AUTOMATIC SKINNING OF THE SIMULATED MANIPULATOR ROBOT ARM

Salih Rashid Majeed and Klaus D.Kuhnert
Real-time learning system institute, Siegen University, Germany

ABSTRACT

this paper study the skinning of the manipulator robot arm (JACO) and how to implement this into arm skeleton with easy way and see the effect of the distributed specified weight and how to control it to get the best motion results.

KEYWORDS

Skinning, blender, vertex, weight distribution.

1. INTRODUCTION:

The simulations programs are used in many fields specially the robot fields. Our paper deals with one of the most using manipulator robot arm into human application such as eating or drinking for handicapped people. After simulate the arm and add bones as actuator. We have to make sure that the skin of the arm moves according to the bones movement without any deformation this called skinning. We have also studied the influence of the distributed weight and how to specify the required weight which is corresponding to the effort.

2. PREVIOUS WORK:

A fully-automatic method to extract a kinematic skeleton, joint motion parameters, and surface skinning weights from an animation of a single triangle mesh the result is a compact representation of the original input that can be easily rendered and modified in standard skeleton-based animation tools without having to modify them in the slightest this way we are able to preserve the great modeling flexibility of purely mesh-based approaches while making the resulting skeleton-less animations straightforwardly available to the animator's repertoire of processing tools our results show that the efficient combination of skeleton learning and temporally-coherent blending weight computation enables us to effectively bridge the gap between the mesh-based and skeleton-based animation paradigms[1]. extend approaches for skinning characters to the general setting of skinning deformable mesh animations we provide an automatic algorithm for generating progressive skinning approximations that is particularly efficient for pseudo-articulated motions, Our contributions include the use of nonparametric mean shift clustering of high-dimensional mesh rotation Sequences to automatically identify statistically relevant bones, and robust least squares methods to determine Bone transformations, bone-vertex influence sets, and vertex weight values [2]**.**Object deformation with linear blending dominates practical use as the fastest approach for transforming raster images, vector graphics, geometric models and animated characters, Unfortunately, linear blending schemes for skeletons or cages are not always easy to use because they may require manual weight painting or modeling closed polyhedral envelopes around objects, our goal is to make the design and control of deformations simpler by allowing the user to work freely with the most convenient combination

DOI : 10.5121/ijcga.2016.6101

of handle types [3].A two-layer sparse compression approach to effectively compress the weights of the widely used linear blend skinning model, by employing virtual bones to cache transformation blending of master bones, our approach can significantly reduce computational cost, with insignificant loss of the accuracy of the original skinning model, The virtual bones allow users to optimally distribute and control the approximation errors on different regions of 3D models, while keeping our compressed skinning model friendly to parallel implementation on multi-core processors[4].A process called multi-weight enveloping for deforming the skin geometry of the body of a digital creature around its skeleton. It is based on a deformation equation whose coefficients we compute using a statistical fit to an input training exercise, In this input, the skeleton and the skin move together, by arbitrary external means, through a range of motion representative of what the creature is expected to achieve in practice [5]. A new algorithm to compute surface normal with minimal overhead both in terms of the memory footprint and the required per-vertex operations, by accounting for the variation of the skinning weights over the surface, we achieve a higher visual quality compared to the standard approximation ubiquitously used in video-game engines and other real-time applications, Their method supports Linear Blend Skinning and Dual Quaternion Skinning[6].A simple and fast skinning algorithm for skeleton-driven deformations of articulated characters. During the animation, the deformation model preserves the volume and allows for passive jiggling behavior, their system initializes the blend weights and the soft constraints automatically, so it does not require a considerable set-up effort, Artists can define specific areas of the body and decide on the amount of jiggling affecting them by tuning a single scalar stiffness parameter [7]

3. THEORETICAL WORK

The skinning as general definition is how to overcome and control the deformation which is as result for bones movement i.e. when we want to move the skin according to the bone movement the is some mismatched between the two movement which is caused the deformation. The work deals with the robot arm as the human arm and distribute the bones on the arm joints. The arms with the bones are like the complete moving system one complete the other in motion. The bones should move accurately and exactly like the real human joints when moved with the skin of the arm so we used BLENDER as program for drawing the arm and add the bones. The are many algorithms for the skinning but the most famous is linear skinning. All the details about this algorithm explain briefly in this work. The distribution of the weight in BLENDER follows this algorithm. We study here the effect when applying the automatic skinning (automatic weight distributed on the bones) and the exactly weight distribution according to the arm motion effect. The weight distribution effect represented as the color percentage so the color ratio from dark blue to dark red (from the zero weight effect to the maximum weight).

We can also explain briefly about the programs that have been used:

- Blender: is the free and open source 3D creation suite, it supports the entirety of the 3D pipeline modeling, rigging, animation, simulation, rendering, compositing and motion tracking, even video editing and game creation, advanced users employ Blender's API for Python scripting to customize the application and write specialized tools; often these are included in Blender's future releases.[8]

The JACO manipulator robot arm has been programmed and simulated in this paper, JACO moves smoothly and silently around 6 degrees of freedom with unlimited rotation on each axis, the axes are aluminum compact actuator discs (CADs) of a unique design, each JACO robot arm consists of 2 distinct sets of 3 identical, interchangeable, and easy-to-replace CADs linked together by a ZIF (zero insertion force) cable, its main structure, entirely made of carbon fiber, delivers optimal robustness and durability as well as a cutting-edge look-and-feel, the arm is mounted on a standard aluminum extruded support structure that can be affixed to almost any surface, the gripper consists of 3 under actuated fingers that can be individually controlled, their unique bi-injected plastic structure (patent pending) endows them with great flexibility and unrivalled grip, JACO technology allows the fingers to adjust to any object whatever its shape; as a result, they can gently pick up an egg or firmly grasp a jar [9]

Enveloping (or skinning) is a common and fundamental task in computer graphics, Whenever an animator controls a character via a skeleton, enveloping is used to map these controls to deform a mesh surface, there is no standard way to envelope, an artist may run a physical simulation to model muscle deformations or tune complex systems of shape blending for more direct control. For interactive applications, linear blend skinning enjoys the most popularity, it is easy to use and can be accelerated on graphics hardware [10]. Linear blend skinning, also known as skeleton-subspace deformation, (single-weight-) enveloping, or matrix-palette skinning, is the basic and most well-known algorithm for direct skeletal shape deformation [11**].** The traditional interactive skinning model goes by many names[12] Call it Skeleton Subspace Deformation or SSD, Maya calls it “smooth skinning” and we call it linear blend skinning [13]

Linear skinning assumes the following input data:

• **Rest pose shape**, typically represented as a polygon mesh. The mesh connectivity is assumed to be constant, i.e., only vertex positions will change during deformations. We denote the rest-pose vertices as $\mathbf{v}_1 \ldots \mathbf{v}_n \in \mathbb{R}^3$. It is often convenient to assume that $\mathbf{v}_i$ are in fact $\mathbb{R}^4$ vectors with the last coordinate equal to one, according to the common convention of homogeneous coordinates.

• **Bone transformations**, represented using a list of matrices $\mathbf{T}_1 \ldots \mathbf{T}_m \in \mathbb{R}^{3\times4}$,the matrices $\mathbf{T}_i$ can be conveniently defined using an animation skeleton; in this case they corresponds to spatial transformations aligning the rest pose of bone $i$with its current (animated) pose, bone transformations are typically the only quantity that is allowed to vary during the course of an animation.

• **Skinning weights.** For vertex $\mathbf{v}_i$, we have weights $w_{i,1}, \ldots, w_{i,m} \in \mathbb{R}$. Each weight $w_{i,j}$ describes the *amount of influence* of bone j on vertex i. A common requirement is that $w_{i,j} \geq 0$ and $w_{i,1} + \cdots + w_{i,m}$ = (partition of unity). Linear blend skinning computes deformed vertex positions $\mathbf{v}_i$ *0* according to the following formula:

$$\mathbf{v}_i' = \sum_{j=1}^{m} w_{i,j}\mathbf{T}_j\mathbf{v}_i = \left(\sum_{j=1}^{m} w_{i,j}\mathbf{T}_j\right)\mathbf{v}_i \qquad (1)$$

The latter form highlights the fact that the rest pose vertex $\mathbf{v}_i$ is transformed by a linear combination (blend) of bone transformation matrices $\mathbf{T}_j$. These matrices are the deformation primitivesof linear blend skinning, i.e., elementary building blocks of deformations. While arbitrary affine transformations are allowed, sometimes it is convenient to assume that $\mathbf{T}_j$ are rigid body transformations, i.e.,$\mathbf{T}_j \in SE(3)$. Note that many implementations assume that most of the weights $w_{i,1}, \ldots, w_{i,m}$ are zero. Due to graphics hardware considerations, it is common to assume there are at most four non-zero weights for every vertex; different limits can be found in some systems.[11]

4. EXPERIMENTAL WORK AND RESULTS

The challenge in this work is implementing the movement of the manipulator robot arm (JACO) which is as example. The arm has many joints each one of these joints must rotate about specific axis after drawing the arm with all the material specifications and vertex to be like the real arm reality. The bones have been added to the skin of the robotic arm. The bones here are the actuators which will be moved the skin accordantly.

The basic idea is transfer the arm parts softly form place to another without deformation in the arm joints and links. The first step in blender skinning is the weight distribution: to move the skin we need to distribute the effect of each bone on the right skin part. Each bone responsible for moving a specific arm part. As shown in figure (1)

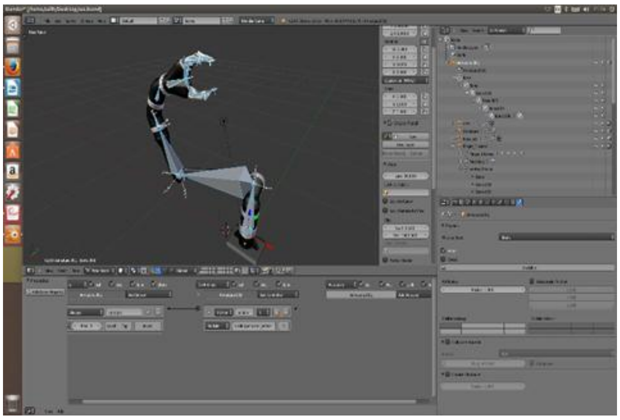

Figure1.The JACO arm with the bones.

4.1. Weight distribution:

The main step of skinning is how to apply the suitable weight to the vertex there are two ways to do that in Blender:

1. Applying constant value of weight to the whole vertex and then edit the vertex position when the deformation will happen.
2. Applying the suitable and specific weight to each vertex this allow to vertex to move flexibly according to the required applied weight.

The second way is more flexibility and reliability than the first because the arm has many joints and vertex and each group of vertex moved in the specific axis. The first way suitable for the animation that hasn't many vertex groups and also in the second way the deformation may happen or may not because of the good weight distribution way as shown in figure (2)

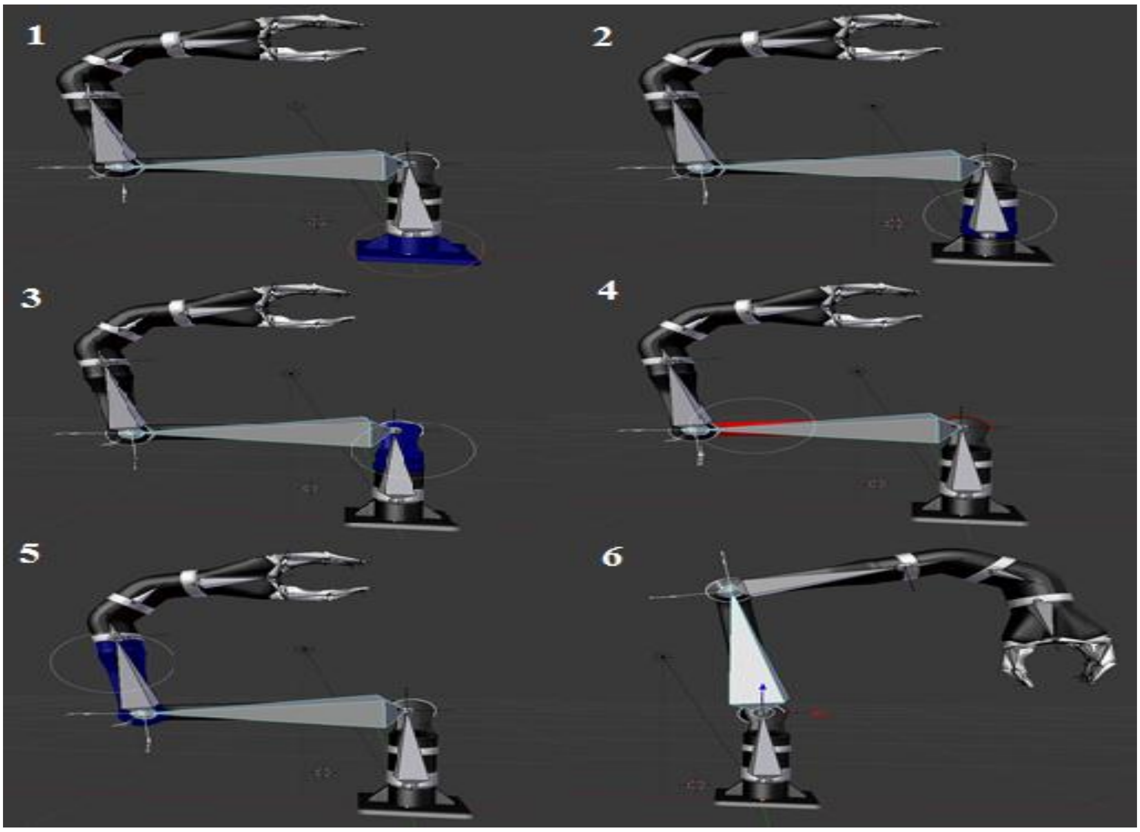

Figure.2 applying the second way in weight distribution

We will take the rotation of one link as example and we will apply this link to these two methods for weight distribution. in the second method each one of the vertex groups will be applied to the specific weight which is exactly or near to the required value. As shown here the motion of bone numbers two as example. Obviously the effect of bone number two on the base vertex group in photo (1) and on the link two vertex group in photo(2) and on the link three vertex group in photo(3) and on the link number four vertex group which will move photo (4) and the last one the vertex group of link five .We can see that the link number four moved very softly and without any deformation.

Applying the first way. When the deformation happened in one vertex group which moved in specified axis we need to edit the vertex position in this axis for overcoming the deformation but this has side effect on the other vertex groups and there is other deformation will appear in the other vertex groups. as has been described in figure (2) we can see in the figure (3) the effect of bone number two on each one of the vertex groups links but here when applied “automatic weight distribution” and the deformation will happened clearly because of the non-effective distribution of the weight along the vertexes. in the figure (3) the effect of bone number two on the base vertex group in photo (1) and on the link two vertex group in photo (2) and on the link three vertex group in photo (3) and on the link number four vertex group which will move photo (4) and the last one the vertex group of link five.

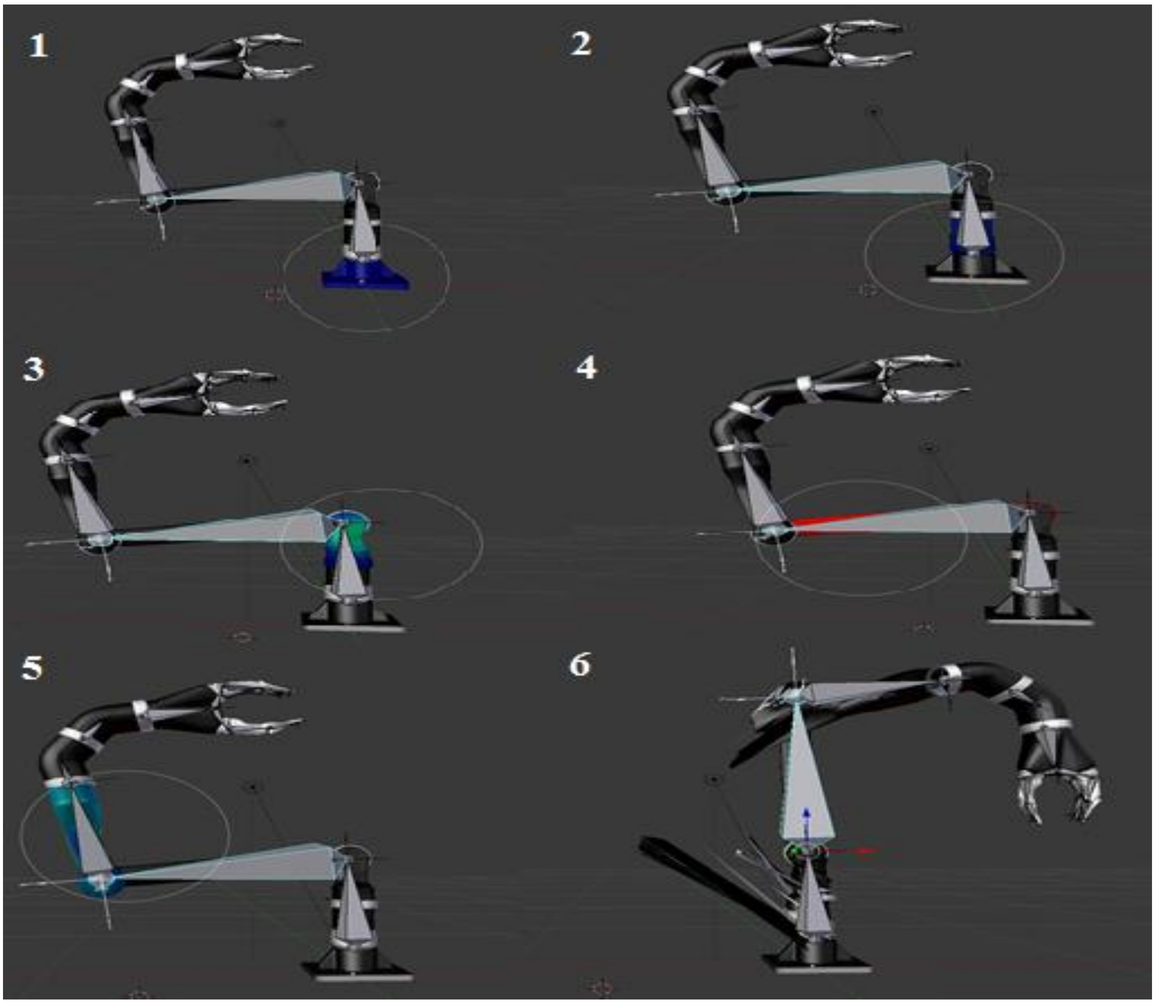

Figure.3 applying the first way in the weight distribution.

4.2. The rotational axis

The axis of the bone rotation must be specified. The wrong bones axis can be caused absolutely many deformation in the skin of robot arm the rotation axis must be known for each joints to make the right decision when choosing the rotational axis as shown in figure(4). In BLENDER many rotational axis can be shown but the problem is how to choose the right axis which the rotation is about it.

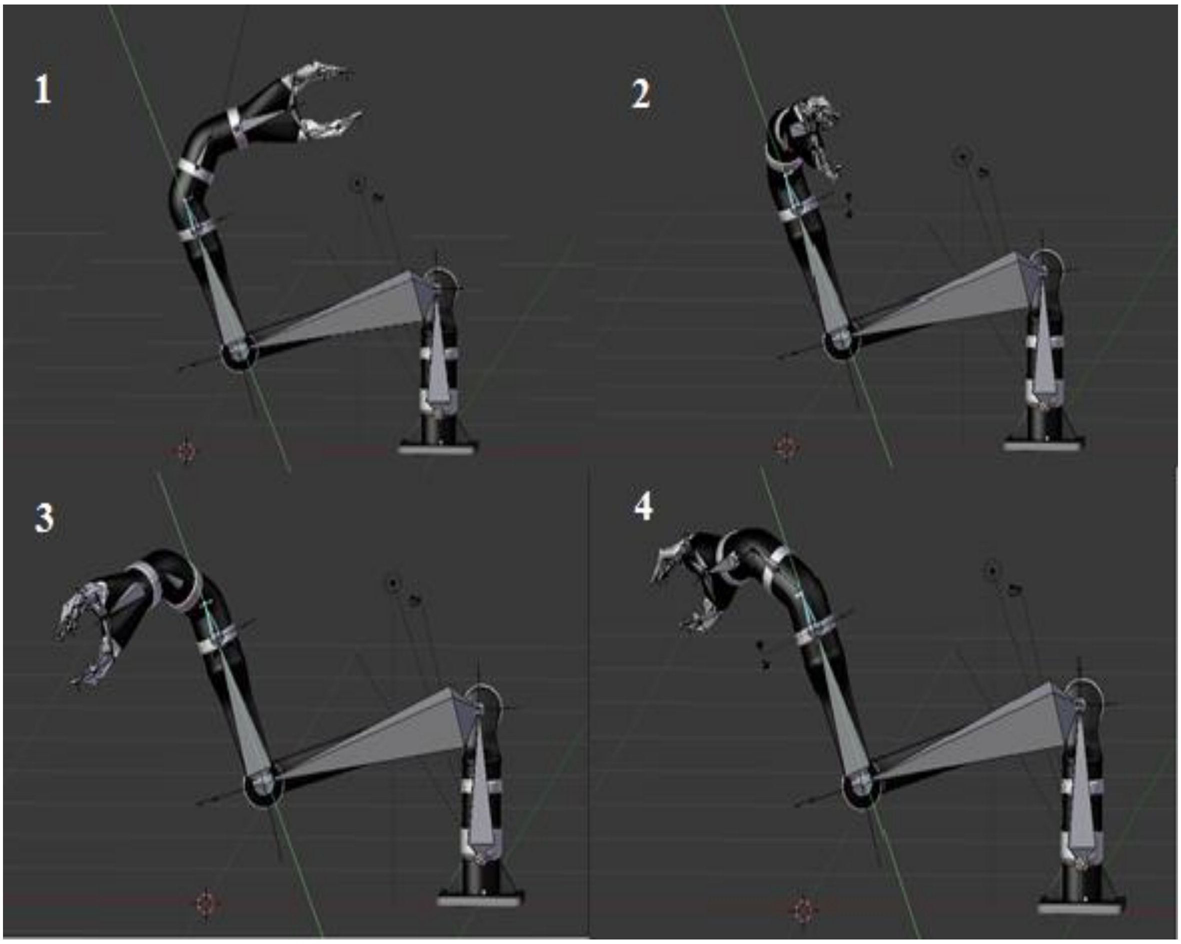

Figure.4 the arm rotate softly with the right specified axis.

We can see here an example figure (5) for one joint which is surrounded by the red circle in the left side we can see the zoom joint and the deformation which happened clearly in figure (5). The reasons of this were discussed with the details in [14]. In short, by splitting a rigid transformation into a rotation and translation pair, we are committing to a specific pivot point (center of rotation), around which the rotations will be interpolated, by default, this center of rotation corresponds to the origin of material-space coordinates, which is typically located near the object's center of mass this explains the unusual result [11].

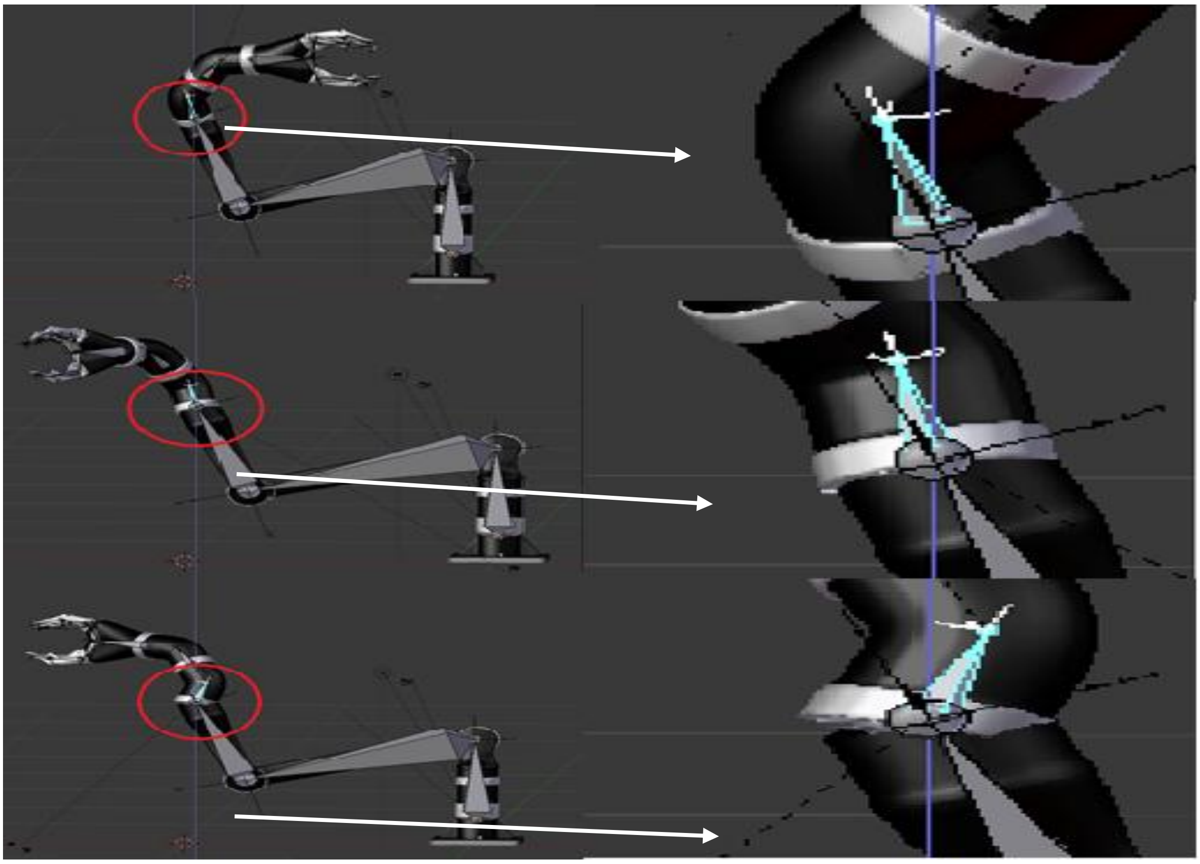

Figure.5 the deformation when moving the joint in wrong axis.

5. CONCLUSION

In this paper we presented very easy and effective way for skinning animation especially in the robot arm field using BLENDER. This way is effective solution instead of the classic ways and its calculations and discussed the problems that will happen during the execution of the arm simulation and how to fix it.

REFERENCES

[1] Edilson de Aguiar1, Christian Theobalt, Sebastian Thrun, Hans-Peter Seidel, 2008,''Automatic Conversion of Mesh Animations into Skeleton-based Animations''EUROGRAPHICS 2008 / G. Drettakis and R. Scopigno (Guest Editors), Volume27, Number 1

[2] Theodore Kimy, Doug L. James, 2011‘’Physics-based Character Skinningusing Multi-Domain Subspace Deformations’’, SCA '11 Proceedings of the 2011 ACM SIGGRAPH/Euro graphics Symposium on Computer Animation

[3] Alec Jacobson, Ilya Baran, Jovan Popovi´c, Olga Sorkine, 2011‘’Bounded Biharmonic Weights for Real-Time Deformation’’, ACM SIGGRAPH

[4] Binh Huy Le_ Zhigang Dengy ,2013‘’Two-Layer Sparse Compression of Dense-Weight Blend Skinning’’ACM Transactions on Graphics (TOG) - SIGGRAPH Conference Proceedings, Volume 32 Issue 4, July

[5] Xiaohuan Corina Wang, Cary Phillips, 2002‘’Multi-Weight Enveloping: Least-Squares Approximation Techniques for Skin Animation’’,SCA '02 Proceedings of the ACM SIGGRAPH/Euro graphics symposium on Computer animation

[6] Marco Tarini, Daniele Panozzo, Olga Sorkine-Hornung ,2014‘‘Accurate and Efficient Lighting for Skinned Models’’, The Eurographics Association and Blackwell Publishing Ltd., Volume 33 number 2

[7] Nadine Abu Rummany, MarcoFratarcangeli, 2015 ‘’ Position-based Skinning for Soft Articulated Characters’’ Computer Graphics Forum, Volume 34, Issue 6

[8] http://www.blender.org

[9] Admin on June 1, 2011 “Jaco Robot Arm” Robotic Magazine, Available on http://www.roboticmagazine.com/robot-review/jaco-robot-arm-2

[10] Robert Y. Wang1 Kari Pulli1, Jovan Popovi´c1, 2007‘’Real-Time Enveloping with Rotational Regression’’ ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH, Volume 26 Issue 3, July

[11] Ladislav Kavan, 2014,”Skinning: Real-time Shape DeformationPart I: Direct Skinning Methods and Deformation Primitives”, SIGGRAPH course— Skinning: Real-time Shape Deformation

[13] Alex Mohr, Michael Gleicher, 2003”Building Efficient, Accurate Character Skins from Examples“, University of Wisconsin, Madison, ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH,Volume 22 Issue 3, July

[12] LEWIS, J. P., CORDNER, M., AND FONG, N. 2000. “Pose space deformations: A unified approach to shape interpolation and skeleton-driven deformation”. Proceedingsof ACM SIGGRAPH, Annual Conference Series, ACM SIGGRAPH.

[14] KAVAN L., COLLINS, S., ZARA, J., AND O’SULLIVAN, C. 2008. “Geometric skinning with approximate dual quaternion blending.” ACM Trans. Graph. 27, 4, 105:1–105:23.

Authors:

Salih Rashid Majeed: received his bachelor and master from Baghdad university in mechatronics and robot engineering and now he is PhD student in the institute for real time learning system (Siegen University). Interested researches are robot, artificial intelligence, image processing.

Klaus D.Kuhnert: Received his Dipl.-Ing. In Computer Science from the Technical University of Aachen (Germany), in 1981, and a Ph.D. Degree in Computer Science from the Uni Bw München (Germany) in 1988.Referee for IEEE journals and program committees he also served for several research foundations. He is European Editor of the International Journal of Intelligent Systems Technologies and Applications (IJISTA). Member of the center of sensor-systems NRW and founding member of the IEEE TC robotics learning. He is now the head of real time learning system institute in Siegen University