DIPLOMATIC THESIS

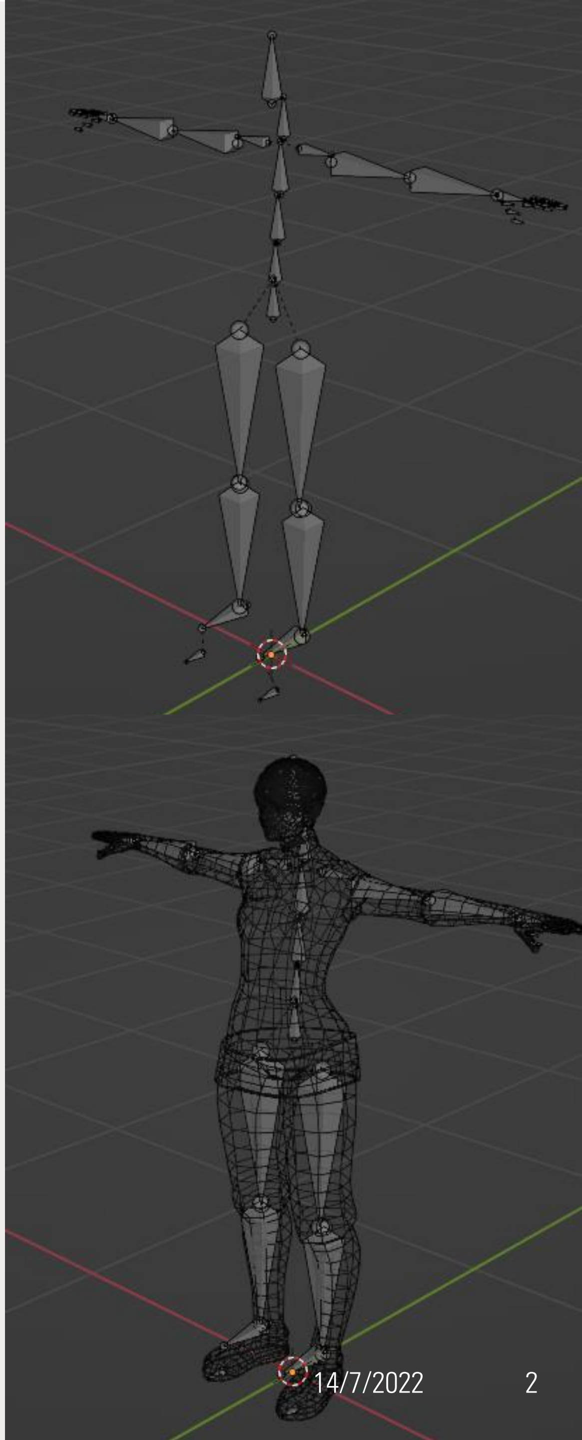
METHODS FOR EFFICIENT CHARACTER ANIMATION

Department of Computer Science and Engineering
University Of Ioannina

Author: Dimokas Angelos
Supervisor: Ioannis Fudos

14/7/22

CONTENTS



SUMMARY

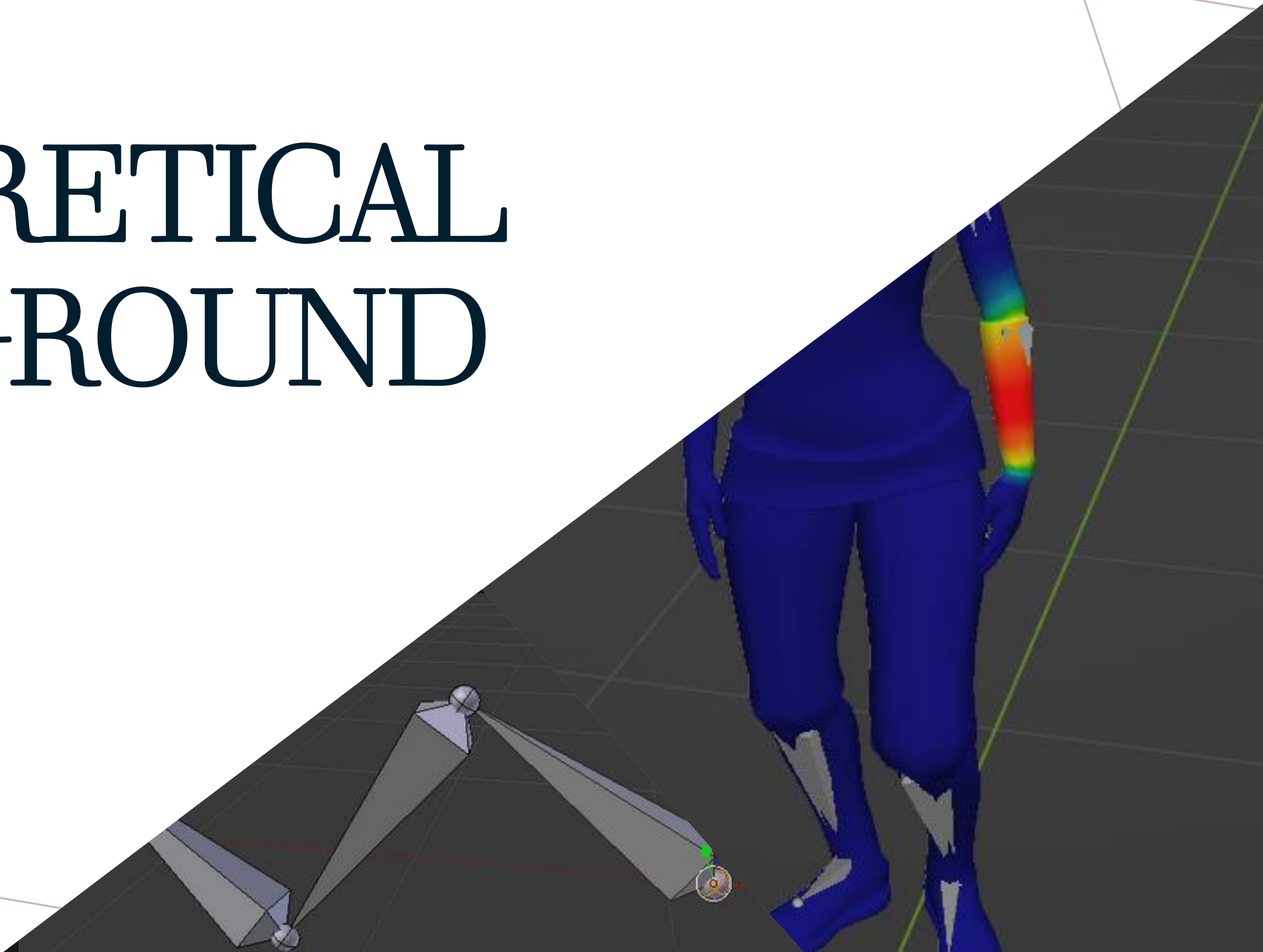
We used the Blender3d software and python3 programming language to implement the virtual model's armature and weights approximation algorithms.

We developed 2 approximation methods, one linear and one non-linear, which we tested on 3 humanoid models to approximate armature and weights. Finally we compared results and deviations from the actual values of their mesh vertices.

PURPOSE

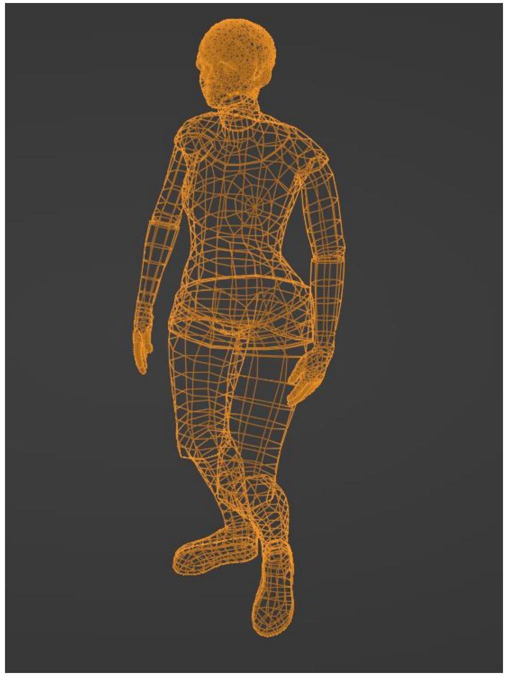
- **Development** of methods to optimally approximate the armature and the influence weights of the vertices and bones of a virtual character, for a simulation of its mesh animation
- **Reducing the required bandwidth** for the simulation of the virtual model's animation.
- **Comparing results** of the approximation methods in terms of their efficiency.

THEORETICAL BACKGROUND



BASIC CONCEPTS OF VIRTUAL MODELS

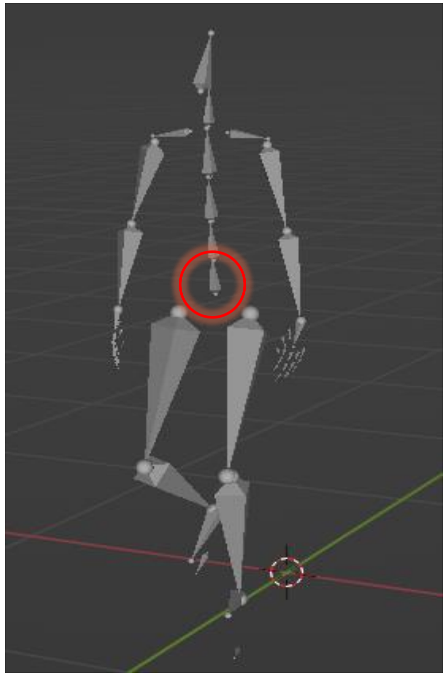
VERTICES



$$v = [\ v_x \quad v_y \quad v_z \quad 1\ ]$$

N = Mesh total **vertices**

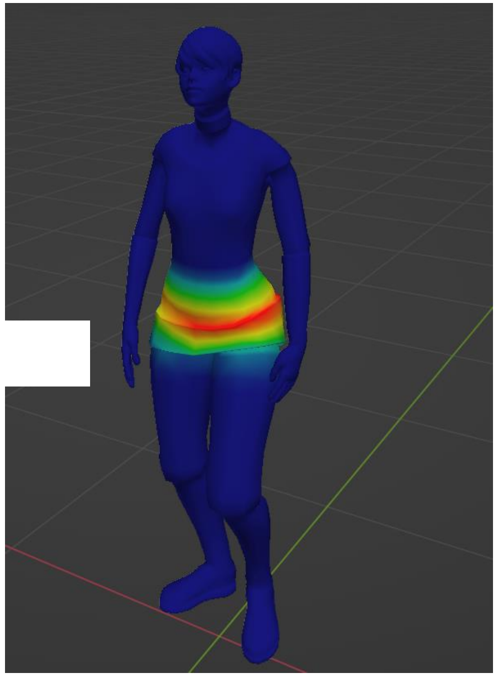
BONES



$$T = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

B = Armature's total **bones**

WEIGHTS

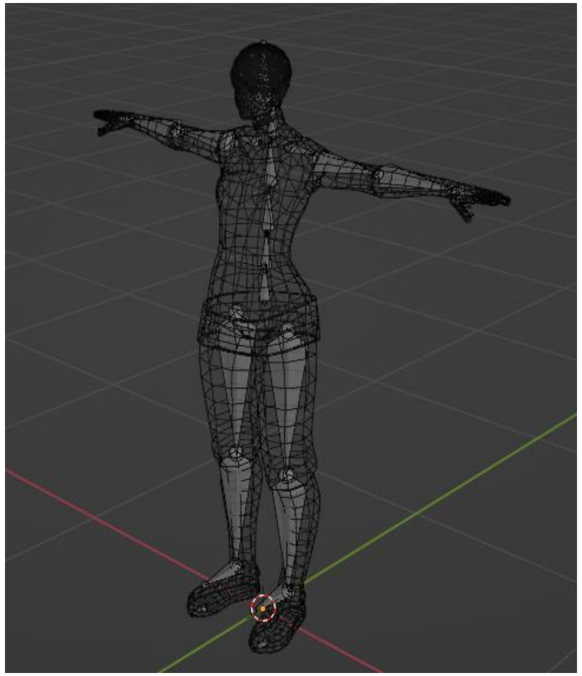


$$\sum_{b=1}^{B} \boldsymbol{w_{ib}} = \mathbf{1}$$

$\forall i \in \{1,\ldots,N\}: w_i \geq 0$

MaxBones = max limit of influence weights

REST POSE



Rest Pose = (R)
Example Pose = (P)

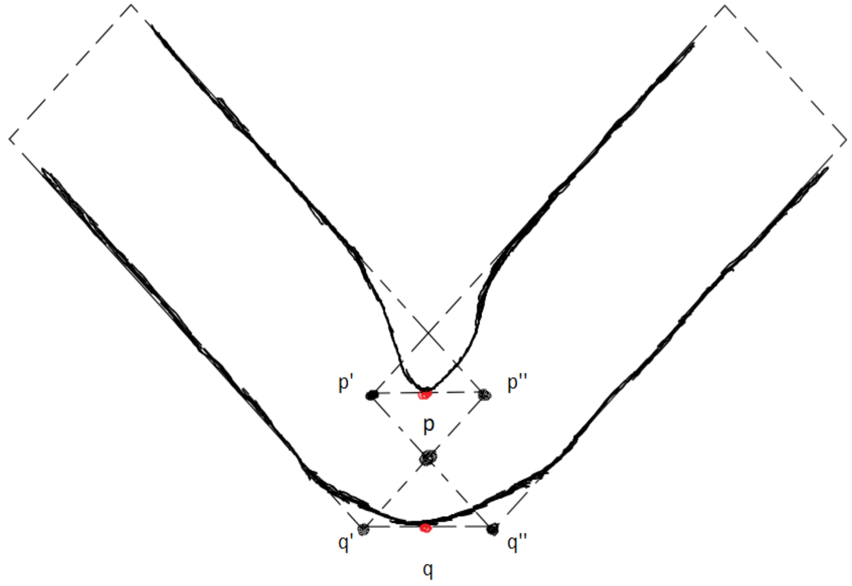
P = total character animation **frames**

LINEAR BLEND SKINNING

- Virtual model of a character with **N vertices, B bones and w_{ib} influence weights** and animation in **P frames** with **rest pose** R.
- To approximate the vector that translates the vertice **v'^p_i** at frame **p∈{1,…,P},** according to the **Linear Blend Skinning** Method we calculate the following equation:

$$v'^{p}_{i} = \left(\sum_{b=1}^{B} w_{ib} T_b^p \right) v_i^R$$

- The equation determines a **linear space,** in which the approximation of a vertice can slide, according to the influence weights that it has.



LINEAR APPROXIMATION METHOD

GENERAL IDEA

- We approach solutions for the minimization function with the **least squares method**. The uknown variable we approximate are the influence weights of the vertices and the matrices of the bones.
- **Repeat consecutevily** approaching better solutions each time for the weights and the bones.

MINIMIZATION FUNCTION

- The minimization function is as follows:

$$f = \sum_{i=1}^{N} \| v'^{p}_{i} - v_i^{p} \|^2$$

- In other words we **minimize the approximation error** of the vertices to the real data of the mesh.

The minimization function can be simplified to :

$$\|Ax - b\|^2$$

- The **matrix x** contains the bones and weights.
- **Matrix A** contains the rest elements fo the **LBS** equation for the vertice approximation.
- **Matrix B** contains the real vertices.

NON LINEAR APPROXIMATION METHOD

GENERAL IDEA

- We used the algorithm of **Limited Memory – BFGS (L-BFGS)** of the **quasi-Newton optimization** techniques, to develop a **non linear method,** where we minimize an objective function through iterations:

$$f = \sum_{p=1}^{P} \sum_{i=1}^{N} \mathrm{eucD}istance\left(v_i'^{\,p}, \mathrm{v}_\mathrm{i}^\mathrm{p}\right) + \sum_{i=1}^{N} \lambda * \left[\left[\sum_{b=1}^{B} w_{ib}\right] - 1\right]^2$$

Charles George **Broyden**, Roger **Fletcher**, Donald **Goldfarb** & David **Shanno**

TERMINATION CONDITION

As **termination condition** we define a value **f**$_{tol}$, when the following inequality is valid the iterations stop:

$$\frac{f^k - f^{\{k+1\}}}{\max\{|f^k|, |f^{\{k+1\}}|, 1\}} \le f_{tol}$$

We tried a large range of values for the **f**$_{tol}$ for each modelto determine the convergence of the algorithm

MODELS AND METRICS

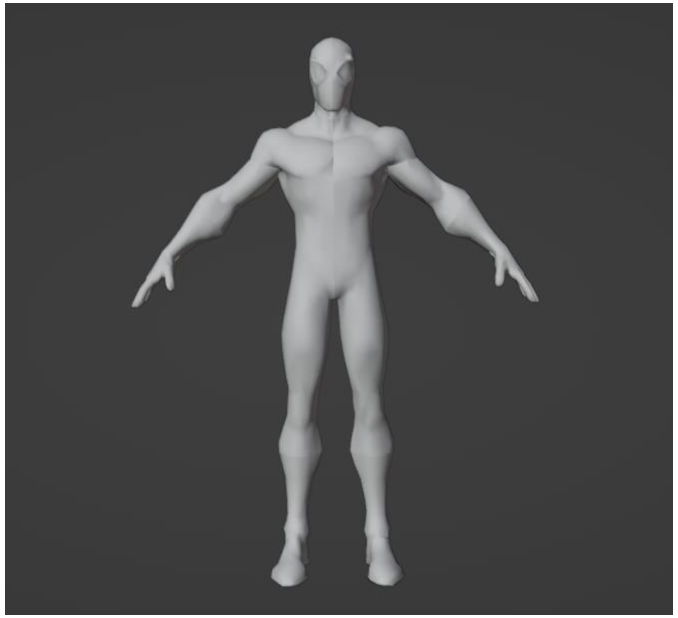
EXAMPLE MODELS

LOLA

5006 vertices, 52 bones, MaxBones = 4, **137 animation frames**.

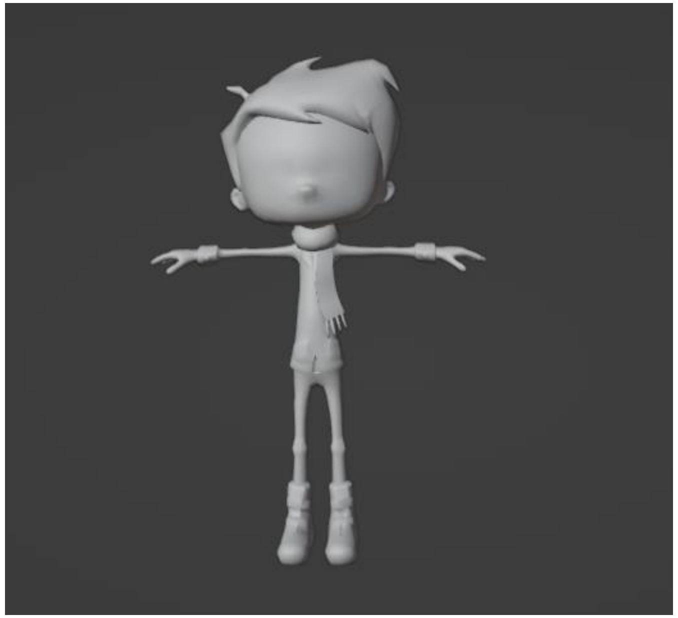
SPIDERMAN

2956 vertices, **78 bones, MaxBones = 8**, 28 animation frames.



TIMMY

2932 vertices, **63 bones, MaxBones = 10**, 24 animation frames.



RESULTS ERROR EVALUATION METRICS

ROOT MEAN SQUARE ERROR

The RMSE or ERMS metric shows the **mean deformation** of the vertices we approached compared to the real vetrices of the model's mesh.

$$ERMS = 100 * \frac{\| V_{orig} \_ V_{approx} \|_F}{\sqrt{3NP}}$$

DISTORTION PERCENTAGE

The Distortion Percentage metric (DisPer), represents the **deformation percentage** of the vertices.

$$DisPer = 100 * \frac{\| V_{orig} \_ V_{approx} \|_F}{\| V_{orig} \_ V_{avg} \|_F}$$

MAXIMUM AVERAGE DISTANCE

The Maximum Average Distance metric (MaxAvgDist) represents the **mean of the maximum approximation error** of a vertice in all the animation frames.

$$MaxAvgDist = \frac{1}{P}\sum_{p=1}^{P} \max_{i=1,\dots,N} \| v_o^{p,i} - v_a^{p,i} \|_F$$

Matrices Shape: V_{orig}, V_{approx}, V_{avg} = (3NP, 1)

LINEAR METHOD

LINEAR METHOD ALGORITHM $\left(\|Ax - b\|^2 \right)$

Elements approximation of the bone translation matrix:

$$x = \begin{bmatrix} a_{11}^{1} \\ a_{12}^{1} \\ \vdots \\ a_{34}^{1} \\ \vdots \\ a_{11}^{B} \\ \vdots \\ a_{34}^{B} \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} a \\ a \\ \vdots \\ a \end{matrix}}\right\} \text{b = 1} \\ \vdots \\ \left.\vphantom{\begin{matrix} a \\ \vdots \\ a \end{matrix}}\right\} \text{b = B} \end{matrix} \qquad b = \begin{bmatrix} v_{1x}^{P} \\ v_{1y}^{P} \\ v_{1z}^{P} \\ \vdots \\ v_{Nx}^{P} \\ v_{Ny}^{P} \\ v_{Nz}^{P} \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} a \\ a \\ a \end{matrix}}\right\} \text{v = 1} \\ \vdots \\ \left.\vphantom{\begin{matrix} a \\ a \\ a \end{matrix}}\right\} \text{v = N} \end{matrix}$$

, p∈{1,…,**P**}

$$A[:3] = \left.\begin{bmatrix} w_{11}v_{1x}^{R} & w_{11}v_{1y}^{R} & w_{11}v_{1z}^{R} & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \\ w_{11}v_{1x}^{R} & w_{11}v_{1y}^{R} & w_{11}v_{1z}^{R} & 1 \\ 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ w_{11}v_{1x}^{R} & w_{11}v_{1y}^{R} & w_{11}v_{1z}^{R} & 1 \\ & \cdots & & \end{bmatrix}\right\} \text{b = 1}$$

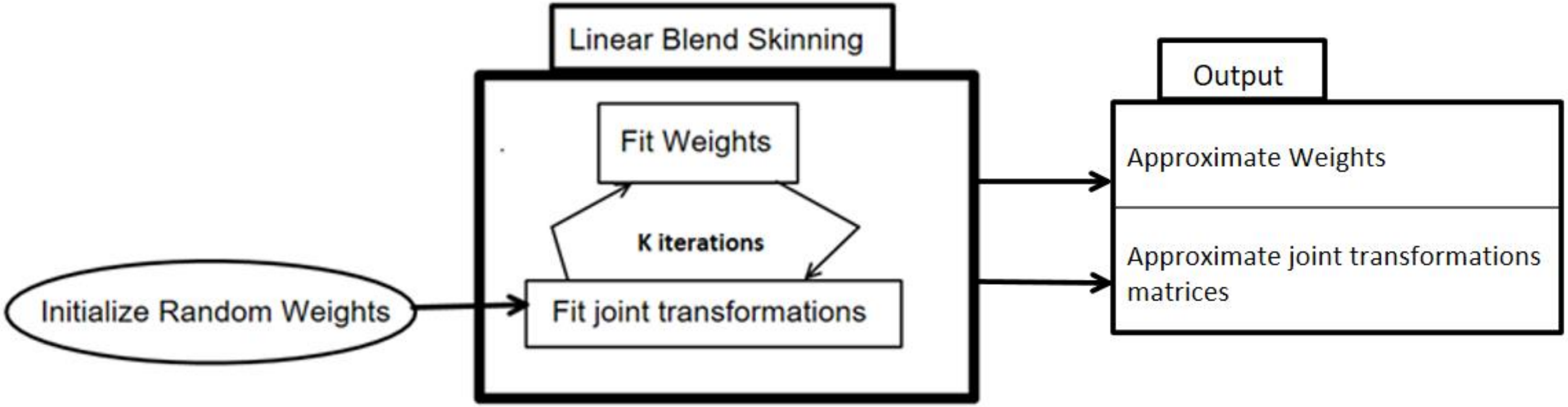
Influence weights approximation:

$$x = \left.\begin{bmatrix} w_{1j} \\ w_{1k} \\ w_{1l} \\ w_{1m} \\ \vdots \end{bmatrix}\right\} \text{v = 1..} \qquad b = \begin{bmatrix} v_{ix}^{1} \\ v_{iy}^{1} \\ v_{iz}^{1} \\ \vdots \\ v_{ix}^{P} \\ v_{iy}^{P} \\ v_{iz}^{P} \\ 1 \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} a \\ a \\ a \end{matrix}}\right\} \text{p = 1} \\ \vdots \\ \left.\vphantom{\begin{matrix} a \\ a \\ a \end{matrix}}\right\} \text{p = P} \\ \end{matrix}$$

, i∈{1,…,**N**}

$$A = \begin{bmatrix} a_1^{i1}v_1^R & a_1^{k1}v_1^R & a_1^{l1}v_1^R & a_1^{m1}v_1^R \\ a_2^{i1}v_1^R & a_2^{k1}v_1^R & a_2^{l1}v_1^R & a_2^{i1}v_1^R \\ a_3^{i1}v_1^R & a_3^{k1}v_1^R & a_3^{l1}v_1^R & a_3^{i1}v_1^R \\ \vdots & \vdots & \vdots & \vdots \\ a_3^{iP}v_1^R & a_3^{kP}v_1^R & a_3^{lP}v_1^R & a_3^{iP}v_1^R \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} a \\ a \\ a \end{matrix}}\right\} \text{p = 1} \\ \cdots \\ \left.\vphantom{\begin{matrix} a \\ a \end{matrix}}\right\} \text{p = P} \\ \end{matrix}$$

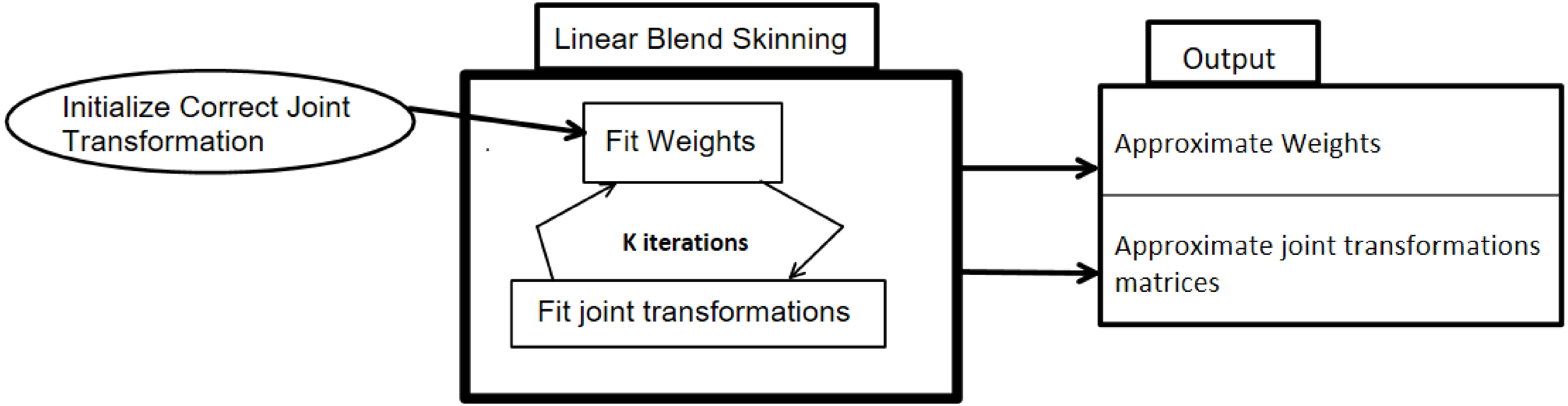
LINEAR METHOD RESULTS



1. RANDOM WEIGHT INITIALIZATION

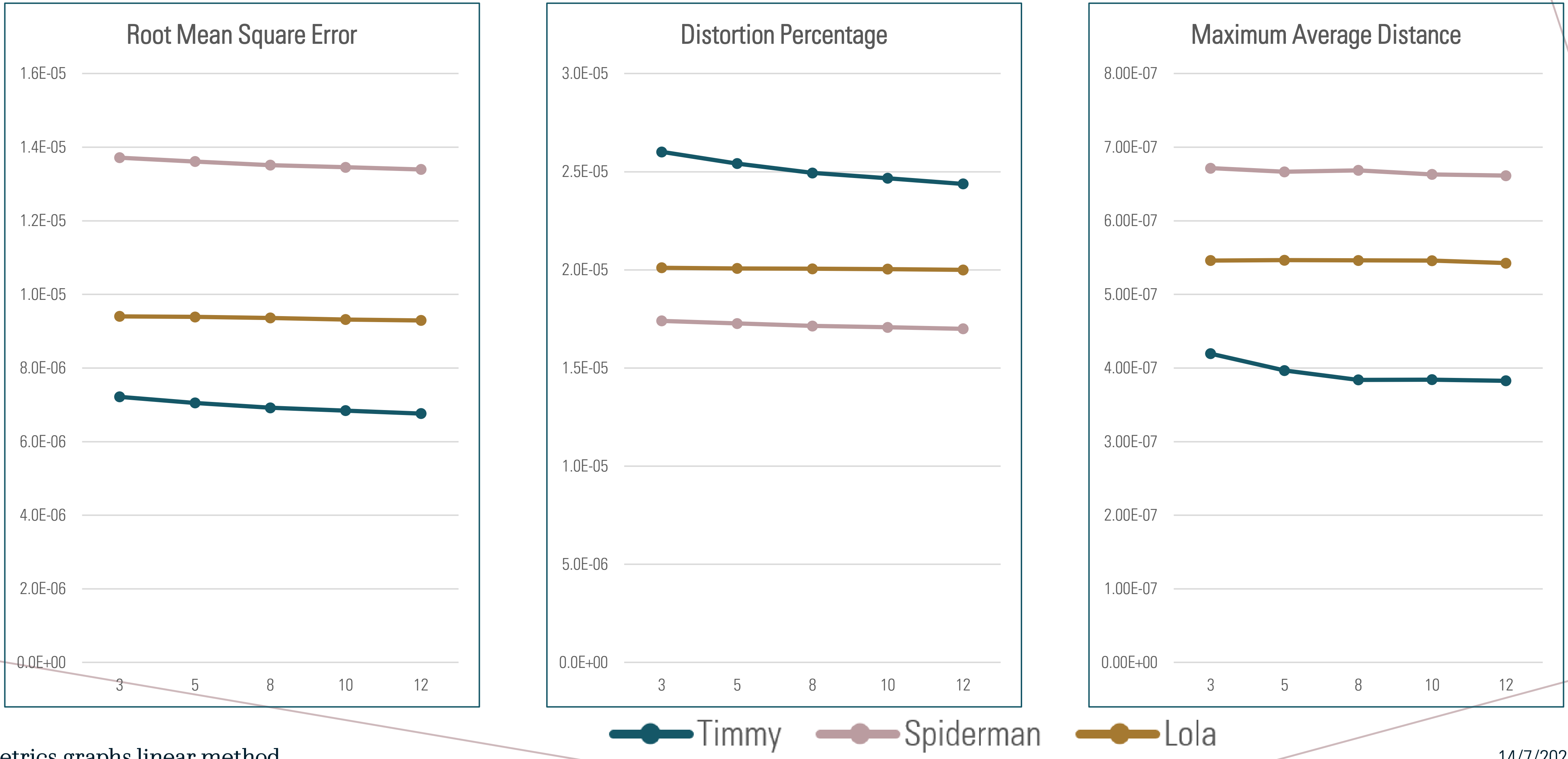
ERROR EVALUATION METRICS OF THE LINEAR METHOD WITH RANDOM WEIGHTS

Error metrics graphs linear method

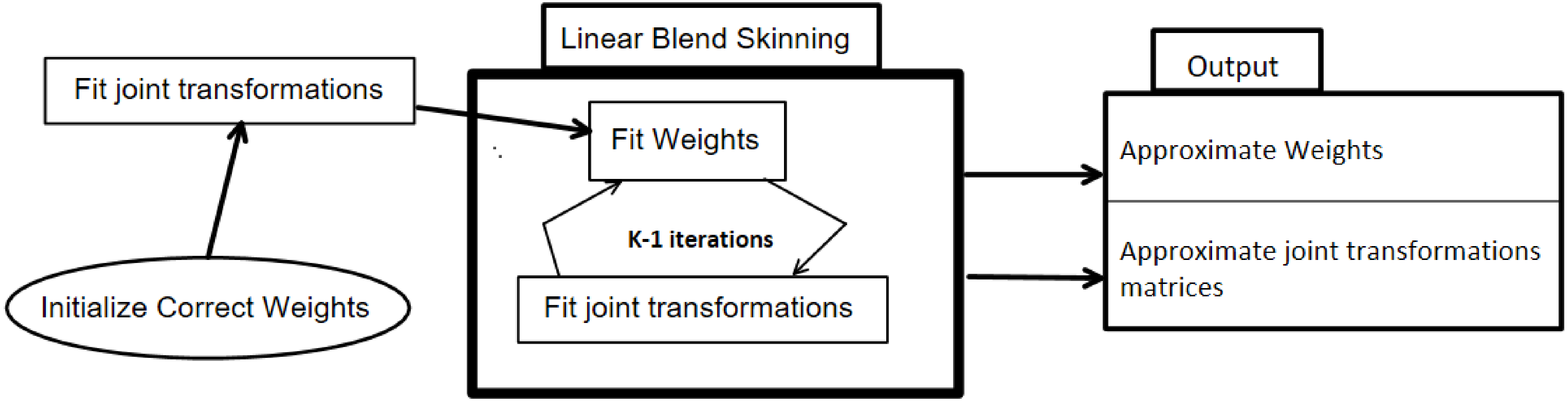


2. REAL BONE DATA INITILIZATION

ERROR EVALUATION METRICS OF THE LINEAR METHOD WITH REAL BONE DATA

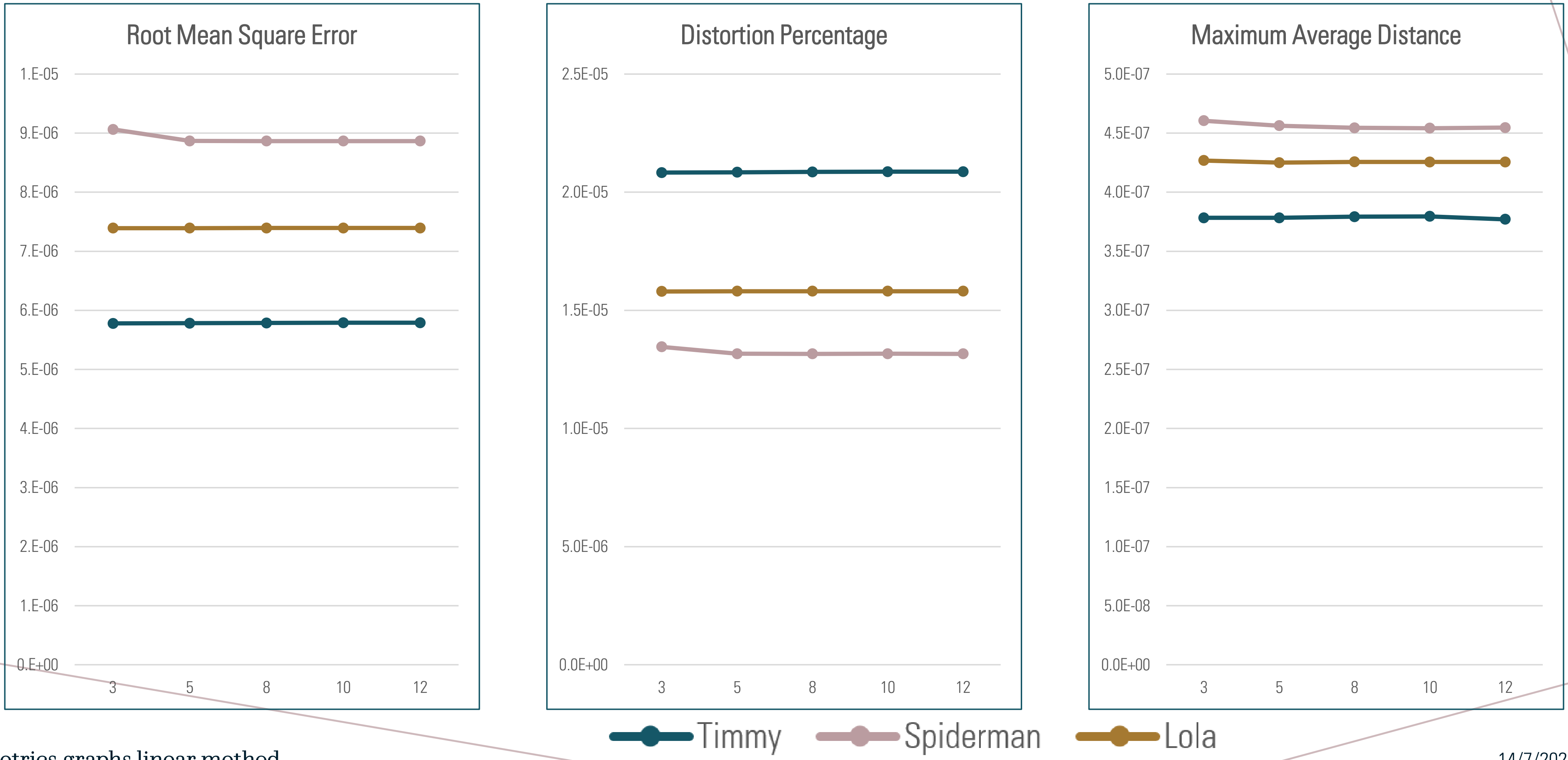


Error metrics graphs linear method



3. REAL WEIGHT INITILIZATION

ERROR EVALUATION METRICS OF THE LINEAR METHOD WITH REAL WEIGHT INITILIZATION



NON LINEAR METHOD

NON LINEAR METHOD *(LBFGS)*

INITIAL POSITION

- **LBFGS method** takes as argument the matrix with the **initial values** of the variables that the objective function contains:

$$x_o = \left[\begin{array}{c} w_{ij} \\ \dots \\ w_{kl} \\ \hline a_{11}^{11} \\ a_{12}^{11} \\ \dots \\ a_{11}^{PB} \\ \dots \\ a_{34}^{PB} \end{array}\right]$$

Βάρη κορυφών με περισσότερα από 1 βάρη

b=1, p=1

b=B, p=P

OBJECTIVE FUNCTION

- The objective function is the result of two sums:

1. **Total of Euclidean distance** in each frame **of all the vertices** we approached from the real data.
2. **Total of influence weights** for each vertex that has more than 1 weights, minus 1 and all of it squared.

TERMINATION

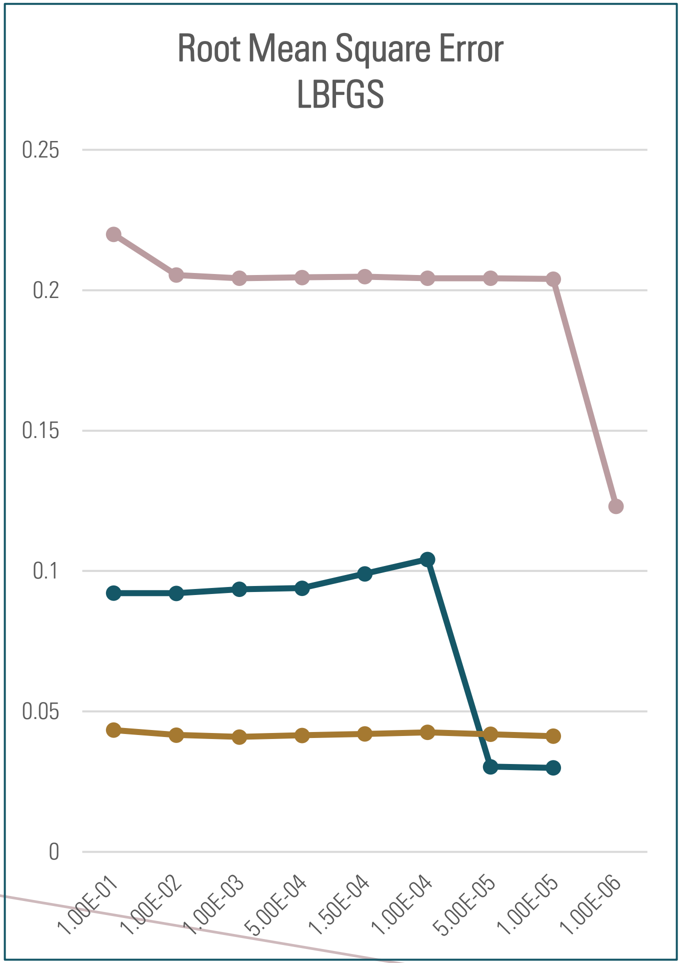
- The f_{tol} value defines the time of termination for the iterations of the algorigthm
- The smaller the f_{tol} value, the more precise the algorithm will be at the cost of more iterations, hence much slower results.

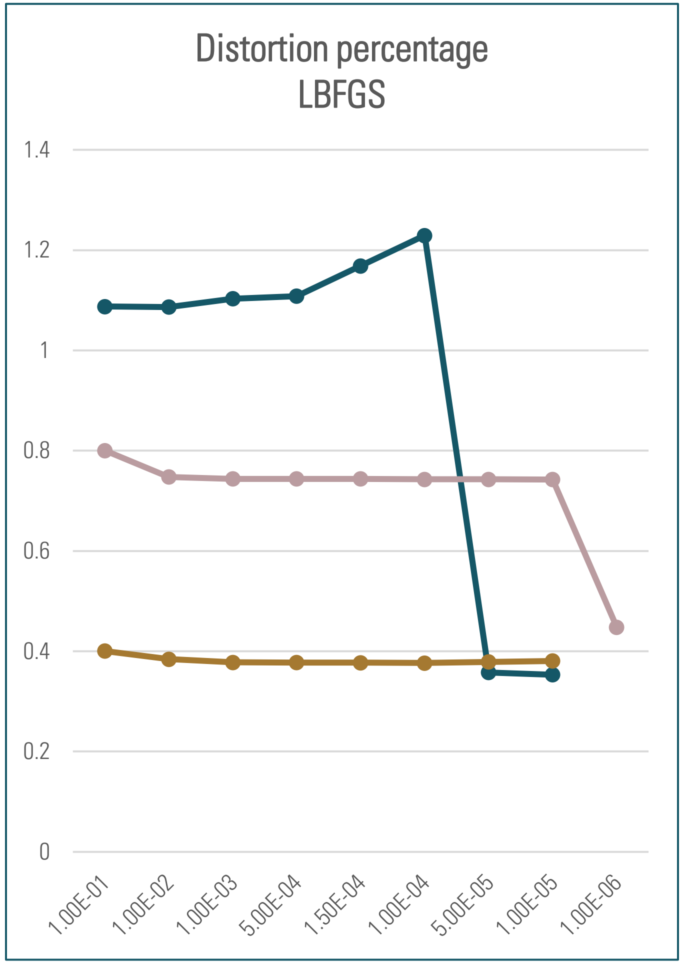
$$\frac{f^k - f^{\{k+1\}}}{\max\{|f^k|, |f^{\{k+1\}}|, 1\}} \le f_{tol}$$

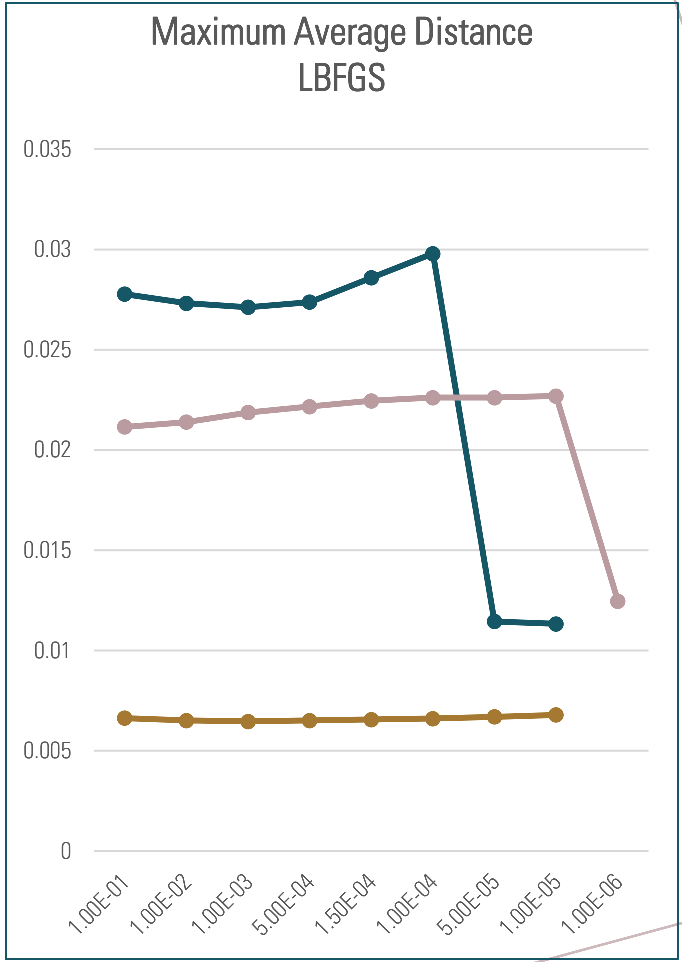
$$f = \sum_{p=1}^{P} \sum_{i=1}^{N} \mathrm{eucDistance}\left(v_i'^{\,p}, \mathrm{v}_i^{\mathrm{p}}\right) + \sum_{i=1}^{N} \lambda * \left[\left[\sum_{b=1}^{B} w_{ib}\right] - 1\right]^2$$

NON LINEAR METHOD RESULTS

ERROR EVALUATION METRICS OF THE NON LINEAR METHOD (LBFGS)







METHOD PERFOMANCE COMPARISON

LINEAR AND NON LINEAR METHOD PERFOMANCE COMPARISON

CONCLUSIONS

CONCLUSIONS

The **non Linear Method** approaches **more appropriate values** for the weights and the bones of a virtual model than the linear method. However there is a great cost in terms of algorithm **complexity.**

The **Linear Method** produces faster and accurate approaches. Running the algorithm for a big amount of iterations does not really improve the approaches, as it happens with the **Non linear Method.**

Required Bandwidth comparison for the reproduction of the character animation:

1. For the reproduction for the animation from the **mesh data** of a model we need:

$$\text{bandwidth1} = {}^{\mathbf{3N*64}\boldsymbol{bit}}\!/_{\mathbf{1}\boldsymbol{s}/\mathbf{24}}\ \text{(bit/s)}$$

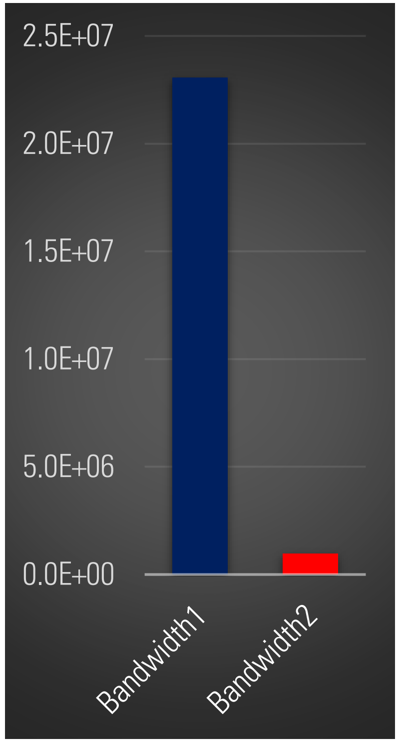
2. For the reproduction with any of the **methods we developed** we only need:

$$\text{Bandwidth2} = {}^{\mathbf{12B*64}\boldsymbol{bit}}\!/_{\mathbf{1}\boldsymbol{s}/\mathbf{24}}\ \text{(bit/s)}$$

E.G: For the model Lola we have N=5006, B = 52:

Bandwidth1 = 23,067,648 bit/s and

Bandwidth2 = 958,464 bit/s



FUTURE EXPANSIONS

- Future expansion of this thesis can include the creation of the **derivative function** for faster calculations in the **Non Linear Method.**
- Adding parallel execution of the code on the graphics card, to minimize the execution time.

BIBLIOGRAPHY

BILBIOGRAPHY

1. Georgios Antwnopoulos (November 2010) “Fast realistic skinning for animating highly deformable objects”, pp 31-39, https://www.cs.uoi.gr/wp-content/uploads/publications/MT-2010-23.pdf
2. Anastasia Moutafidou, Vasileios Toulatzis, Ioannis Fudos, arXiv:2109.07249 [cs.GR] Temporal Parameter-free Deep Skinning of Animated Meshes (15 Sep 2021)
3. Ladislav Kavan, Peter-Pike Sloan, Carol O'Sullivan. Fast and Efficient Skinning of Animated Meshes. Computer Graphics Forum 29(2) [Proceedings of Eurographics], Volume 29 (2010), Number 2
4. LEWIS, J. P., CORDNER, M., AND FONG, N. 2000. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In Proceedings of ACM SIGGRAPH
5. Blender python api https://docs.blender.org/api/current/
6. *Broyden, C. G. (1970), "The convergence of a class of double-rank minimization algorithms", Journal of the Institute of Mathematics and Its Applications, **6**: 76–90*
7. *Fletcher, R. (1970), "A New Approach to Variable Metric Algorithms", Computer Journal, **13** (3): 317–322, doi:10.1093/comjnl/13.3.317*
8. *Goldfarb, D. (1970), "A Family of Variable Metric Updates Derived by Variational Means", Mathematics of Computation, **24** (109): 23–26, doi:10.1090/S0025-5718-1970-0258249-6*
9. *Shanno, David F. (July 1970), "Conditioning of quasi-Newton methods for function minimization", Mathematics of Computation, **24** (111): 647–656*
10. https://github.com/a-dimokas/MethodsForRealisticCharacterAnimation

THANK YOU

Dimokas Angelos

Ioannis Fudos