matrices. The recovery can be done until the rank of the matrix \( A \) to be improved is neither excessively high nor the numeral of nonzero marking made in \( E \) is too huge thereby decreasing the natural convex surrogate for \( \text{rank}(A) + \gamma \|E\|_0 \) can accurately improve \( A \). This convex relaxation obtained replaces \( \text{rank}(\cdot) \) moreover with the nuclear norm or the total of the singular values and replaces the \( \ell^0 \)-norm with \( \ell^1 \)-norm. This can be integrated in RASL [82] formulation by changing constraint. The equation is as given below.

\[
\min_{A,E,\tau} \|A\|_* + \gamma \|E\|_1 \text{ s.t. } \|D \circ \tau - A - E\|_F \leq \epsilon
\]

**Outer loop of RASL Algorithm**

**Input:** Images \( I_1, \ldots, I_n \in \mathbb{R}^{w \times h} \), initial transformations \( \tau_1, \ldots, \tau_n \) in a certain parametric group \( G \), weight \( \lambda > 0 \)

**WHILE** not converged **DO**
Step 1: Calculate Jacobian matrices with respect to transformation:

\[ J_i = \frac{\partial}{\partial \zeta} \left( \frac{\text{vec}(I_i \circ \zeta)}{\|\text{vec}(I_i \circ \zeta)\|_2} \right)_{\zeta=\tau^*}, \quad i = 1, \ldots, n; \]

Step 2: Warp and normalize the images:

\[ D^{\tau^*} = \begin{bmatrix} \text{vec}(I_i \circ \tau_1) \\ \|\text{vec}(I_i \circ \tau_1)\|_2 \\ \cdots \\ \text{vec}(I_i \circ \tau_1) \\ \|\text{vec}(I_i \circ \tau_1)\|_2 \end{bmatrix}; \]

Step 3: (inner loop): solve the linearized convex optimization:

\[ (A^*, E^*, \Delta \tau^*) \leftarrow \arg \min_{A, E, \Delta \tau} \|A\|_* + \lambda \|E\|_1 \text{ s.t. } D \circ \tau + \sum_{i=1}^n I_i \Delta \tau \epsilon_i \epsilon_i^T = A + E; \]

Step 4: Update transformations: \( \tau \leftarrow \tau + \Delta \tau^*; \)

END WHILE

Output: Solution \( A^*, E^*, \tau^* \) to problem

\[ \min_{A, E, \tau} \|A\|_* + \lambda \|E\|_1 \text{ s.t. } D \circ \tau - A - E \|_F \leq \epsilon \]

### 5.2.3 Sparse Feature Extraction

The sparse feature extraction method is one of the active appearance models. Gauss-Newton gradient descent is performed by the algorithm concurrently on the warp \( p \) and appearance \( \lambda \) parameters [71]. The incremental updates to the warp \( q \) and appearance \( \lambda \) parameters are taken from \( q \) and is employed to update the warp \( W(x, p) \leftarrow W(x, p) * \text{inv}(W(x, p)) \) and appearance parameters \( \lambda \leftarrow \lambda + \lambda. \)
Pre-Computation

Step 1: Evaluate the gradient of the base appearance $\nabla A_0$ and $\nabla A_i$ for $i=1,..., m$

Step 2: Assess the Jacobian of the warp $\frac{\partial w}{\partial p}$ at $(x; 0)$

Iterate

Step 1: Warp I with $W(x; p)$ to calculate $I(W(x; p))$

Step 2: Calculate the error image $E_{app}(x) = I(W(x; p)) - [A_0(x) + \sum_{i=1}^{m} A_i(x)]$

Step 3: Compute the steepest descent images

$$SD_{sim}(x) = \left( \Delta A \frac{\partial W}{\partial p_1}, \cdots, \Delta A \frac{\partial W}{\partial p_n} A_1(x), \cdots, A_m(x) \right)$$

Step 4: Calculate the Hessian and invert it $H_{sim}^{-1} = \sum_x SD_{sim}^T(x) SD_{sim}(x)$

Step 5: Compute $\sum_x SD_{sim}^T(x) E_{app}(x)$

Step 6: Compute $\Delta q = -H_{sim}^{-1} \sum_x SD_{sim}^T(x) E_{app}(x)$

Step 7: Update the warp $W(x; p) \leftarrow W(x; p) \circ W(x; \Delta p)^{-1}$ and $\lambda \leftarrow \lambda + \Delta \lambda$

5.2.4 Motion based 3D Face Reconstruction

The 3D facial shape [79] is estimated through the multi-stage linear approach. Here pair-wise reconstructions are performed and then robustly aligned in pairs. Then they are aligned universally by calculate approximately their indefinite relative scales simultaneously. The system is incorporated with the translations
done and it copes up efficiently with the substantial outliers. Now it becomes easy to parallelize as well as remove any necessity for frequent bundle-adjustment.

- **Notation and Preliminaries**: A set of cameras with the projection matrices \( P_i \) observe a set of points in 3D \( X_i \). The \( i \)-th camera has focal length \( f_i \) and has a centre of projection \( C_i \). The camera intrinsic and camera pose (rotation, translation) are denoted as \( K_i = \text{diag}(f_i, f_i, 1) \) and \( (R_i, t_i) \) respectively.

- **Match and Image-pair Graphs**: From pair wise point matches, it forms a pruned match graph \( G_m \), consisting of nodes for each image and edges between images with better matches. Compute the full match graph \( G \) by exhaustively matching all image pairs and using the match inlier counts as the corresponding edge weights. The graph \( G_m \) is initialized to the maximum spanning tree of \( G \). Every node in \( G_r \) corresponds to an edge in \( G_m \) and represents image pairs with a sufficient number of matches. Two nodes in \( G_r \) are connected by an edge if and only if the corresponding image pair shares a camera and 3D points in common.

### 5.2.5 Feature Extraction and Matching

- **Point features extraction**: Point features are extracted using kd-tree based pair wise matching to obtain the initial two-view matches based on photometric similarity.

- **Line segments and Vanishing points (VP)**: 2D line segments are extracted through edge detection algorithms. Orthogonal regression is used to fit straight line segments to these. VP estimation in each image to repeatedly search for subsets of concurrent line segments.
• **VP and Line Segment Matching**: VPs are matched in every image pair represented in the pruned match graph $G_m$ for which a pair wise rotation can be computed. Line segments are matched using appearance as well as guided matching technique.

5.2.6 Computing Rotations

For the given three orthogonal scene directions, $d_1 = [1, 0, 0]^T$, $d_2 = [0, 1, 0]^T$ and $d_3 = [0, 0, 1]^T$, the global camera rotation in a coordinate system aligned with the $d_i$'s, can be estimated from the VPs corresponding to these directions.

$$v_{im} = \text{diag}(f_i, f_i, 1)R_i d_m$$

• **Rotations from VP Matches**: Pair wise angles between all $M$ directions are computed. Every image where at least two VPs were detected and rank the $M$ directions with decreasing weights, which is by counting the number of supporting line segments over all images where a corresponding VP was detected. For all images at least two of these directions are observed and computed using the equation $v_{im} = \text{diag}(f_i, f_i, 1)R_i d_m$.

• **Global Rotations**: If a single camera set is found, it has done. Otherwise, the $K$ camera sets must be rotationally aligned to obtain the global camera rotations. Each estimate of $q_{ij}$ provides a non-linear constraint relating the unknown rotations of the two camera sets denoted by $q^a$ and $q^b$ respectively.

$$q^a = (q^a_i \cdot q_{ij} \cdot (q^b_j)^{-1}) q^b$$
Where \( q_i^a \) and \( q_i^b \) denotes the known rotations of the \( i \)-th and \( j \)-th camera in their own camera sets. In the absence of VPs, rotations can be recovered via the essential matrices obtained from pair wise point matches for image pairs with an adequate number of matches.

### 5.2.7 Linear Reconstruction

When the intrinsic \( K_i \) and rotations \( R_i \) are known, every 2D image point \( x_{ij} \) can be normalized into a unit vector \( \tilde{x}_{ij} = (K_i R_i)^{-1} x_{ij} \), which is related to the \( j \)-th 3D point \( X_j \) as

\[
\tilde{x}_{ij} = d_{ij}^{-1} (X_j - C_i)
\]

Where \( d_{ij} \) is the distance from \( X_j \) to the camera centre \( C_i \).

- **Two-view Reconstruction**: A pair wise reconstruction for cameras \((a, b)\), treated as a translating pair, is denoted as \( R^{ab} \) is \( R^{ab} = \{C_a^{ab}, C_b^{ab}, \{X_j^{ab}\} \} \) where superscript denotes a local coordinate system. Under pure translation, the epipole in the two images coincide and all points in the two views \( x_{aj} \) and \( x_{bj} \) are collinear with the common epipole also known as focus of expansion (FOE) \( x_{aj}^T [e]_{x} x_{bj} = 0 \). The epipole \( e \) is a vector that points along the baseline for the translating camera pair. Estimating \( e \) by finding the smallest eigenvector of a matrix produced by summing the outer product of all 2D lines \( 1 = x_{aj} \times x_{bj} \) and then chooses \( C_a^{ab} = 0 \) and \( C_b^{ab} = \hat{e} \) corresponding to a unit baseline. Each point \( X_j^{ab} \) is then triangulated using the following linear method.
\[ x_{kj} \times (X_j^{ab} - C_k^{ab}) = 0, \quad \text{for } k \in \{a, b\} \]

- **Robust Alignment**: Each pair wise reconstruction \( R^{ab} \) involving cameras \((a, b)\) differs from a global reconstruction by 4-dofs are unknown scale \( s^{ab} \) and translation \( t^{ab} \), unique up to an arbitrary global scale and translation. Suppose, \( R^{bc} \) and \( R^{ab} \) share camera \( b \) and some common 3D points to robustly align \( R^{ab} \) to \( R^{bc} \) by computing 4-dofs 3D similarity \( S^{ab}_{bc} \). A hypothesis is generated from two 3D points common to both reconstructions and it has been chosen randomly sampling two common 3D points or one common point when the camera centre of \( b \) is selected as the second point. The exact correspondence for one of the two points in \( R^{bc} \) and \( R^{ab} \) gives a translation hypothesis \( t \). The scale \( s \) is computed by minimizing the image distance between the observed and re-projected points for the second 3D point. The hypothesis \((s, t)\) is then scored using the total symmetric transfer error for all common 3D points in all three images is given below.

\[
\sum_k d \left( x_{kj}, f_k^{ab} \left( S^{-1}X_j^{bc} \right) \right) + \sum_k d \left( x_{kj}, f_k^{bc} \left( SX_j^{bc} \right) \right)
\]

The function \( f_k^{ab} \) projects a 3D point into each of the two cameras of \( R^{ab} \) where \( k \in \{a, b\} \), \( f_k^{ab} \) is defined similarity for \( R^{bc} \), and \( d \) robustly measures the distance of the projected points from the original 2D observations \( x_{kj} \), where \( k \in \{a, b, c\} \).

- **Global scale and translation estimation**: Once a sufficient number of transformations \((s^{ab}_{bc}, t^{ab}_{bc})\) between reconstructions \( R^{ab} \) and \( R^{bc} \) are known, their
absolute scale and translations, denoted by \((s^{ab}, t^{ab})\) and \((s^{bc}, t^{bc})\), can be estimated using the relation,

\[
s^{bc}X + t^{bc} = s^{bc}_a(s^{ab}X + t^{ab}) + t^{bc}_a
\]

Where \(X\) is an arbitrary 3D point in global coordinates and eliminating \(X\), gives us four equations in eight unknowns

\[
W^{bc}_{ab}(s^{bc} - s^{bc}_a s^{ab}) = 0,
\]

\[
W^{bc}_{ab}(s^{bc} t^{bc}) = W^{bc}_{ab}(s^{bc}_a t^{ab} + t^{bc}_a)
\]

Here, the weight \(W^{bc}_{ab}\) is set to the number of three view consistent points found common to \(R^{ab}\) and \(R^{bc}\). The scale of any one reconstruction is set to unity and its translation set to zero to remove the global scale and translational ambiguity.

### 5.2.8 3D Dense Mean Model Adaptation

Generalized Poly Cube (GPC) tri-variate spline [77] is a universal uni-piece parametric demonstration in terms of shape with non-trivial topology that stops unnecessary artifacts created by stitching or trimming over of common borders. Concrete primitives are employed. This framework of the whole spline construction gives an alteration sequentially and also in detail control [39]. The global structure is abstracted through the GPC-graph. Each node in the GPC-graph denotes a native cube and the edge among two neighboring nodes denote the overlapping or gluing of equivalent cubes.
**GPC construction and parameterization**

The GPC-graph and parameterization takes one concrete model as a common input and computes simultaneously as follows.

- **Topological breakdown:** The model is partitioned to numerous components as a result that every component has trivial topology and homology. Based on this closed genus-g plane is calculated routinely. The surface is broken into a set of pants pieces canonically. Further each volumetric primitive bounded by every pants piece is broken into 4 topological cubes.

- **Geometric Decomposition:** Detect long and thin branches for each volumetric primitive using the shape skeleton. Then remove long branches one by one and parameterize it to form an only cube domain.

- **Cube Parameterization:** Parameterize every sub-region using a cube domain.

**Point based splines**

Each control point \( C_i \) is related with three bond vectors along with three major axis guidelines \( r = [r_1, r_2, r_3, r_4, r_5] \), \( s = [s_1, s_2, s_3, s_4, s_5] \), \( t = [t_1, t_2, t_3, t_4, t_5] \), where \( r_3 = 0, s_3 = 0 \) and \( t_3 = 0 \). For some example point with \( (u, v, w) \) as its parameter, the blending function is given by
$B_i(u, v, w) = N_r(u) \times N_s(v) \times N_t(w)$ Where $N_r$, $N_s$ and $N_t$ are cubic B-spline basis functions related with the bond vector $r$, $s$ and $t$ correspondingly. The formulation for a PB-spline on this point is specified by

$$P(u, v, w) = \frac{\sum c_i B_i(u, v, w)}{\sum B_i(u, v, w)}$$

**Spline Hierarchical Spline Fitting Algorithm**

For all cube domain $D_i$ do

Set $H_0^i$ and control points, $k=0$

End for

Loop

Step 1: Navigate and acquire know vectors for all control points.

Step 2: Estimate the spline function $P(u) = \frac{\sum c_k B_k(\phi^{ij}(u')-c_k^i)}{\sum c_k B_k(\phi^{ij}(u')-c_k^i)}$ for all sample points.

Step 3: Spline Fitting: Minimize $E_{dist} = \sum_{i=0}^{n} \|P(u_i) - f(u_i)\|^2$

Step 4: Calculate the fitting error of each cell:

For all cells $H_f^k$ in level $k$ do

If Error $H_f^k \geq$ tolerance then

Subdivide $H_f^k$ and add new control points.

End if

End for
If no cell is subdivided then

Stop.

End if

\[ k \leftarrow k + 1 \]

End loop

Where \( u^j \) is the native parametric coordinate of point \( u \) in the cube domain \( D^j \), \( Q^{ij} \) is the transition function from cube domain \( D^j \) to \( D^i \), \( C^i_k \) indicates the control point \( k \) in the cube domain \( D^i \), and \( C^i_k \) is its native coordinate.

### 5.3 IMPLEMENTATION AND RESULT ANALYSIS

The FRMSOI design was experienced by FacePix image dataset. Face datasets like FERET, XM2VTS, the CMU PIE Dataset, the Yale Face Dataset, the Yale B Database, and the MIT Dataset have been famous for estimating the performance of face recognition algorithms. These dataset offers face images captured in a collection of poses and in various lighting angles. Though, they do not have sets of face images in a range of poses or in varied illumination angles. FacePix dataset is employed to estimate research aimed at generating robust face detection and recognition, and also to give a face dataset shown in Figure 5.3 that allows for preciseness in measurements for robust image and comparative performance with reverence to the variation in poses and lighting.

The model is executed in steps. First, the model executes with initial prompting. Secondly, the model pulls out the feature points by using concurrent
inverse compositional algorithm. Figure 5.4 demonstrates the feature points of the input subject. In the third step, it reconstructs the 3D face through multi-stage linear approach and the output is exposed in Figure 5.5.

The features of the 2D face images obtained from image facepix dataset are taken out and then aligned. Then a 3D model is rebuilt from the features of 2D image. Self-occlusion is eradicated and a dense mean model with texture of 2D image is tailored on 3D face model. The Root Mean Square (RMS) error value is obtained to compare the result of the FRM SOI model with available datasets. The RMS value is the squared value of the quantity obtains over an interval and the square root of the mean value.

5.3.1 Result Analysis

The formula employed to compute the value is,

\[
RMS_{error\ value} = \sqrt{(os - cs)^2 / countof(os)}
\]

where \(os\) is the original share and \(cs\) is the current shape. Reduced set vectors (RSV) are a ratio of the RMS value of the feature point that approximates the decision function. As feature points very much affect the reconstruction and the self-occlusion happening is increased. It is vital to estimate the effectiveness of the feature extraction procedure. RSV is computed as,

\[
RSV = \frac{2*D* RMS*A}{2*D*RMS*B}
\]

Here \(A\) is error at self-occluded FFPs and \(B\) is error at visible FFPs bilateral position to the self-occluded FFPs.
Figure 5.3  Model Faces (Pose and Diverse Angles), (a), (d), (g) and (j) pose; (b), (e), (h), and (k) pose +30°; (c), (f), (i) and (l) pose –30°
Table 5.1 depicts +30° value of RSV, 0° of RSV and –30° of RSV rotated input images. It also reveals the fault value RMS of the face rebuilt and processing time of SCMM framework. Table 5.2 depicts the same value and RMS error value in the proposed FRMSOI model. Figure 5.6 shows the comparative graph for the processing time of both the model. Figure 5.7 depicts the compared RMS value of both the model.

Figure 5.4 Extraction of Feature Points, (a) Feature Points of Frontal Image (b) Feature Points of -30 Degree Orientation (c) Feature Points of 30 Degree Orientation
GPC trivariate spline method is used with 3D dense mean model in FRMSOI model. The plotted graph as given in Figure 5.8 reveals that the SCM model has an edge above the SCMM framework. Table 5.3 shows the average RSV value for $+30^\circ$ and $-30^\circ$ degree rotated input image.

![Figure 5.5 3D Reconstructed Face](image)

**Figure 5.5 3D Reconstructed Face**

![Figure 5.6 Processing Time for Various RSV](image)

**Figure 5.6 Processing Time for Various RSV**
Table 5.1 RSV, RMS and Processing Time for SCMM Framework

<table>
<thead>
<tr>
<th>Subject</th>
<th>RSV of +30°</th>
<th>RSV of 0°</th>
<th>RSV of –30°</th>
<th>Time (s)</th>
<th>RMS (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0251</td>
<td>0</td>
<td>1.0365</td>
<td>10.3654</td>
<td>2.0105</td>
</tr>
<tr>
<td>2</td>
<td>1.1548</td>
<td>0</td>
<td>1.198</td>
<td>10.5986</td>
<td>1.6587</td>
</tr>
<tr>
<td>3</td>
<td>1.2154</td>
<td>0</td>
<td>1.2096</td>
<td>10.3485</td>
<td>1.8524</td>
</tr>
<tr>
<td>4</td>
<td>1.0986</td>
<td>0</td>
<td>1.1024</td>
<td>10.3475</td>
<td>1.9654</td>
</tr>
<tr>
<td>5</td>
<td>1.2147</td>
<td>0</td>
<td>1.0214</td>
<td>10.2157</td>
<td>1.7436</td>
</tr>
<tr>
<td>6</td>
<td>1.1548</td>
<td>0</td>
<td>1.1648</td>
<td>10.3658</td>
<td>1.5416</td>
</tr>
<tr>
<td>7</td>
<td>1.1485</td>
<td>0</td>
<td>1.1371</td>
<td>10.3568</td>
<td>1.6874</td>
</tr>
<tr>
<td>8</td>
<td>1.0615</td>
<td>0</td>
<td>1.0879</td>
<td>10.3248</td>
<td>1.6594</td>
</tr>
<tr>
<td>9</td>
<td>1.1462</td>
<td>0</td>
<td>1.1689</td>
<td>10.3758</td>
<td>1.4875</td>
</tr>
</tbody>
</table>

Table 5.2 RSV, RMS and Processing Time for FRMSOI Framework

<table>
<thead>
<tr>
<th>Subject</th>
<th>RSV of +30°</th>
<th>RSV of 0°</th>
<th>RSV of –30°</th>
<th>Time (s)</th>
<th>RMS (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.8475</td>
<td>0</td>
<td>0.7932</td>
<td>7.668</td>
<td>0.0659</td>
</tr>
<tr>
<td>2</td>
<td>0.8326</td>
<td>0</td>
<td>0.8365</td>
<td>7.7796</td>
<td>0.0357</td>
</tr>
<tr>
<td>3</td>
<td>0.931</td>
<td>0</td>
<td>0.8572</td>
<td>7.5986</td>
<td>0.0425</td>
</tr>
<tr>
<td>4</td>
<td>0.8247</td>
<td>0</td>
<td>0.8429</td>
<td>7.6897</td>
<td>0.0365</td>
</tr>
<tr>
<td>5</td>
<td>0.8364</td>
<td>0</td>
<td>0.8716</td>
<td>7.6941</td>
<td>0.0642</td>
</tr>
<tr>
<td>6</td>
<td>0.8647</td>
<td>0</td>
<td>0.8925</td>
<td>7.5971</td>
<td>0.0548</td>
</tr>
<tr>
<td>7</td>
<td>0.8934</td>
<td>0</td>
<td>0.8684</td>
<td>7.7324</td>
<td>0.0325</td>
</tr>
<tr>
<td>8</td>
<td>0.9657</td>
<td>0</td>
<td>0.9357</td>
<td>7.6914</td>
<td>0.0497</td>
</tr>
<tr>
<td>9</td>
<td>0.8147</td>
<td>0</td>
<td>0.8293</td>
<td>7.6748</td>
<td>0.0525</td>
</tr>
</tbody>
</table>
Table 5.3 Average RSV

<table>
<thead>
<tr>
<th>Model</th>
<th>Avg. RSV of +30°</th>
<th>Avg. RSV of 0°</th>
<th>Avg. RSV of –30°</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCMM</td>
<td>1.1879</td>
<td>0</td>
<td>1.141</td>
</tr>
<tr>
<td>FRMSOI</td>
<td>0.8505</td>
<td>0</td>
<td>0.8841</td>
</tr>
</tbody>
</table>

Table 5.4 Average Time Required for Processing and RMS

<table>
<thead>
<tr>
<th>Model</th>
<th>Avg. time of reconstruction (s)</th>
<th>Avg. RMS of reconstructed 3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCMM</td>
<td>10.3483</td>
<td>1.6343</td>
</tr>
<tr>
<td>FRMSOI</td>
<td>7.681</td>
<td>0.0466</td>
</tr>
</tbody>
</table>

Table 5.4 shows the comparison of the average of the processing time for reconstruction and the average RMS of the 3D face for both the model. It is clear from the table that via SCM with GPC trivariate spline and 3D dense mean model adaptation consumed less time than the SCMM framework. From Table 5.2, it is

Figure 5.7 Matching RMS Error Value of Various RSV
clear that the processing time is reduced in FRMSOI model and minimal RMS value when compared to the SCMM framework. This proves enhanced efficiency in the model proposed for 3D face recognition and face reconstruction.

Figure 5.8  RSV of Input Image (a) RSV in +30 (b) RSV in −30 Degree
5.4 SUMMARY

SfM-based method that is strong to self-occlusion is generated for 3D face reconstruction. This is developed using suitable dataset (FacePix) for enhanced efficiency while analyzing various applications. The model uses RASL for robust Face Alignment. The Simultaneous inverse compositional algorithm used for Sparse Feature Extraction and Multi-Stage Linear Approach used for Structure from Motion based 3D Face Reconstruction. Further, GPC Trivariate Spline employed for 3D Dense Mean Model Adaptation. RSV and RMV parameters are used for comparison of both the model. The proposed FRMSOI model improves the efficiency of the 3D reconstruction based face recognition.