**Overview of Simple Multi Person Linear Model**

**SMPL**



*Krishna Satish D S*

**Contents**

[**Abstract 3**](#_Toc56158008)

[**Motivation 3**](#_Toc56158009)

[**Related Work: 5**](#_Toc56158010)

[**Model Formulation: 6**](#_Toc56158011)

[**Training: 8**](#_Toc56158012)

[**SMPL Evaluation 10**](#_Toc56158013)

[**DMPL 11**](#_Toc56158014)

[**Limitations 11**](#_Toc56158015)

[**Future work 12**](#_Toc56158016)

[**Conclusion 12**](#_Toc56158017)

# Abstract

This work focuses on creating a realistic human body models that can represent different body shapes, deform naturally with pose, and exhibit soft-tissue motions like those of real humans and making sure that these models are fast to render, easy to deploy, and compatible with existing rendering engines.

The current trend in industry is to manually sculpt the models by hand rigging the mesh which demands huge amount of time and cost to create a realistic models, one good example is a movie of “Fast and Furious” , one of the final scene was synthetically created to mimic a person’s face and it costed around 50 M$ for footage of few seconds,

In order to solve this problem automatically without much manual effort researchers have worked on creating a statistical model learnt from many 3D scans of human subjects but unfortunately these methods cannot be ported to current rendering engines which works on skinning techniques.

This motivated the authors to create a mathematical model which automatically learn a body model which is simple, realistic, and compatible with the current rendering engines all at the same time. This led to the introduction of SMPL (Skinned Multi-Person Linear) model,

# Motivation

Classical methods model the mapping between the vertices and underlying skeleton structure. One such example of this approach is Basic linear blend skinning (LBS), these models are the most widely used because of their simple and efficient rendering,

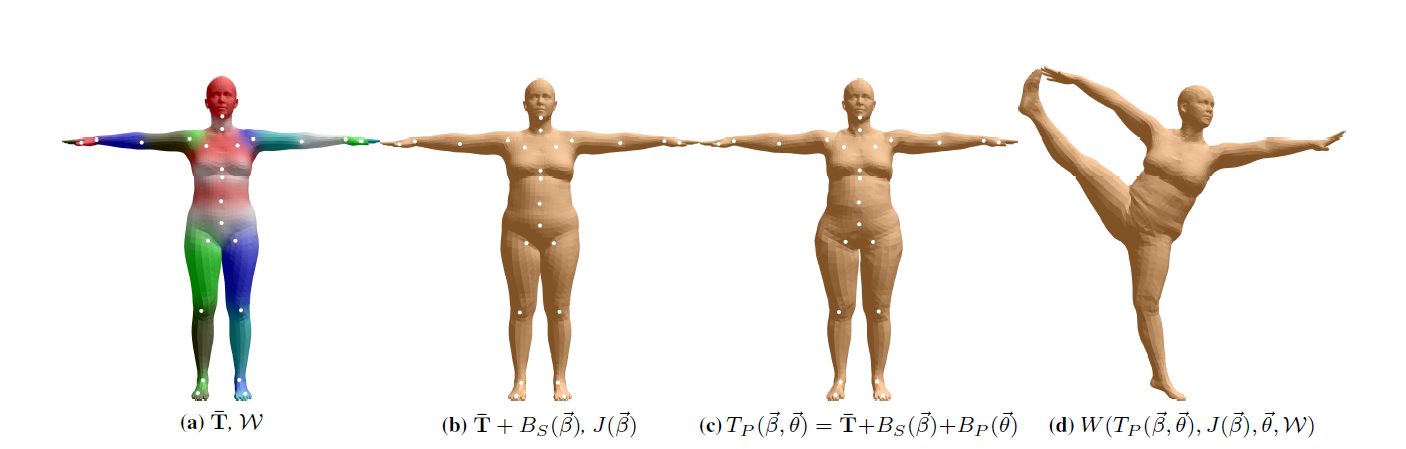
Due to this simplicity of LBS it is not very accurate and efficient when it comes to skinning of the areas where the mesh significantly deforms (like elbow areas) resulting in the creation of “bowtie” effects on the final render as shown in the picture below,

****

Even though there were tremendous amount of research has been happened to fix these types of anomalies, most of the solutions either fails to look like human body deformation or it is not compatible with the current rendering engines,

In contrast to all the previous works, SMPL model learns blend shapes to correct for the limitations of standard skinning methods.

Different blend shapes for identity, pose, and soft-tissue dynamics are additively combined with a rest template mesh before being transformed by blend skinning.



A key component of this approach is that the pose blend shapes are formulated as a linear function of the elements of the rotation matrices which is bounded and hence makes the model generalize better.

In this approach, the authors have implemented an objective function that penalizes the pervertex disparities between registered meshes and our model.

In order to learn how humans deform with various body poses, the authors have used 1786 high-resolution 3D scans of different subjects in a wide variety of poses and they had aligned the template mesh to each scan to create a training set. Optimization was done on the blend weights, pose-dependent blend shapes, the mean template shape (rest pose), and a regressor from shape to joint locations to minimize the vertex error of the model on the training set.

This work has been possible by creating a linear model based on CAESAR dataset which consists of thousands of body scans of both male and female body shapes, the first step in this process of creating a linear model is to perform PCA(Principle Component Analysis) of each and individual scans and pose normalize this data(which is crucial for training SMPL model).

Both BlendSCAPE model, which is primarily a deformation based model and SMPL model which is a vertex based model was trained on same dataset and the performance was evaluated both quantitatively(vertex errors) and qualitatively(animating and visual inspection) , two different types of skinning methods were employed namely LBS(Linear Blend Skinning) and DQBS(Dual- Quaternion blend skinning), even though the LBS is relatively naive method of skinning it has been shown that it outperforms the more complicated deformation based BlendSCAPE model.

This learnt body model is only capable of showing static soft tissue deformation and hence it fails to reproduce the realistic body motion during activities like jumping and jiggling, in order to solve this limitation this model has been extended to recreate dynamic soft tissue deformations by utilizing Dyna model and hence this extended model is called Dynamic-SMPL or DMPL in short.

One thing to note here is that the Dyna model is based on triangular deformations while the DMPL model is a vertex based deformations, DMPL model is trained on the same 4D dataset as that of Dyna, in this approach the vertex errors between the vertices of SMPL model’s mesh and Dyna model’s mesh are computed to produce dynamic blend shapes, then the model is trained based on angular velocities and acceleration of body parts, since the soft-tissue deformations vary w.r.t the body shapes and BMI, DMPL has been trained to account for these parameters as well to create more realistic mesh deformation.

Since SMPL and DMPL models use simple vertex-based deformations it is both low on computation and compatible with most of the rendering engines available today like Unreal, Unity etc...

# Related Work:

There were many attempts to automatically rig the models to achieve realistic looking models with varying degree of success, few of them are discussed below,

**Blend Skinning**: Probably the most famous and simple type of skinning methods is called as Blend Skinning where each vertex in the mesh surface is transformed using a weighted influence of its neighboring bones. Since this only considers the simple mesh deformation it produces significant artifacts which leads to unrealistic body movements.

**Blend shapes** : also known as “Scattered data interpolation” was introduced in order to overcome the limitations of basic blend skinning methods, the pose space deformation model(PSD) was introduced which defines deformations (as vertex displacements) relative to a base shape, where these deformations are a function of articulated body pose.

One limitation of this approach is that one must manually sculpt the mesh for various poses before rendering.

In order to overcome the above mentioned limitation another model was introduces to automatically mesh for deformations, in this approach Allen et al. [2002] has only considered the torso and arms area to train the model and hence discarding the full body scans and hence this approach is not fully recreate the deformation across entire human body,

All the above-mentioned methods focuses on learning pose blend shapes for a single body shape, however the author’s focus in this study is to make the model generic so that it will work for all poses of different shapes and sizes.

**SCAPE:** The most successful work to represent body shapes and pose dependent shapes is known as SCAPE, these models depends on triangular deformations rather that simple vertex deformation hence these types of representations are not compatible with the current rendering engines,

The final goal of this work is to derive a vertex-based learnt model which is as powerful as triangular deformation models which are efficient in representing wide range of body poses.

# Model Formulation:

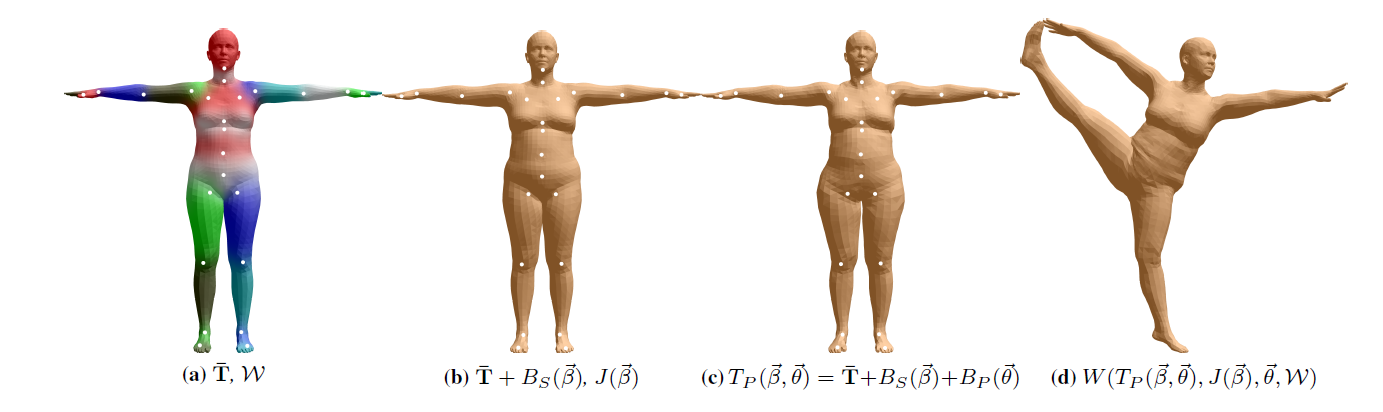
In SMPL, a single blend shape is represented as a vector of concatenated vertex offsets, we first use the manually sculpted mesh with N = 6890 vertices with 23 joints,

By convention, a model is defines with mean template shape T with a rest pose theta(\*) and a set of blend weights( )and a blend shape function B\_s(beta) where beta is the shape parameter, another function which predicts K joint locations are defined as J(beta) and also we have an exclusive function to compensate for the pose dependent blend shapes B\_p(teta),

All these functions are added together and then it is added to the rest pose template mesh and this updated mesh is passed into a skinning function W to rotate the vertices around the estimated joint centers with smoothing defined by the blend weights., which generates a skinned model as shown below, the resulting function is as given below



Here is the learnable parameter which is responsible in correcting our mesh according to various poses and shapes,

****

As shown above on the left we have a template base mesh(T) with certain blend weights(W) [a], we add shape dependent deformations to the base mesh[b], we also add a pose related mesh in [c] to represent the full body deformation.

**Blend skinning:**

The pose of the body is represented from the rigged internal skeleton, each of K joints are articulated with the help to axis-angle representation of each joint denoted by,



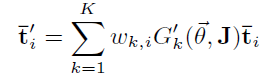
Since we have 23 joints in our parent tree, our pose parameters have 72 parameters which consists of the following,

* 3 parameters for Global co-ordinate transformation,
* 23\*3 parameters for each individual join in [x, y, z] direction

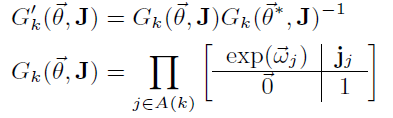
Each joint’s corresponding axis-angle is transformed into a rotation matrix using Rodrigues formula as shown below,



Using the above formula each vertex t\_i is transformed into t\_i` by the following formula,

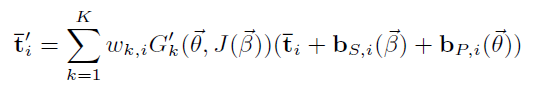


Where G\_k(teta,J) is given by,

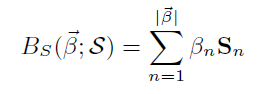


where w\_k, is an element of the blend weight matrix W, representing how much the rotation of part k effects the vertex i, exp(teta\_j) is the local 3x3 rotation matrix corresponding to joint j, Gk(teta, J) is the world transformation of joint k, and G\_k`(teta, J) is the same transformation after removing the transformation due to the rest pose,

after taking all blend shapes and pose dependent weights in consideration we could write the transformation matrix as follows,

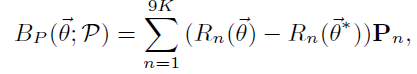
****

The body shape varies from one person to another, we represent this as a linear function of Bs(beta) as shown,



Here the S is a learnt parameter which fully represents the changes in shape in template mesh due to changes in shapes.

Pose blend shape is formulated as shown below,



Here, R\_n is a function that maps a pose vector teta to rotation matrix represented by exp(w), Since our skew symmetric matrix in Rodrigues formula has 3x3 matrix(9 elements) and we have k=23 joints, totally we have 23x9 = 207 pose blend shapes.

Joint location is a function of body shape, in other words the join locations varies depending on the build of the body. it is formulated as shown below,



where is a matrix that transforms rest vertices into rest joints, the regression matrix is learnt from examples of different people

# Training:

We train the SMPL on two datasets,

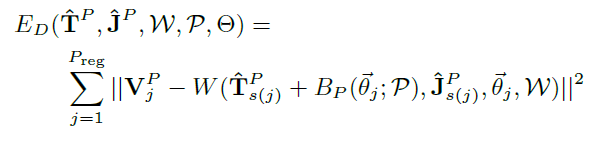
* Multi-pose
* Multi-shape

The first step is to align these poses and shapes into our template mesh so that our model could accurately represent the 3D scan and hence learn from it, this process is called as “registration”.

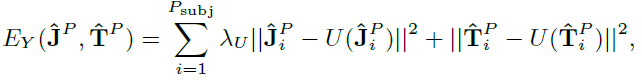
Since our model decomposes shape and pose, we train these separately, which helps in simplifying the optimization process. We first train {J, W, P} using our multi-pose dataset and then train {`T, S} using our multi-shape dataset.

**Pose Training**:

To understand how well our template mesh can represent the body scans from multi-pose dataset, we simply calculate the Euclidian distance between the vertices of the registered mesh and 3D model’s mesh using the following formula,

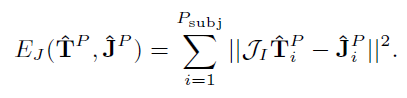


To prevent over-fitting of our model, we use many regularization techniques, one such technique is the penalize left-right symmetry as shown below,

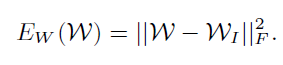


Where U(T) finds the mirror image of the vertices in template T.

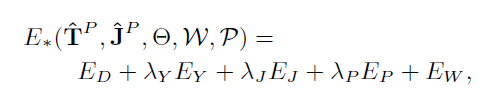
This model is manually segmented into 24 parts, we use this segment to predict the joint centers and regression matric J\_i When estimating the joints for each subject we regularize them to be close to this initial prediction,



We also regularize the blend weights towards the initial weights, W\_I,



By combining everything we get the following equation,

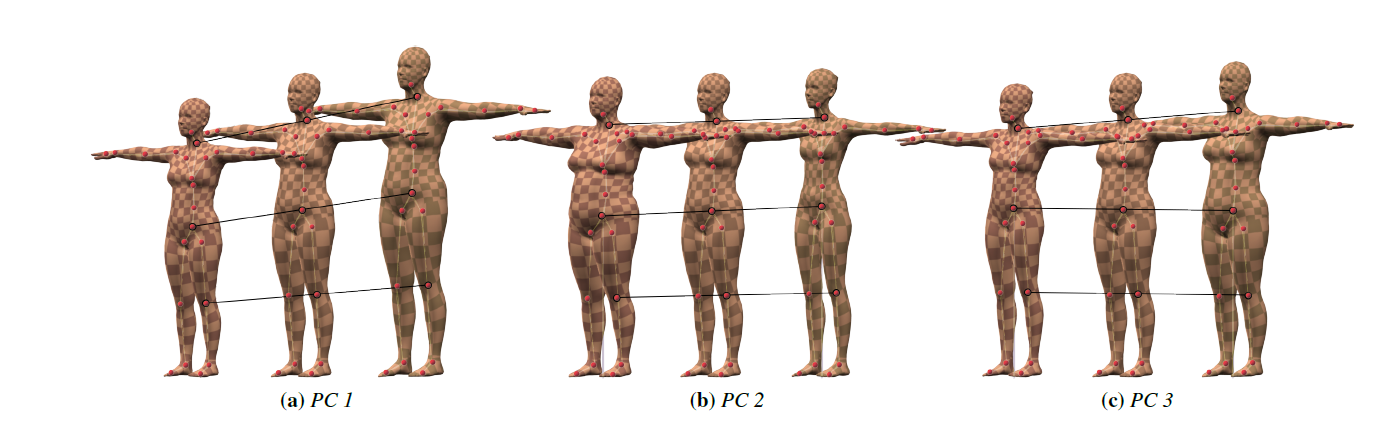


To predict the regressor matrix J, Non-negative least square method with the weights adding up to one is used. Making weights positive and add to one discourages predicting joints outside the surface.

**Shape Parameter Training:**

After pose normalizing the scans from multi-shape database we run a PCA(Principle Component Analysis) to obtain a shape space represented by {T,S},

The pose normalization transforms a raw registration (V\_j)^s into registration (T\_j)^s in the rest pose theta,

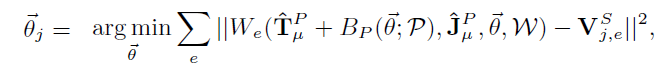


To pose-normalize a registration, VS j , we first have to estimate its pose.

Let, the below equation represent the edges between the model and registered mesh.

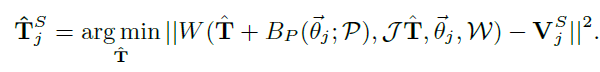


Where, represent the mean shape and mean pose from the multi-shape dataset.An edge is obtained by subtracting a pair of neighboring vertices. To estimate the pose using an average generic shape we minimize the following,



Once the pose teta\_j is known, we calculate the shape that, when posed, matches the training

Registration using the following equation,



This optimization of pose is very important when building a shape basis from vertices. Without this, the pose variations of the subjects in the shape training dataset would be captured in the shape blend shapes. The resulting model would not be decomposed properly into shape and pose.

# SMPL Evaluation

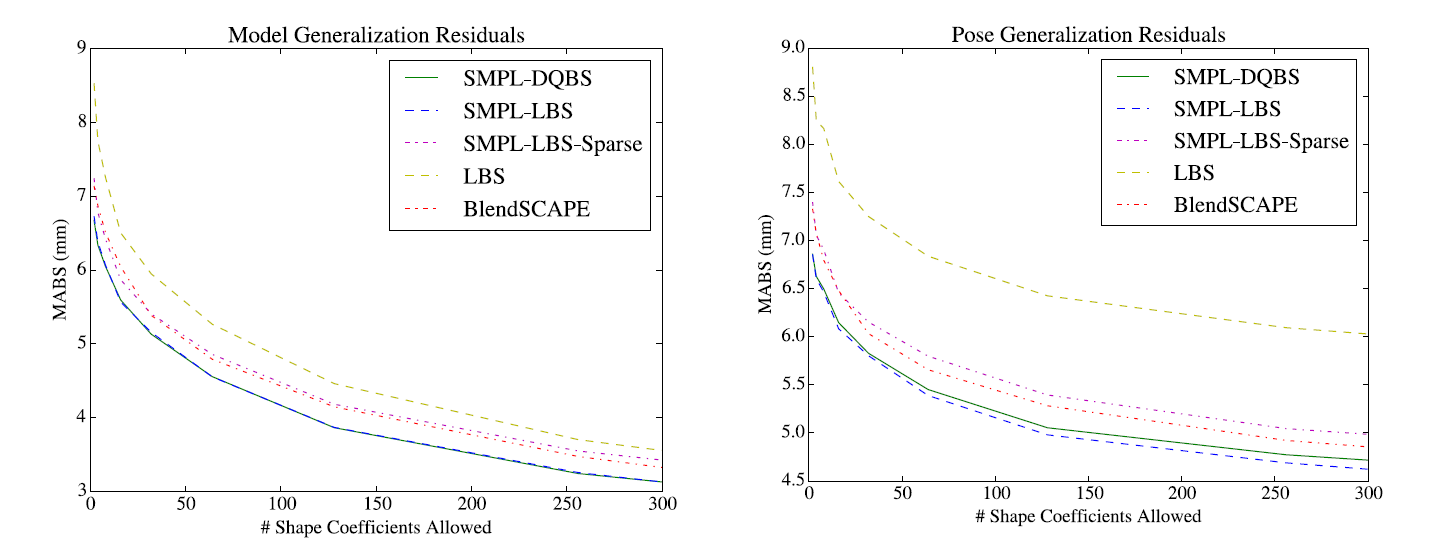
This model has been evaluated both qualitatively and quantitatively to test our model’s behavior,

In quantitative evaluation we perform two main task, i.e

* Model generalization: it is the ability of the model to fit to meshes of new people and poses; this test both shape and pose blend shapes
* Pose generalization is the ability to generalize a shape of an individual to new poses of the same person

To measure model generalization, we first fit each model to each registered mesh, optimizing over shape beta and pose teta to find the best fit in terms of squared vertex distances.

For pose generalization, we take each individual mesh, select one of the estimated body shapes from the generalization task, and then optimize the pose, theta, for each of the other meshes of that subject, keeping the body shape fixed. The assumption behind pose generalization is that if a model is properly decomposed into pose and shape, then the model should be able to fit the same subject in a different pose, without readjusting shape parameters



In case of qualitative analysis, the mesh is animated and visualized by a human observer to judge its performance.

# DMPL

While SMPL models static soft-tissue deformations with pose it does not model dynamic deformations that occur due to body movement and impact forces with the ground. Given 4D registrations that contain soft-tissue dynamics, we fit them by optimizing only the pose of a SMPL model with a personalized template shape. Displacements between SMPL and the observed meshes correspond to dynamic soft-tissue motions. To model these, we introduce a new set of additive blend shapes that we call dynamic blend shapes. These additional displacements are correlated with velocities and accelerations of the body and limbs rather than with pose. We follow the approach of Dyna,

We extend our linear formulation and simply add the dynamic blend shape function, B\_D(beta,teta), to the other blend shapes in the rest pose before applying the skinning function. The shape in the zero pose becomes



# Limitations

The pose-dependent offsets of SMPL are not dependent on body shape. This generic approach would likely not work if we were to model a space of nonrealistic animated characters in which body part scales vary widely, or a space of humans that includes infants and adults. This limitation could be addressed by training a more general function.

As described, the SMPL model is a function of joint angles and shape parameters only: it does not model breathing, facial motion, muscle tension, or any changes independent of skeletal joint angles and overall shape.

We manually define the segmentation of the template into parts, the topology of the mesh, and the zero pose. Theoretically these could also be learned but we expect only marginal improvements for significant effort.

# Future work

SMPL uses 207 pose blend shapes. This could likely be reduced by performing PCA on the blend shapes. This would reduce the number of multiplications and consequently increase rendering speed. Also, our dynamic model uses PCA to learn the dynamic blend shapes, but we could learn the elements of these blend shapes directly as we do for the pose blend shapes

# Conclusion

The goal was to create a skeletally driven human body model that could capture body shape and pose variation as well as, or better than, the best previous models while being compatible with existing graphics pipelines and software. To that end, SMPL uses standard skinning equations and defines body shape and pose blend shapes that modify the base mesh. We train the model on thousands of aligned scans of different people in different poses. The form of the model makes it possible to learn the parameters from large amounts of data while directly minimizing vertex reconstruction error. Specifically, we learn the rest template, joint regressor, body shape model, pose blend shapes, and dynamic blend shapes. Using 4D registered meshes we extended SMPL to model dynamic soft-tissue deformations as a function of poses over time using an autoregressive model.