©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.3.VSTIA-513

Tracking and Estimating Tridimensional Position Through Camera-PT Array

Edrei Reyes-Santos [a], Hugo Jimenez-Hernandez [a], Leonardo Barriga-Rodríguez [a], JA Soto-Cajiga [a], Herlindo H [a], F. Carrizosa-Corral [a].

[a] CIDESI, Pie de la cuesta 702 Desarrollo San Pablo, Queretaro, Mexico.
Further author information: (Send correspondence to Edrei Reyes-Santos)
Edrei Reyes-Santos. E-mail: edrei.reyessantos@gmail.com, Telephone: +52 (442) 211 98 47.

Abstract

Several methods for 3D tracking use previous knowledge of scenario, which include workspace geometry, or active markers, that make feasible the tridimensional tracking of objects. However, in non-controllable scenarios there is a great challenge to guarantee a reliable and robust method. Parallel tracking method using PT-cameras are complicated because there are several conditions that affect motion detection (light, object displacement, PT-camera velocity, to mention a few). This work proposes a strategy for object tracking and estimating tridimensional position through camera-PT array. The camera array is used as a redundant way of focusing on reducing the error calculation. This method consists in simultaneously tracking the target object in all different cameras. Pan & Tilt are used as parameters of vectors in spherical coordinates. The tracking process is performed via active contours, which consists of a set of markers enclosing the target object and considering the contour as a high-energy zone. The tracking is then denoted as a Newton Rapson Optimization process which solves the problem of locating the maxima energy zone by superposing the latest reference position over the newest position in a given pair of images. Finally, our approach is tested in a controlled scenario. Luminance conditions are controlled and local references are used to match the estimated position and the real position.

Introduction

Growing cities make it difficult to monitor and analyze information for decision making. In order to deal with this situation, technology takes an important role as a tool to establish strategies for identifying and handling circumstances represent relevant information. Smart sensors are becoming more reliable and economical, which makes it feasible to implement solutions in crowded cities.

While the technology becomes more affordable, developing the complementary infrastructure required to analyze the large amounts of data in a short period becomes challenging. A typical task in automatic surveillance and monitoring is tracking objects. Nevertheless, this task is affected by several aspects such as camera perspective, color affectations and synchrony absence transmission which make it infeasible to track objects in outdoor scenarios.

Some works, such as [1, 2, 3], proposed strategies for reliable tracking in outdoor scenarios, although, as it can be appreciated the repeatability and accuracy of these methods is compromised in inverse factor. This work proposes a strategy for object tracking and estimating tridimensional position through camera-PT array, the basis of this approach is to dynamically track the target object in all different cameras.

Pan & Tilt parameters of each camera are used as parameters of vectors in spherical coordinates. The radius represents the distance between a specific camera and the target object. Using the same parameters from each camera, a linear system is generated, where the unknown values represent each camera's radius. Beyond this point, the process is considered as an optimization task, the objective of which is reducing the error estimation of the distance from the camera-PT array to the target object.

It is considered that the distance between real and estimated centers of the image is so small that it can be ignored when estimating the tridimensional position. Then, each camera is represented by a two-dimensional plane using local coordinates. For validation purposes, an experimental analysis is done. This analysis consists in validating artificial trajectories in closed scenarios, matching the real and estimated trajectories.

Active Contours

By selecting energy terms that are appropriate for minimization, the model can be forced to the desired solution enduring local minimums. The obtained active model approaches the desired solution when located near it. Models that minimize energy have been widely used in vision, perhaps being Sperling's stereo model [6] one of the earliest works. The problem that active contours attempts to solve is locating salient image features (i.e. edges, lines, and subjective contours) and tracking these features. Active contours differs from other techniques for finding salient contours since it uses an active model. Thus, it minimizes its energy functional and presents dynamic behavior. Considering that the contours appear to slither as their energy is minimized, they are known as snakes (see Figure 1)[14]. The robustness of active contours is dependable on what features are used as references to perform the minimization step, in particular the approach presented in this work employs the first derivative.

The snake approach does not attempt to solve the whole salient image contours problem, it requires other techniques to approximate the desire contour before beginning. Although, even when these techniques are not suitable, semiautomatic interpretation of the image might still be achieved by the snake approach by using an expert user to place the snake near the intended contour. The basic snake model consists of a controlled continuity [7] spline

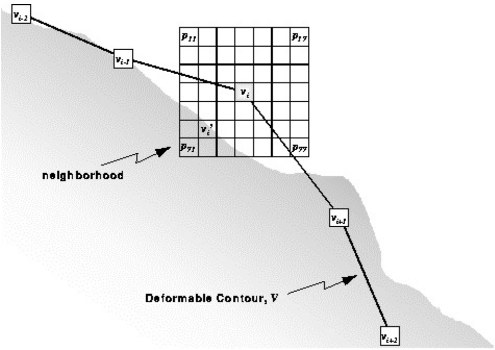


Figure 1. *Active contour.*

under the influence of image and external constraints energy.

The internal spline energy imposes a piecewise smoothness constraint. The image energy approaches the snake to salient image features such as lines, edges and subjective contours. The external constraints locate the snake near the desired local minimum. The position of the snake can be parametrically represented by $v(s) = (x(s), y(s))$, thus, the energy functional is given by

$$\begin{aligned} E^*_{snake} &= \int_0^1 E_{snake}(v(s))ds \\ &= \int_0^1 E_{int}(v(s)) + E_{image}(v(s)) + E_{con}(v(s))ds, \end{aligned} \tag{1}$$

where E_{int} is the internal energy of the spline that arises from bending, E_{image} represents the image energy, and E_{con} refers to the external constraint energy [5]. E_{int} and E_{image} are defined in the next subsection.

Internal Energy

Internal energy spline can be expressed as

$$E_{int} = \frac{(\alpha(s)\,|V_s(s)|^2 + \beta(s)\,|V_{ss}(s)|^2)}{2}. \tag{2}$$

The spline energy comprises a first-order term adjusted by selecting $\alpha(s)$ and a second-order term tuned by choosing $\beta(s)$; $\alpha(s)$ provokes the snake to behave as a membrane and $\beta(s)$ induces it to act as a thin plate. When $\beta(s) = 0$ is selected, the snake becomes a second-order discontinuity and develops a corner [5].

Image Energy

To make snakes useful for early vision, energy functionals are required to approach them to the most salient features of the image. This method considers three energy functionals which approach the snakes to lines, edges and terminations. By combining these using weights, the total image energy can be expressed.

$$E_{image} = w_{line}E_{line} + w_{edge}E_{edge} + w_{term}E_{term}. \tag{3}$$

Several snake behaviors might be created depending on the selected weights [5].

E_{line}: The most commonly used image functional is the image intensity. The sign of W_{line} determines whether the snake will be attracted to light or dark lines.

E_{edge}: It is possible to find the edges in an image by using a very simple energy functional defined by $E_{edge} = -|\nabla I(x,y)|^2$, then the snake will be attracted to contours with large image gradients.

E_{term}: The termination of line segments and corners can be found by slightly smoothing the image and using the curvature of level lines. Being $C(x,y) = G_\sigma(x,y) \otimes I(x,y)$ the smoothed image, $\theta = \tan^- C_y/C_x$ the gradient angle, and $n = (\cos\theta, \sin\theta)$ and $n_\perp = (-\sin\theta, \cos\theta)$ the unit vectors along and perpendicular to the gradient direction; the curvature of the level contours is given by

$$\begin{aligned} E_{term} &= \frac{\delta\theta}{\delta n_\perp} \\ &= \frac{\delta^2 C/\delta^2 n_\perp}{\delta C/\delta n} \\ &= \frac{C_{yy}C_x^2 - 2C_{xy}C_xC_y + C^{xx}C_y^2}{(C_x^2 C_y^2)^{3/2}}. \end{aligned} \tag{4}$$

Snakes can be attracted to edges or terminations depending on E_{term} and E_{edge}.

An unusual characteristic of the snake model is that it may present hysteresis when there is a moving stimuli, as the snake is constantly trying to minimize their energy [5].

Distance function

Defining A as the set of all pairs of integers (i, j), the function f from $A \times A$ into the nonnegative integers is called a distance function (see Figure 2) if it satisfies:

a) Positive definite, $f(x,y) = 0$ if and only if $x = y$.

b) Symmetric, $f(x,y) = f(y,x)$ for all x, y in A.

c) Triangular, $f(x,z) \leq f(x,y) + f(y,z)$ for all x, y, z in A.

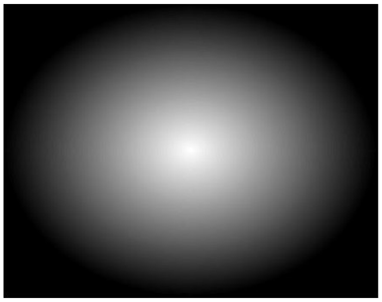


Figure 2. *Distance function of a circle.*

Estimating Object Position

Our proposal is based on redundant references given by a set of PT cameras. All cameras are synchronized, tracking and centering in its local projection the object of interest. For each time stamp, PT parameters as well as PT Camera position are referred in a global coordinates reference. Thus, with this information a linear system is built and local position is computed.

Pseudoinverse of Matrices

The inverse matrix concept can be extended for rectangular matrices, resulting in the pseudoinverse. This matrix is a particular case of the generalized inverse matrix, which is described below.

As matter of fact, matrix multiplication is not commutative; thus, for any matrix $A \in \mathbb{R}^{m\times n}$, there are almost two matrices A^{-1-} and A^{-1+} such that left-multiplication or right-multiplication with A give us the identity matrix I. Whenever A^{-1-} becomes the same that A^{-1+} we say that it is a true inverse matrix.

This situation become feasible when A is is squared and null space is zero. Otherwise, A^{-1-} and A^{-1+} are different matrices. Both matrices are useful to give an approximation to linear system expressed in A. This is, for a given system $A \times \mathbf{x} = B$ for non-squared matrix A, a possible solution becomes those matrix, such that be the left-inverse of this system; i.e $A^{-1-}A \times \mathbf{x} = A^{-1-}B$. In general terms, is the linear solution of rank projection of A matrix. The pseudoinverse of A as a function of the orthogonal projections associated with the subsets $R(A)$ and $N(A)$ is defined as follows: Let $A \in \mathbb{R}^{mxn}$, the pseudoinverse matrix of A or Moore-Penrose matrix $A^\dagger \in \mathbb{R}^{mxn}$ is a matrix which satisfies [11]

1.− $P \equiv A^\dagger A$ is the orthogonal projection on $N(A)^\perp$.

2.− $\overline{P} \equiv AA^\dagger$ is the orthogonal projection on $R(A)^\perp$.

Singular value decomposition

The singular value decomposition (SVD) of a matrix $A \in \mathbb{R}^{mxn}$ provides an extension of the concept of square matrix diagonalization, i.e., an orthogonal transformation of A into a diagonal matrix of $n \times n$ that retains the norm $\|\|_2$. The following theorem establishes the existence of the SVD for any rectangular matrix. Let $A \in \mathbb{R}^{mxn}$ with $rank(A) = r \leq n$ then there are unit matrices $U \in \mathbb{R}^{mxm}$ and $V \in \mathbb{R}^{nxn}$ and a matrix $\Sigma \in \mathbb{R}^{mxn}$ such that:

$$\begin{aligned} A &= U\Sigma V^H \qquad (5) \\ \Sigma &= \begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix} \\ \Sigma_1 &= \begin{pmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ & & \ddots & \\ 0 & 0 & \dots & \sigma_r \end{pmatrix} \in \mathbb{R}^{rxr}. \end{aligned}$$

The non-null elements of the main diagonal of the Σ_1 block satisfy $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$ and are called singular values of A [13]. Note that if $U = (u_1 ... u_r)$ and $V = (v_1 ... v_r)$ where u_i and v_i are the columns of U and V respectively, then the SVD of A can be written as

$$A = U\Sigma H = \sum_{i=1}^{r} \sigma_i u_i v_i^H. \qquad (6)$$

Hence, the SVD of A can be interpreted as the sum of $r = rank(A)$ matrices of rank 1 (that is the matrices $\sigma_i u_i v_i$).
The SVD of A is relevant since it allow to define the solution to the linear least square problem as a function of the matrices $U\,V\,y\,\Sigma$ mentioned previously [9].

Estimating Object's Position

Under the assumption that PT camera might align an object of interest with its optical axis; the PT camera parameters represent a vector in direction of it. Similarly with all others cameras. Thus, in a particular time stamp, with camera position in a global reference, and its direction vector to the object, a linear system is built where intersection represents local coordinates where object is located.

This system become feasible whenever equation system be free of any error source. But some approximation can be reach under follow assumptions

- Center of image is the same than the optical axis.
- The relation Camera Resolution and step PT parameters are consistent.
- Distance between camera and object are far enough that object volume is dismiss.
- Error object location is assumed as Gaussian.

Then, a linear system might be solved via pseudoinverse, which give an approach to the linearization of this system. For further works, each one of last described situations should be improved.

The difference between two images is used for detecting objects, as shown in Figure 3. Then, the active contours algorithm proposed by M. Kass et al. [5] is employed to locate the contour near the target object contour. This active contour model is defined by an energy functional and it finds a solution using techniques of variational calculus. In the work of Kass et al. they represent the contour with a vector, $v(s) = (x(s), y(s))$, which has the arc length, s, as a parameter. Also, they define the energy functional of the contour and describe a method for finding the contours associated with the local minimum of the functional.
The difference between two images enables obtaining the movement in the scene (the object being moved through the workspace in a controlled environment). After detecting the object by using the difference between two images, the centroid is obtained through the distance function algorithm (Figure 3) proposed by Rosenfeld & Pfaltz [8]. The distance function, also known as distance transform, of a binary image X relates each pixel with the shortest path length (straight line) to the complement of the binary image X. The algorithm is described by the following steps [15]:

1. Forward scan of all pixels $p \in D_X$
   if $X(p) = 1$
   $X(p) = 1 + Min\left\{X(q) : q \in N_G^-(p)\right\}$.

2. Backward scan of all pixels $p \in D_X$
   if $X(p) \neq 0$
   $X(p) = Min\left\{X(p), 1 + Min\left\{X(q) : q \in N_G^+(p)\right\}\right\}$.

The centroid of the object is calculated to locate the active contour on its x, y coordinates.

The starting radius from the markers of the active contour to the centroid is obtained by using a factor of $1.5 * max(distancefunction)$, assuring that the active contour will be located near the target contour (Figure 4), enabling the method of M. Kass et al [5] to converge in each iteration of the algorithm proposed herein.

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.3.VSTIA-513

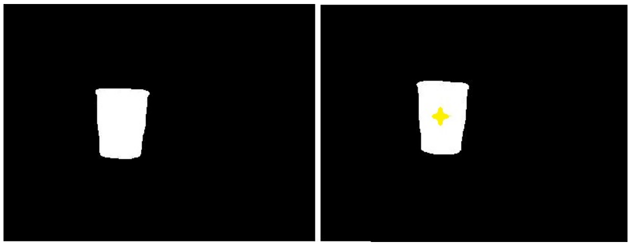


Figure 3. *a) Object detection using frame difference. b) Centroid of the target object.*

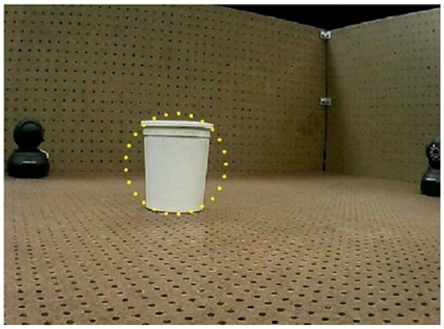


Figure 4. *Active contour located near the target object.*

When the active contour converges to the desired contour (as seen in Figure 5), the centroid is recalculated. This time, the snake points are used to ensure it belongs to the target object and not to some false positive result of the movement detection. Hence, tracking and estimating the position of undesired objects is discarded. The metric space generated by the active contour (using the first derivative) for the test object is presented in Figure 6.

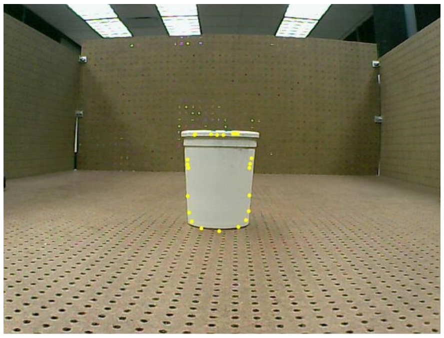


Figure 5. *Active contour using 20 points.*

Once the x, y coordinates of the active contour matching the target object are obtained, the x and y axis distance from the coordinates of the centroid to the center of the image are calculated. These distances are then employed to center the objects and the vision field of the camera. This is accomplished by using the Pan & Tilt of each camera, which will be referred in the rest of this paper as θ and ϕ.

Therefore, to obtain the tridimensional position of the object, a system of equations is created starting from the following assumptions:

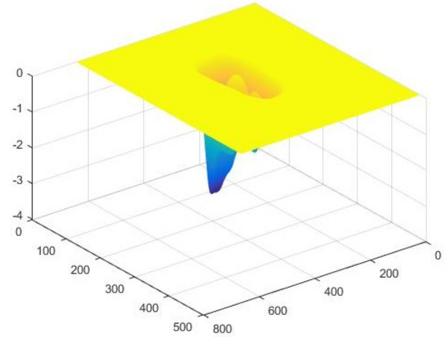


Figure 6. *Metric space employed in active contours.*

1.– The distance between real and estimated center of the image is so small that it can be ignored when estimating the tridimensional position.

2.– The position of the cameras in the workspace is known, as required by the geometric approach used.

3.– Each camera has a global reference from which every trajectory begins.

As each camera centers the target object in its vision field, it gives the (x, y) coordinates of the object within the image and the angles θ and ϕ (see Figure 7).

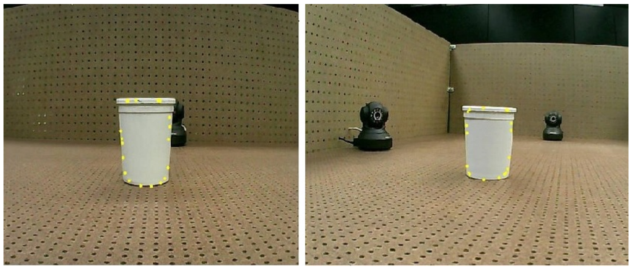


Figure 7. *Target object centered by the cameras.*

In order to develop the mathematical model that obtains the position of the object, we define the tridimensional position of the object as the coordinates (X_r, Y_r, Z_r) and the two-dimensional coordinates of the cameras as $(x_1, y_1; x_2, y_2 \ldots x_n, y_n)$ which can also be expressed as $p_i = (x_i, y_i)$, also the PT-cameras parameters are known and given by $(\theta_1, \phi_1, \theta_2, \phi_2 \ldots \theta_n, \phi_n)$, finally using spherical coordinates the calculated distances from the object to each camera are denoted as r_i.

Using local coordinates, for each camera the object is located in

$$R_i = \underbrace{p_i}_{camera\ position} + \underbrace{(r_i \sin\phi \cos\theta\ , r_i \sin\phi \sin\theta)}_{Ob\ ject\ distance\ to\ the\ camera}. \qquad (7)$$

Then for the whole camera-PT array, a linear system of equations is built as follows:

$$\begin{pmatrix} R_1 \\ R_2 \\ R_3 \\ \vdots \\ R_n \end{pmatrix} = \begin{pmatrix} p_{1x} & + & p_{1y} & P_{1x} & P_{1y} \\ p_{2x} & + & p_{2y} & P_{2x} & P_{2y} \\ p_{3x} & + & p_{3y} & P_{3x} & P_{3y} \\ \vdots & + & \vdots & \vdots & \vdots \\ p_{nx} & + & p_{ny} & P_{nx} & P_{ny} \end{pmatrix}, \qquad (8)$$

where $P = (r_i \sin\phi \cos\theta\ , r_i \sin\phi \sin\theta)$.

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.3.VSTIA-513

As observed in the last equation, it is assumed that all components are linearly related. In real scenarios this might not hold and the linear relation needs to be approximated.

Assuming that noise in physical variables is normal, pseudoinverse is employed. $R_1 \dots R_n$ values represent the system solution and the target object position can be estimated using any R_i value and substituting in any reference solution.

Finally, our approach is tested in a controlled scenario. Luminance conditions are controlled and local references are used to compare the real and the estimated position, figures 8 and 9 present illustrations of the testing scenario.

In this scenario, there are artificial objects that are displaced.

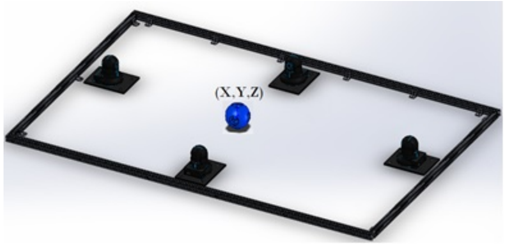


Figure 8. *Testing scenario.*

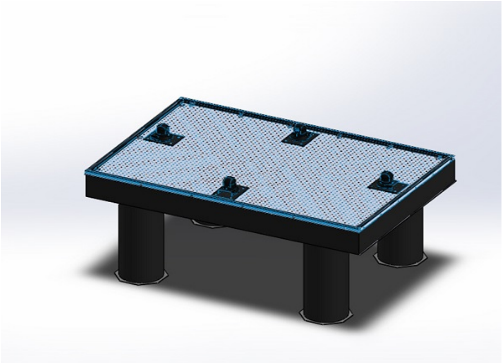


Figure 9. *Testing environment.*

For each displacement, every camera centers the object and their parameters θ and ϕ are obtained. In this paper, only xy plane displacements are being considered (Figure 10).

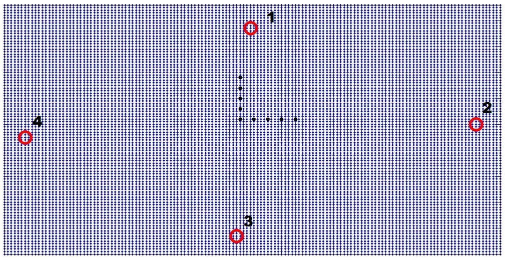


Figure 10. *Virtual workspace, where each red circle represents a camera and the black dotted line is the trajectory of the object.*

Table 1 presents the average iterations for each camera in the array, that is the average number of iterations that each camera needed to center the object during each tested trajectory. This data is also presented as graphs in figures 11 and 12. Although, the most important contribution of this work is proving that object tracking and estimating tridimensional position is possible by using a camera-PT array, the information regarding iterations will be useful for future work.

Camera array average iterations during each tested trajectory.

Trajectory	Iterations by camera
1.- Set square	2
2.- Wave Square	3
3.- Saw tooth	3
4.- Rectangle	3
5.- Triangle	2
6.- Circle	2
7.- Square	2
8.- Straight vertical	3
9.- Straight horizontal	2
10.- Diagonal	3

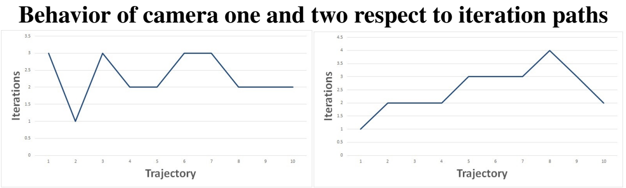


Figure 11. *Camera 1 & 2 iterations in experimentation.*

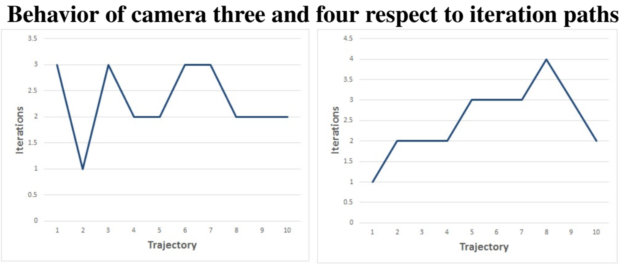


Figure 12. *Camera 3 & 4 iterations in experimentation.*

Paths used in experimentation. Figures 13 and 14 show some of the trajectories that were used during experimentation to compare them with the position estimated by the algorithm proposed herein. Each trajectory was designed to test the effectiveness of our algorithm, in terms of its ability to detect whether the final position is different from the initial one.

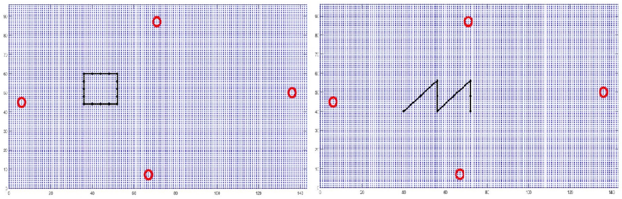


Figure 13. *Square & Saw tooth.*

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.3.VSTIA-513

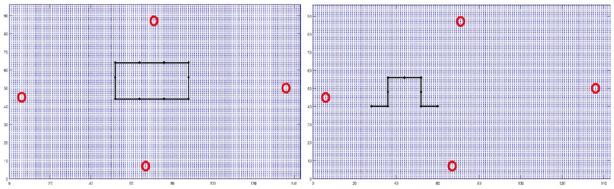
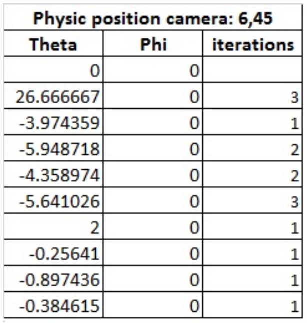


Figure 14. *Rectangle & Square wave.*

Figure 15 presents information obtained from one of the cameras, that is necessary for calculating the tridimensional position of the object. Each camera in the array provides the same parameters for each trayectory employed in object tracking.



Physic position camera: 6,45		
Theta	**Phi**	**iterations**
0	0	
26.666667	0	3
-3.974359	0	1
-5.948718	0	2
-4.358974	0	2
-5.641026	0	3
2	0	1
-0.25641	0	1
-0.897436	0	1
-0.384615	0	1

Figure 15. *Information table to obtain tridimensional position of the object.*

Using the proposed approach, the estimated positions for different trajectories are presented in figures 16, 17 and 18.

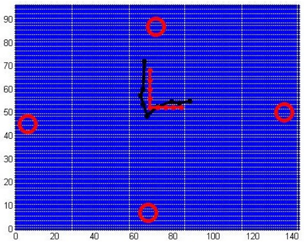


Figure 16. *Real vs estimated L trajectory.*

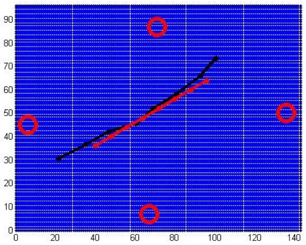


Figure 17. *Real vs estimated diagonal trajectory.*

A comparison of the real and estimated coordinates for the trajectories is shown in figures 19, 20 and 21.

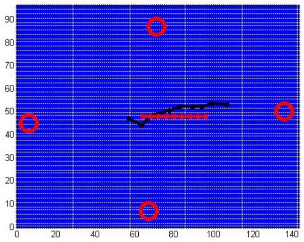
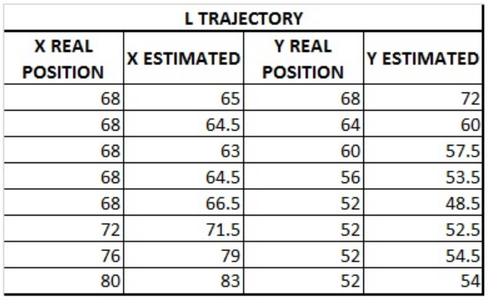
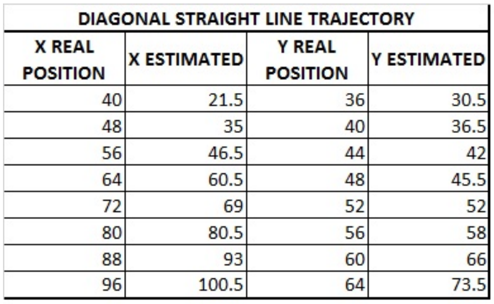


Figure 18. *Real vs estimated horizontal trajectory.*



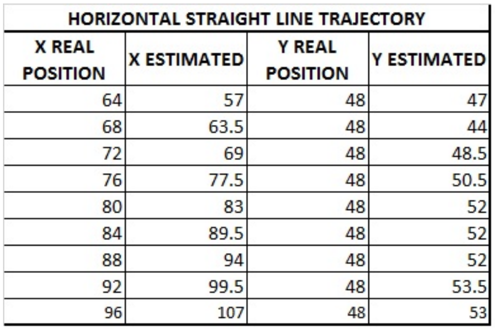
L TRAJECTORY			
X REAL POSITION	**X ESTIMATED**	**Y REAL POSITION**	**Y ESTIMATED**
68	65	68	72
68	64.5	64	60
68	63	60	57.5
68	64.5	56	53.5
68	66.5	52	48.5
72	71.5	52	52.5
76	79	52	54.5
80	83	52	54

Figure 19. *Comparison of real and estimated coordinates for L trajectory.*



DIAGONAL STRAIGHT LINE TRAJECTORY			
X REAL POSITION	**X ESTIMATED**	**Y REAL POSITION**	**Y ESTIMATED**
40	21.5	36	30.5
48	35	40	36.5
56	46.5	44	42
64	60.5	48	45.5
72	69	52	52
80	80.5	56	58
88	93	60	66
96	100.5	64	73.5

Figure 20. *Comparison of real and estimated coordinates for diagonal trajectory.*



HORIZONTAL STRAIGHT LINE TRAJECTORY			
X REAL POSITION	**X ESTIMATED**	**Y REAL POSITION**	**Y ESTIMATED**
64	57	48	47
68	63.5	48	44
72	69	48	48.5
76	77.5	48	50.5
80	83	48	52
84	89.5	48	52
88	94	48	52
92	99.5	48	53.5
96	107	48	53

Figure 21. *Comparison of real and estimated coordinates for horizontal trajectory.*

Conclusion

This work presented a unified approach for object tracking and estimating tridimensional position based on active contours. Distance function is employed to determine the centroid of the object and the tracking is achieved by comparing it to the center of the image. Hence, the parameters of the camera-PT array are used to center the object and the vision field of each camera. Finally, the pseudoinverse is implemented to solve the equation system generated by using these parameters. Experimental data supports

DOI: 10.2352/ISSN.2470-1173.2016.3.VSTIA-513

our initial assertion that once the object is detected and focused by the camera array, it is possible to calculate the distance from each camera to the object which allows determining the tridimensional position of the object. On the other hand, the algorithm requires that the object is recognized by at least three cameras in order to effectively estimate its location relative to the camera array. Occlusions do not hinder the method unless the visibility of the object is limited to less than three cameras. Furthermore, the proposed method was successfully tested in a controlled scenario. Finally, this work will provide the basis for an evolution of the algorithm and future work focusing on moving the cameras to spatial position and experimenting with conditions resembling a real environment.

Acknowledgments

We would like to thank our colleagues, CIDESI^{TM} and ASA-CONACYT project with number ASA-2014-1-242864 for their support, regarding the infrastructure for the development of this work.

References

[1] Jin, Z.,& Bhanu, B. Analysis-by-synthesis: Pedestrian tracking with crowd simulation models in a multi-camera video network. Computer Vision and Image Understanding, 134, 48-63.(2015)

[2] Ali, M. N., Abdullah-Al-Wadud, M., & Lee, S. L. Multiple object tracking with partial occlusion handling using salient feature points. Information Sciences, 278, 448-465.(2014).

[3] Maddalena, L., Petrosino, A., & Russo, F. People counting by learning their appearance in a multi-view camera environment. Pattern Recognition Letters, 36, 125-134.(2014).

[4] Ballesta, S., Reymond, G., Pozzobon, M., & Duhamel, J. R. A real-time 3D video tracking system for monitoring primate groups. Journal of neuroscience methods, 234, 147-152. (2014).

[5] Kass, M., Witkin, A., & Terzopoulos, D. Snakes: Active contour models. International journal of computer vision, 1(4), 321-331.(1988).

[6] Sperling, G. Binocular vision: A physical and a neural theory. The American Journal of Psychology, 461-534.(1970).

[7] Terzopoulos, D. Regularization of inverse visual problems involving discontinuities. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 4, 413-424. (1986).

[8] Rosenfeld, A., & Pfaltz, J. L. Distance functions on digital pictures. Pattern recognition, 1(1), 33-61. (1968).

[9] Cahueñas, O., Ramos, L. M. H., & Raydan, M. Precondicionamiento por aproximaciones a la matriz pseudoinversa para el problema de mínimos cuadrados lineales. (2009).

[10] Björck, Ake, Numerical methods for least squares problems, Siam, Philadelphia, PA 1996, pg. 9.

[11] Penrose, R, A generalized inverse for matrices. In Mathematical proceedings of the Cambridge philosophical society, Vol. 51, No. 03, pg. 406-413. 1955.

[12] Stoer, Josef, & Roland Bulirsch, Introduction to numerical analysis, Vol. 12. Springer Science & Business Media, New York, NY, 2013, pg. 345.

[13] Golub, G. & Van Loan, C. Matrix Calculations. Johns Hopkings University Press, Baltimore, MD 2013, pg. 389.

[14] García, E. ”CONTORNOS ACTIVOS.” Universidad de Matanzas, 1,6. (2003).

[15] Soille, P. Morphological Image Analysis: Principles and Applications. Springer Science & Business Media, New York, NY, 2004, pg. 48.

Authors Biography

Edrei Reyes Santos received the BSc. in mechatronics engineering at Instituto Tecnológico de Poza Rica (ITSPR), México, 2012. Currently a candidate to a MSc. degree in Mechatronics at CIDESI, in Querétaro, México. Nowadays, his recent researches focus on tracking and estimating tridimensional position through camera-PT array, detecting dynamic object's interaction and augmented reality.

Hugo Jiménez Hernandez, PhD in Computer Vision. in CICATA-IPN, Mexico, MS in Computer Sciences in CIC-IPN,Mexico, Bachelor in Computer Sciences in ITQ-Mexico. His current interest includes Automatic Learning, Computer Science Theory and Automatic Visual Inferences. Nowadays he is the head of Computer Vision Lab at CIDESI-Mexico. He has around 50 works in several international journals and congress.

Leonardo Barriga Rodríguez, PhD in Mechatronics, in CIDESI, Mexico; with 10 years of experience in software development, computer vision, image processing and inspection robotics. He has participated in the development of more than 20 research projects in the industry. He is currently a research professor of CIDESI (Center for Engineering and Industrial Development).

J.A. Soto-Cajiga (M'11) received the BEng Degree in Control and Instrumentation Engineering from the Technological Institute of Queretaro, Mexico, in 2003, the MEng Degree in Mechatronics from the PICYT (interinstitutional graduate in science and technology), Mexico, in 2006 and the Ph.D. Degree also from PICYT in 2012. He works for CIDESI (Engineer Center and Industrial Development), as researcher in the applied research area, designing embedded systems and real time signal processing algorithms. He is also professor of graduate level in PICYT, and his research interests include real-time signal processing implemented in hardware, algorithms parallelizing for signals and video processing and electronics for ultrasonic PIGs.

Herlindo Hernandez Ramirez received the BSc. in mechatronics engineering at Universidad Politecnica de Victoria (UPV), México, 2011. Aspiring to a MSc. degree in Mechatronics at CIDESI, in Querétaro, México. His research interests include automatic computational modelling of humans behavior and visual surveillance.

Fernando Carrizosa-Corral received the B.E. degree in mechatronics engineering from the Instituto Tecnologico de Culiacan, Sinaloa, Mexico, in 2013. In 2015, he enrolled as a Master student at Centro de Ingenieria y Desarrollo Industrial (CIDESI), in Queretaro Mexico. He is interested in embedded systems, computer vision, robotics, and automatic control. His current research examines FPGA embedded processor-based vision systems and implementation of modern control techniques.