A NEW TRIANGLE THEOREM TO SOLVE THE INVERSE KINEMATICS PROBLEM FOR CHARACTERS WITH HIGHLY ARTICULATED LIMBS

Enrique A. ROSALES[1] and Luis E. FALCÓN-MORALES[2]
[1]Universidad Panamericana, México
[2]Instituto Tecnológico y de Estudios Superiores de Monterrey, México

ABSTRACT: This paper presents a new geometrical method to solve the inverse kinematics (IK) problem, for characters with highly articulated limbs (HAL). Our method relies on a new triangle theorem, which is the main contribution of this work. This theorem, allows to compute the intersection between two circles, without any sine or cosine computation, and in less time than the classic algebraic aproach. We also present the mathematical proof for this theorem, which is valid for any triangle. Applying this new method, we are able to produce complex shapes as spirals or springs, with stable and smooth animations, in redundant kinematic chains (KC).

Keywords: Character animation, geometry, triangle theorem, inverse kinematics, circle-circle intersection, geometry theorem

1. INTRODUCTION

Despite massive amount of computer animation tools, some types of characters still remain very difficult to animate. HAL like an elephant trunk [14] or an octopus arm are very difficult to animate using traditional IK solvers.

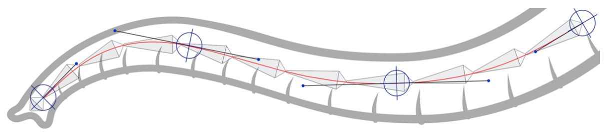

Figure 1: A Bézier curve rig for an elephant trunk

To animate this kind of limbs, the most common solution is to use an IK solver based on Bézier curves [2]. The curve has vertices and handles which are moved by the animator, and each link adopts a tangent direction to the curve as seen in Figure (1).

The purpose of this work was to find a new solution for this problem, and our main goal was to produce a tool to easily animate a spiral shape like Figure (2) while keeping curves ratios [11] and smooth movements.

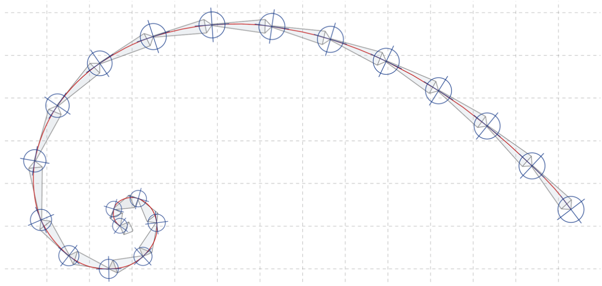

Figure 2: A spiral made by Bézier curves

2. INVERSE KINEMATICS

The IK research in robotics, have its own concerns as inertia, forces, or angular and linear velocities [1, 7, 8]. But this physical concerns can be ignored in animation and computer graphics [10].

So if we work in a virtual world, we only need to calculate the position for each link in the KC. So it is possible to determine our IK problem as equivalent to a *Circle-Circle Intersection Problem* (CCI) [9, 15, 16] for each pair of links, as seen in Figure (3), where the origin denoted as O_1 is the center of the first circle, the end effector denoted as O_2 is the center of the second circle, the distance between the circles denoted as d, the radii of the circles denoted as l_1 and l_2 representing two links, and P_1 and P_2 as the two possible solutions for the intersection.

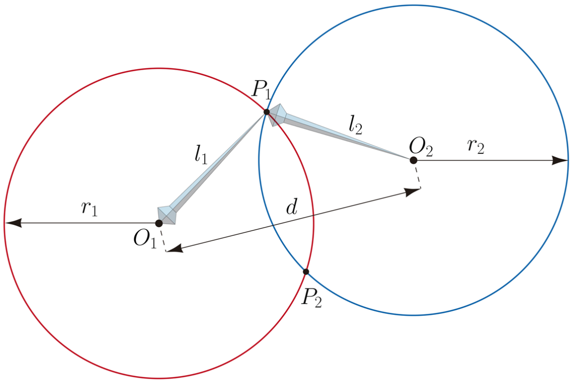

Figure 3: Inverse kinematics as circle-circle intersection

2.1 Singularities

In every KC there are some configurations where the IK problem has no solution, for example: the goal could be unreachable for the end effector, this types of no-solution configurations are known as a singularities.

Given two circles in a two dimension frame of reference, they could have several relative positions and relations of their radii. We defined five possible combinations, as the singularities of our system.

Exterior Singularities

When $d \geq r_1 + r_2$ ee have two possible singularities:

(1) Exterior circles
When $d = \overline{O_1O_2} > r_1 + r_2$, and there are no intersections between the circles.

(2) Tangent exterior circles
When $d = \overline{O_1O_2} = r_1 + r_2$, and there is only one intersection point between the circles.

In this two cases, the KC must adopt a fully extended posture where the orientation of each link is the vector $\overline{O_1O_2}$, so no IK compute is needed.

Interior Singularities

When $d \leq |r_1 - r_2|$ we can have three different singularities:

(3) Interior circles
When $d = \overline{O_1O_2} < |r_1 - r_2|$, and there are no intersections between the circles.

(4) Tangent interior circles
When $d = \overline{O_1O_2} = |r_1 - r_2|$, and there is only one intersection between the circles.

(5) Overlapping equal circles
When $O_1 = O_2$ and $r_1 = r_2$ and there are infinite intersections between the circles.

An IK solution is needed for tangent exterior, or tangent interior circles, in this cases, we determine a safe distance that must be larger than $|r_1 - r_2|$, in which the IK solution will be calculated, this distance is a vector with the same direction than $\overline{O_1O_2}$.

3. PROPOSED METHOD

The first step of our method is to find the coordinates of the perpendicular projection from $\overline{OJ}$ towards $\overline{OP}$, denoted as j as seen in Figure (4), using our new theorem that requires a geometrical construction which is explained as follows:

We draw a circle with a radius equal to l_1 and the center of which is at O, the intersection between this circle and $\overline{OP}$ towards P is denoted by S_1. Then we draw another circle with a radius equal to l_2 with P as origin, the intersection between this new circle and $\overline{PO}$ towards O is denoted by S_2.

Now, we determine m as the middle point between S_1 and S_2.

3.1 The new theorem

Let $\triangle OPJ$ be a triangle like the one in Figure (4), where $A = |\overline{OP}|$, $B = |\overline{Om}|$ and $C = |\overline{S_2S_1}|$, the

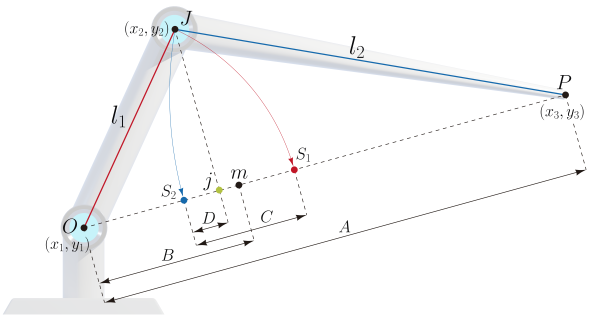

Figure 4: New method construction

distance from S_2 to j (denoted by D) is equal to $\frac{B*C}{A}$, that is:

$$\frac{A}{B} = \frac{C}{D}. \tag{1}$$

The theorem is valid for any configuration of triangles as shown in Figure (5).

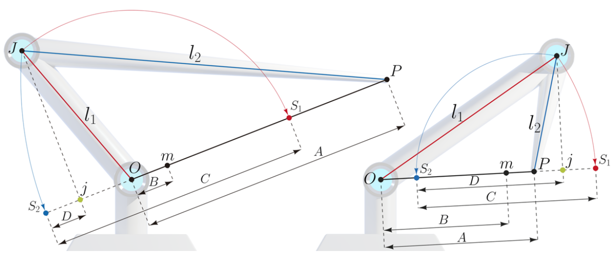

Figure 5: Possible configurations

3.2 Proof

WLOG, we can place the triangle $\triangle OPJ$ on a different direction, matching $\overline{OP}$ with the X positive axis, and O matching the origin of the coordinate system XY. Furthermore, we will use the same notation of the Figure (6) and we will call θ the angle formed by $\overline{OP}$ and $\overline{OJ}$. If $A \geq l_2$, then S_2 is on the positive side of X axis; however, if $A < l_2$, then S_2 is on the negative side of X axis (Figure 6). To simplify the notation, when we wrote $\pm$ will mean that the sign $+$ is for the case of $A \geq l_2$ and the sign $-$ for the case of $A < l_2$; in the same way, when we wrote $\mp$ will mean that the sign $-$ is for the case of $A < l_2$, and the sign $+$ is for the case of $A \geq l_2$.

$$A = |\overline{OP}| \tag{2}$$

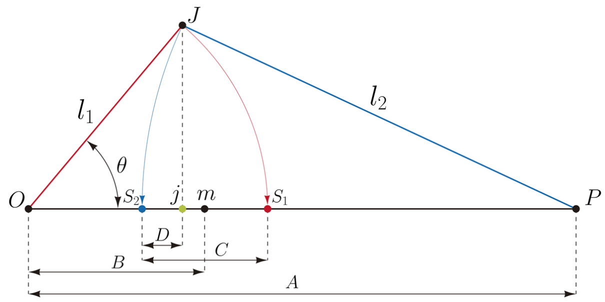

Figure 6: Theorem proof

$$\begin{cases} \overline{OS_1} = l_1 \\ \overline{OS_2} = \pm(A - l_2) \end{cases} \tag{3}$$

$$B = \overline{Om} = \frac{1}{2}[\overline{OS_1} + \overline{OS_2}] = \frac{1}{2}[l_1 + A - l_2] \tag{4}$$

$$C = \overline{S_2S_1} = \overline{OS_1} - \overline{OS_2} = l_1 - (A - l_2) \tag{5}$$

$$D = \overline{S_2j} = l_1 cos\theta - \overline{OS_2} = l_1 cos\theta + l_2 - A \tag{6}$$

Therefore:

$$BC = \frac{1}{2}[l_1 + A - l_2][l_1 - (A - l_2)] \tag{7}$$

$$BC = \frac{1}{2}[l_1^2 - (A - l_2)^2] \tag{8}$$

$$BC = \frac{1}{2}[l_1^2 - A^2 + 2Al_2 - l_2^2] \tag{9}$$

$$AD = A[l_1 cos\theta + l_2 - A] \tag{10}$$

$$AD = Al_1 cos\theta + Al_2 - A^2 \tag{11}$$

On the other hand, by the law of cosines:

$$l_1^2 + A^2 - 2Al_1 cos\theta = l_2^2 \tag{12}$$

From which we can infer:

$$l_1^2 + A^2 - l_2^2 = 2Al_1 cos\theta \tag{13}$$

$$l_1^2 - A^2 + 2Al_2 - l_2^2 = 2Al_1 cos\theta + 2Al_2 - 2A^2 \quad (14)$$

$$\frac{1}{2}[l_1^2 - A^2 + 2Al_2 - l_2^2] = Al_1 cos\theta + Al_2 - A^2 \quad (15)$$

Now, by Equation (9) and Equation (11), it follows that $BC = AD$, which is equivalent to Equation (1). Q.E.D.

Obtaining J

Thus, from this theorem $AD = BC$, and because $D = \vec{j} - \vec{S_2}, A = |\overline{OP}|, B = |\overline{Om}|$ and $C = \vec{S_1} - \vec{S2}$, then we can have an expression for the vector $\vec{j}$ as:

$$\vec{j} = \frac{|\overline{Om}|}{|\overline{OP}|}(\vec{S_1} - \vec{S_2}) + \vec{S_2}. \quad (16)$$

$\overrightarrow{Oj}$ is the projection of $\overrightarrow{OJ}$ over $\overline{OP}$, so the grey triangle $\triangle OjJ$ in Figure (7) is a right-angled triangle of which we know the hypotenuse denoted by l_1 and, from Equation (16), the opposite leg to the angle θ denoted by $|\overline{Oj}|$.

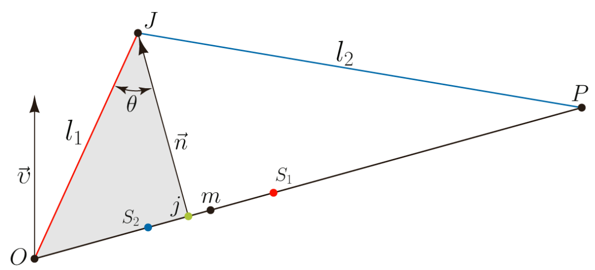

Figure 7: Solving J

With this data, we compute the magnitude of the adjacent leg to the angle θ by the Pythagorean theorem as follows.

$$|\vec{n}| = \sqrt{(l_2)^2 - |\overline{Oj}|^2} \quad (17)$$

The last step to find J is to calculate the vector $\vec{n}$ which must be perpendicular to $\overline{OP}$ with magnitude given by Equation (17) (Figure 7). This orthogonal vector $\vec{n}$ is obtained with two cross products by the following formula:

$$\vec{n} = \left(\overrightarrow{OP} \times \vec{v}\right) \times \overrightarrow{OP} \quad (18)$$

where $\vec{v}$ is a vector not parallel to OP given by the user and which defines the solving plane. Finally we project the adjacent leg to the angle θ in the direction of $\vec{n}$, where we can choose whether to use a positive or negative direction to obtain J at the top or bottom, thus:

$$\vec{n} = \sqrt{l_2{}^2 - |\vec{P} - \vec{j}|^2}\left(\frac{\vec{u}}{|\vec{u}|}\right) \quad (19)$$

And finally, from Equations (16) and (19): $\vec{J} = \vec{j} + \vec{n}$. This method can be implemented in any computer system capable of performing basic vector operations.

4. SOLUTION FOR HAL

In the case of HAL [20] where there are more than two links on the KC, the solution inspired by that of Jamali [3], is to create $n-2$ virtual links denoted as V_1, V_2, ..., V_{n-2} given n links l_1, l_2, ..., l_n, (Figure 8). We use this virtual links to compute the IK problem for the last link denoted as l_n and the first virtual link denoted by V_1. The solution using Equations (16) and (19), and denoted as J_1 will be the end effector for the next IK computation for V_2 and l_{n-1}. This process is repeated until we have the last two links denoted by l_1, l_2 to obtain the solution J_{n-1}.

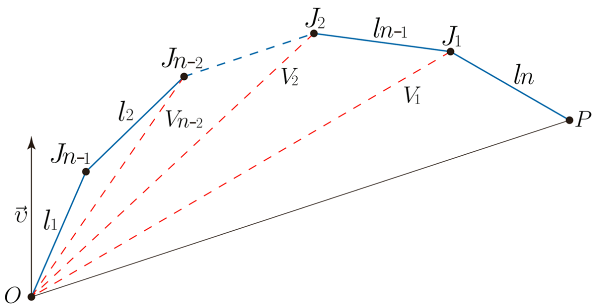

Figure 8: Solution for n links.

In order to use this virtual links denoted by V_k, we need to compute them as prismatic ones with automatic variable length, with the following process: For each link, we define a minimum length denoted by $V_{k_{min}}$ and a maximum length denoted by $V_{k_{max}}$, where the minimum length will be cero and the maximum will be compute with the following equation:

$$V_{k_{max}} = \sum_{i=1}^{n_k} l_i \tag{20}$$

Then, we will need to define a minimum distance from the origin O to the end effector P as $\overline{OP}min$ with any distance > 0. And a maximum distance from the origin O to the end effector P as $\overline{OP}max$ given by:

$$\overline{OP}max = \overline{V}_{k_{max}} \tag{21}$$

Thus, we can define the length for each virtual link V_k with the following formula:

$$|V_k| = \left((V_{k_{max}} - V_{k_{min}}) * \left(\frac{(|\overline{OP}| - \overline{OP}n_{min}) * 100}{\overline{OP}n_{max} - \overline{OP}n_{min}} \right) \right) + V_{k_{min}} \tag{22}$$

5. THE SPIRAL SHAPE

In order to obtain an spiral shape like Figure (2), we gradually decrease the length of the virtual vectors by a given factor denoted by δ:

$$|V_k| \leftarrow |V_k| - \delta \tag{23}$$

Where $|V_k| > 0$. This operation, increases the angle of each pair of links, until obtaining the spiral shape as seen in Figure (9).

6. RESULTS

To demonstrate the benefits of the new method, the spiral shaped example was implemented with 50 links. The platform used was Autodesk 3dsMax, using maxscript. This computational analisys was intended to test the transformational invariance, understood as the stability of the solution under the origin and the end effector transformations.

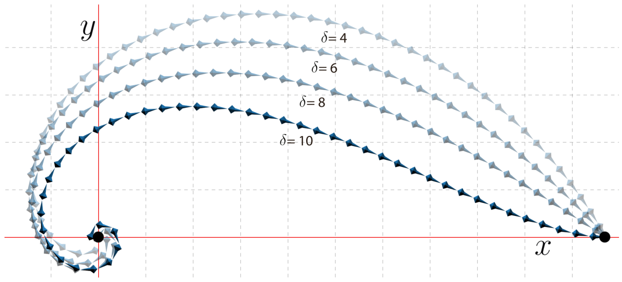

Figure 9: Obtaining the spiral shape

6.1 Native IK solvers.

In Figures (10, 11) we show examples of IK using the native IK solvers from 3dsMax, the IK is resolved, but the way the links are positioned, is far from the aesthetics would be expected for an animation production.

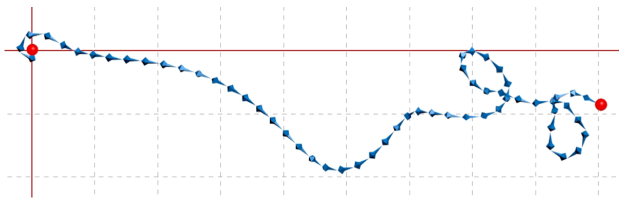

Figure 10: IK solved with History Dependent IK Solver from 3dsMax

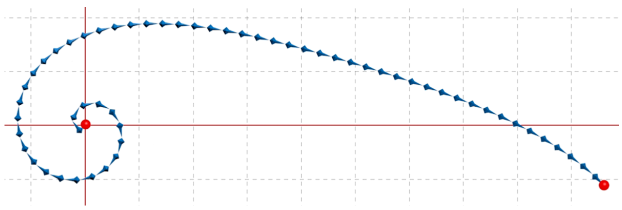

Figure 11: IK solved with History Independent IK Solver from 3dsMax

6.2 The new method result.

According to results, we can infer that the new method is an stable approach to solve the IK problem for HAL, and does not have any sign changes so the IK solution is the same regardless any origin or end effector transformation. Also, the new method presents a continuous and smooth result (Figure 12).

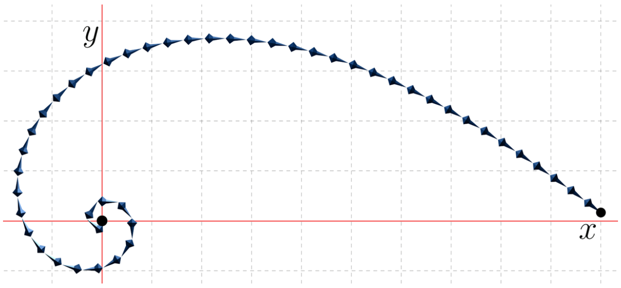

Figure 12: Result with the new method

Also, the method provides an unique solution for each state which is consistent with previous and subsequent states, giving an smooth movement on animation, although the solution for each state is computed independently of the others.

6.3 Results on three dimensions

Using three dimensions coordinates, the new IK solution remains stable during transformations at any quadrant, and the shape and movements are smooth. It is even possible to make a spring shape as seen in Figure (13).

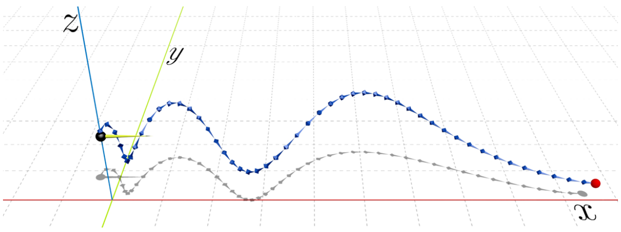

Figure 13: Resulting spring by modifying the $\vec{n}$ vector orientation.

7. CONCLUSIONS

The IK solvers are useful and important tools in animation, it is almost impossible to find a character rig without some sort of IK solver [18], and as the animation industry grows, more complex characters are designed bringing new problems for computer science.

Among the available IK solvers, the spline IK solver is currently the one who gives the best control over curves, while solving the IK problem, but with great effort from the animator that needs to move the vertices and handles of cubic Bézier curves [11].

Many research has been done about IK solutions for articulated and redundant KCs [3–8, 10, 12, 13, 17, 19, 20], but none of them from the sole animation perspective.

In this paper, we proposed a new method for computing the IK problem for highly articulated limbs. Our method solves an important problem in computer animation, because it allows an animator to create smooth and stable animations of complex curves like the ones of the spider concept shown in Figure (14).

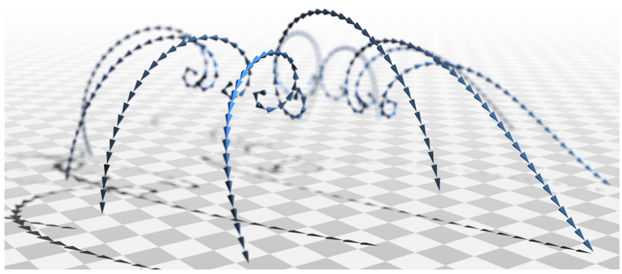

Figure 14: Resulting spring by modifying the $\vec{n}$ vector orientation.

8. CONTRIBUTIONS

The main novelty of this paper, is the new triangles theorem which allows to obtain a circle-circle intersection, with fast and reliable math. The use of this theorem, ensures an stable result under any 3D transformation.

In addition, this work provides an important contribution to the entertainment industry by enabling the making of animations that are currently difficult to obtain with the available animation software .

8.1 Future work

The triangles theorem we proposed, can be used for many other problems such as voronoi diagrams, the pulley problem, or any problem that could need a circle-circle intersection solution.

ACKNOWLEDGMENTS

This work has greatly benefited from discussions with Guillermo A. Parra, Gabriel Castillo, Alejandro García, Joel C. Huegel, Luis E. Mercado, Josué S. Reynoso, Raquel Ruiz, Gildardo Sánchez-Ante, Sergio Velázquez and Julio C. Zamora. To all of them, our thanks and admiration. Also, this work has been partially supported by IJJ, Universidad Panamericana and Instituto Tecnológico y de Estudios Superiores de Monterrey.

REFERENCES

[1] C. T. Adrià Colomé. Redundant inverse kinematics: Experimental comparative review and two enhancements.

[2] J. l. T. Alexandre Derouet-Jourdan, Florence Bertails-Descoubes. Stable inverse dynamic curves. *ACM Transactions on Graphics*, 2010.

[3] M. R. Annisa Jamali, Raisuddin Khan. A new geometrical approach to solve inverse kinematics of hyper redundant robots with variable link length.

[4] A. D. Aurel Fratu, Laurent Vermeiren. Using the redundant inverse kinematics system for collision avoidance.

[5] R. F. B. Benhabib, A.A. Goldenberg. A solution to the inverse kinematics of redundant manipulators.

[6] J. W. Burdick. On the inverse kinematics of redundant manipulators: Characterization of the self-motion manifolds. *Advanced Robotics*, 1989.

[7] W. K. C. Chevallereau. A new method for the solution of the inverse kinematics of redundant robots.

[8] C. W. W. I. Daniel R. Baker. Some facts concerning the inverse kinematics of redundant manipulators.

[9] M. A. J. J. Fatemeh Nourami. Improved circles intersection algorithm for localization in wireless sensor networks.

[10] T. G. Kang. *Solving Inverse Kinematics Constraint Problems for Highly Articulated Models*. Master's thesis, University of Waterloo, 2000.

[11] R. L. Levien. *From Spiral to Spline: Optimal Techniques in Interactive Curve Design*. Ph.D. thesis, University of California, Berkeley, 2009.

[12] C. Q. H. W. Li Sheng, Wang Yiqing. A new geometrical method for the inverse kinematics of the hyper-redundant manipulators.

[13] A. E. Mehmet, Bodur. Redundant manipulator for obstacle avoidance and inverse kinematics solution by least squares.

[14] I. D. W. Michael W. Hannan. Kinematics and the implementation of an elephants trunk manipulator and other continuum style robots. *Journal of Robotic Systems*, 2003.

[15] A. E. Middleditch. Intersection algorithms for lines and circles. *ACM Transactions on Graphics*, 1989.

[16] C. G. J. P. Robert W. Sumner, Matthias Zwicker. Mesh-based inverse kinematics.

[17] B. A. J. I. W. Srinivas Neppalli, Matthew A. Csencsits. A geometrical approach to inverse kinematics for continuum manipulators.

[18] D. C. Tim McLaughlin, Larry Cutler. Character rigging, deformations, and simulations in film and game production.

[19] C. Welman. *Inverse Kinematics and Geometric Constraint for Articulated Figure Manipulation*. Master's thesis, Simon Fraser University, 1993.

[20] S. Yahya, H. A. F. Mohamed, M. M. S., and S. Yang. A new geometrical inverse kinematics method for planar hyper redundant manipulators. *Conference on Innovative Technologies in Intelligent Systems and Industrial Applications*, 2009.

ABOUT THE AUTHORS

1. Enrique A. Rosales is Director of the Digital Animation Engineering Program at Universidad Panamericana in Guadalajara, México. erosales@up.edu.mx

2. Luis E. Falcón-Morales is Director of the Computer Science Master Degree program at Instituto Tecnológico y de Estudios Superiores de Monterrey in Guadalajara, México. luis.eduardo.falcon@itesm.mx