

On triangulation algorithms in large scale camera network systems

Andrea Masiero and Angelo Cenedese

Abstract

Geometric triangulation is at the basis of the estimation of the 3D position of a target from a set of camera measurements. The problem of optimal estimation (minimizing the L_2 norm) of the target position from multi-view perspective projective measurements is typically a hard problem to solve. In literature there are different types of algorithms for this purpose, based for example on the exhaustive check of all the local minima of a proper eigenvalue problem [2], or branch-and-bound techniques [3]. However, such methods typically become unfeasible for real time applications when the number of cameras and targets become large, calling for the definition of approximate procedures to solve the reconstruction problem.

In the first part of this paper, linear (fast) algorithms, computing an approximate solution to such problems, are described and compared in simulation. Then, in the second part, a Gaussian approximation to the measurement error is used to express the reconstruction error's standard deviation as a function of the position of the reconstructed point. An upper bound, valid over all the target domain, to this expression is obtained for a case of interest. Such upper bound allows to compute a number of cameras sufficient to obtain a user defined level of position estimation accuracy.

I. INTRODUCTION AND CAMERA MODEL

The problem of estimating the 3D coordinates of a target from a set of sensor measurements, named reconstruction procedure, is at the basis of motion capture and localization/tracking systems. In the general framework of sensor networks, the problem is usually solved by geometric triangulation or trilateration of measurements [10] and similarly, in the context of camera network systems, the reconstruction of the 3D target location from the information of two cameras' image planes can be attained by means of geometric triangulation of measurements [2]. In particular, in this paper, we consider the case of large scale systems, such as for example those related to marker motion capture with many subjects (that is hundreds/thousands of markers and tens/hundreds of cameras): Nowadays, motion capture systems are used for a wide range of applications, going from biomedical to military, from the movie industry to the sport disciplines. On the one hand, the request of a more and more accurate estimation of the target positions is leading to the use of large camera network systems. On the other hand, the real time use of the system imposes stringent computational requirements.

We assume the camera model as a calibrated pinhole camera; therefore, given a point target ϕ in the 3D space, the measurement taken from camera j corresponds to a 2D position q_j on its viewing sensor (i.e. on its image plane) that is generated by the intersection of the camera's image plane with the ray passing through the point ϕ and the camera's optical center O_j , as shown in Fig. 1 together with the reconstruction procedure (assuming perfect measurements, i.e. infinite sensor sensibility and no measurement noise).

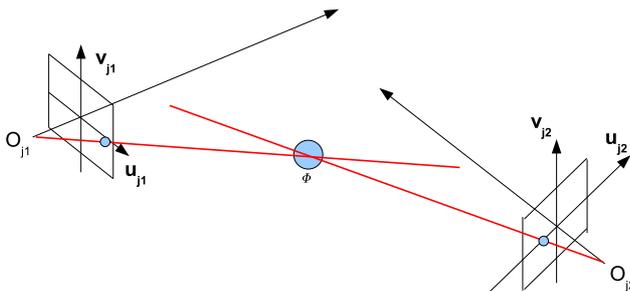


Fig. 1. Projection of a 3D point ϕ onto the image plane and triangulation between two cameras. Measurement q_j is the projection of ϕ on the image plane of camera j . The crossing point between two rays related to the same target seen by different cameras allows to obtain the target's 3D position.

Theoretically, the reconstruction procedure could be solved by two measurements only, but the presence of measurement and quantization noise in addition to specific alignment conditions, suggest the use of many more measurements for the target location estimation. Actually, in large motion capture systems characterized by complex scenarios, the reconstruction procedure must rely on several rays, in order to meet the requirements in terms of accuracy and robustness to the target localization. Exploiting a large number m of 2D camera measurements (3D rays), indeed:

- the visibility of the targets increase and many more targets can be reconstructed,
- the number of ghost target (reconstructed artifacts) decreases, since the reconstruction needs to be obtained by a larger number of 2D measurements,

A. Masiero and A. Cenedese are with the Department of Information Engineering, Università di Padova, via Gradenigo 6/B, 35131 Padova, Italy masiero@dei.unipd.it angelo.cenedese@unipd.it

The research leading to these results has received funding from the European Community's Seventh Framework Programme under agreement n. FP7-ICT-223866 FeedNetBack and n. 257462 HYCON2 Network of excellence. This activity contributes to the seed project R3D of the Department of Information Engineering, University of Padova.

- the reconstruction error is reduced.

In detail, let $\phi = [x, y, z]^\top$ be a 3D target to be reconstructed, then $q_j = [u_j, v_j]^\top$ is the measurement on the image plane of camera $j \in [1, \dots, m]$, given as follows:

$$\begin{bmatrix} u'_j \\ v'_j \\ d_j \end{bmatrix} = P_j \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \quad \begin{bmatrix} u_j \\ v_j \end{bmatrix} = \frac{1}{d_j} \begin{bmatrix} u'_j \\ v'_j \end{bmatrix}, \quad (1)$$

where P_j is the projection matrix associated to j th camera and taking into account the intrinsic and extrinsic parameters [6] [1], and d_j is the distance from the camera to the target plane (assuming the focal length f as known and normalized to 1). Also, let I_j be the image plane and Π_j the target plane (parallel to I_j and passing through ϕ). In practice, being q_j a noisy measurement, its projection on Π_j is $\phi_j \approx \phi$, and finally m_j is the unit vector along the direction from O_j to ϕ_j . Furthermore, let $\hat{\phi} = [\hat{x}, \hat{y}, \hat{z}]^\top$ be the estimated position of the target by the camera network and $\hat{q}_j = [\hat{u}_j, \hat{v}_j]^\top$ its projection on the image plane of camera j (obtained with the same procedure described above). Then, the reconstruction error on the j th image plane is defined as: $e_j = q_j - \hat{q}_j$.

Since now the value of the distance d_j is unknown, the goal of the reconstruction procedure is to estimate the 3D target position exploiting multi-view data on several cameras and minimizing the sum of the square (image plane) errors (reconstruction minimizing the L_2 error norm), i.e. $\sum_{j=1}^m e_j^\top e_j$. Also, to take into account the different level of noise in the measurements and the possible correlation among them, the following functional $\Theta(\phi)$ is introduced:

$$\Theta(\phi) = \mathbf{e}^\top W \mathbf{e}, \quad (2)$$

where $\mathbf{e} = [e_1^\top \ e_2^\top \ \dots \ e_m^\top]^\top$, and $W \geq 0$ is a proper weighting matrix.

When the number of cameras is low, optimal L_2 reconstruction can be solved by probing all the local minima of a proper eigenvalue problem [2] [8]. However, because of computational complexity issues, such method becomes impossible in practice when m increases.

Alternatively, an approximate solution can be computed by means of optimization methods [9], although not ensuring to take to the optimal solution: Usually to reduce the risk of remaining stuck to local minima an initial approximate solution quite close to the true one has to be provided.

This work is organized in two parts, that reflect the twofold contribution the paper aims at: In Secs. II and III, different reconstruction algorithms and their iterative versions are described and compared with respect to three marker distribution case studies and an increasing number of cameras. Then, in Sec. IV, a condition for a “good” reconstruction is discussed and an upper bound to the reconstruction error variance is obtained, which allows to compute the number of cameras needed to attain a chosen reconstruction performance. Finally, in Sec. V some conclusions are drawn.

II. TRIANGULATION METHODS

The reconstruction error on the image plane e_j can be back projected on the plane Π_j : The error on the target plane e'_j is related to e_j by geometrical similarity: $e'_j = d_j e_j$. Substituting (1) into the above equation yields:

$$e'_j = \left(d_j q_j - \bar{P}_j \hat{\phi} \right) = (q_j p_{j,3} - \bar{P}_j) \hat{\phi}, \quad (3)$$

where \bar{P}_j is the matrix formed by the first two rows of P_j , while $p_{j,3}$ corresponds to the third row of P_j :

$$P_j = \begin{bmatrix} \bar{P}_j \\ p_{j,3} \end{bmatrix}.$$

Then, the functional Θ computed at $\hat{\phi}$ becomes:

$$\Theta(\hat{\phi}) = \mathbf{e}'^\top W' \mathbf{e}' = \hat{\phi}^\top A \hat{\phi}.$$

where $\mathbf{e}' = [e'_1{}^\top \ e'_2{}^\top \ \dots \ e'_m{}^\top]^\top$, $W' = D W D$, $D = \text{diag}(d_1^{-1}, d_2^{-1}, \dots, d_m^{-1})$ is the diagonal matrix formed by the inverse of the values of the distances $\{d_j\}$, and A and F_j are defined as follows:

$$\begin{aligned} A &= \mathbf{F}^\top W' \mathbf{F}, \\ F_j &= (q_j p_{j,3} - \bar{P}_j), \quad j = 1, \dots, m, \end{aligned}$$

with $\mathbf{F} = [F_1^\top \ F_2^\top \ \dots \ F_m^\top]^\top$.

A. Linear-Eigen method

The general solution $\hat{\phi}_{LE}$, which minimizes the functional Θ , can be obtained as the eigenvector of A associated to its minimum eigenvalue. Noticeably, this is equivalent to computing the principal component associated to the minimum singular value of B :

$$B = W^{1/2} D [F_1^\top \quad F_2^\top \quad \dots \quad F_m^\top]^\top ,$$

where $W^{1/2}$ is such that $W = W^{1/2} W^{1/2}$.

Since vectors are expressed in homogeneous coordinates, the last component $\hat{\phi}_{LE}(4)$ of the obtained solution $\hat{\phi}_{LE}$ has to be normalized to 1: $\hat{\phi}_{LE}$ is then redefined as $\hat{\phi}_{LE} = \hat{\phi}_{LE} / \hat{\phi}_{LE}(4)$.

The value $\hat{\phi}_{LE}$ just computed using $d_j = 1, \forall j$, corresponds to the solution of the *Linear-Eigen* (LE) method [2].

B. Linear-LS method

Since typically motion capture systems are used in closed areas, the case of targets at very large distance (at limit infinite) from the cameras is quite uncommon. Excluding the case of points at infinity, it is possible to write $\hat{\phi}$ as follows: $\hat{\phi} = [\bar{\phi}^\top \quad 1]^\top$, where $\bar{\phi} = [\hat{x} \quad \hat{y} \quad \hat{z}]^\top$.

The functional Θ results: $\hat{\Theta}(\phi) = [\bar{\phi}^\top \quad 1] A \begin{bmatrix} \bar{\phi} \\ 1 \end{bmatrix}$, where $A = \begin{bmatrix} \Lambda & a \\ a^\top & b \end{bmatrix}$, Λ is a 3×3 matrix, a is a 3×1 column vector, and b is a scalar. By simple matrix manipulations, Θ can be rewritten as follows: $\Theta = \bar{\phi}^\top \Lambda \bar{\phi} + a^\top \bar{\phi} + \bar{\phi}^\top a + b$.

The above equation is a quadratic function, whose minimum can be obtained by imposing the first partial derivative to be zero, i.e.: $\Lambda \bar{\phi} + a = 0$, and finally, $\bar{\phi} = -\Lambda^\dagger a$, where Λ^\dagger is the pseudo-inverse of Λ .

Computing the above estimate $\bar{\phi}$ with $d_j = 1, \forall j$ corresponds to the solution $\hat{\phi}_{LLS}$ of the *Linear-LS* (LLS) method [2].

C. Optimal L_2

If the correct values of $d_j, \forall j$ are known, then the estimate $\bar{\phi}$ computed above corresponds to the optimal solution $\hat{\phi}_{opt}$ of the triangulation problem with L_2 norm.

However, if any other information¹ is not available to the algorithm, the correct values of $d_j, \forall j$ are usually not known.

D. Algebraic method

The *Algebraic* method aims at estimating both ϕ and $d_j, \forall j$: From (3), it follows that:

$$e'_j = (d_j q_j - \bar{P}_j \hat{\phi}) = [-\bar{P}_j \mid q_j] \begin{bmatrix} \hat{\phi} \\ d_j \end{bmatrix} \approx 0 ,$$

and $e' = E [\hat{\phi}^\top \mid d_1^\top \quad d_2^\top \quad \dots \quad d_m^\top]^\top \approx 0$, where

$$E = \begin{bmatrix} -\bar{P}_1 & \left| \begin{array}{ccc} q_1 & 0 & \dots \\ 0 & q_2 & 0 & \dots \end{array} \right. \\ \vdots & \left| \begin{array}{ccc} \vdots & & \\ 0 & 0 & \dots & q_m \end{array} \right. \\ -\bar{P}_m & \left| \begin{array}{ccc} & & \end{array} \right. \end{bmatrix} .$$

Similarly to the LE case, the solution vector $\hat{\phi}_{AL}$ is obtained as the eigenvector corresponding to the smallest singular value of E . Then, the last component $\hat{\phi}_{AL}(4)$ of $\hat{\phi}_{AL}$ has to be normalized to 1: $\hat{\phi}_{AL}$ is then redefined as follows $\hat{\phi}_{AL} = \hat{\phi}_{AL} / \hat{\phi}_{AL}(4)$.

E. Iterative methods

Actually, both the Linear-Eigen and the Linear-LS methods can be applied iteratively. At each iteration the previous solution is used to obtain estimates $\hat{d}_j, \forall j$ of the values of $d_j, \forall j$. The rationale is that by using values of $\hat{d}_j, \forall j$ close to their correct values, the new solution is supposed to be close to $\hat{\phi}_{opt}$. This case is named *iterative Linear-Eigen* (or *iterative Linear-LS*, respectively) method.

Unfortunately, the iterative method (both using Linear-Eigen or Linear-LS) explained above does not correspond to a convex problem, thus the algorithm may eventually fall in a local minimum. However, as shown next in Section III, the results obtained with the iterative method are usually very close to the optimal one.

¹In some applications more geometrical information about the tracked objects can be available, e.g. the area of the target.

III. COMPARISON OF TRIANGULATION ALGORITHMS

In this section the triangulation methods reported previously are compared on three case studies:

- case I: randomly sampled points in target domain \mathcal{D} ;
- case II: points distributed close to the epipolar line connecting the optical centers of two cameras;
- case III: points much closer to one or two cameras than to the others.

The aim of case I is to simulate the methods in their common conditions of use, instead case II and III refer to some particular, but possibly frequent, practical conditions.

The following methods are compared: the optimal L_2 (L_2 Opt), LE, LLS, iterative LE (2 and 10 iterations: LE_2 and LE_{10}), iterative LSS (2 and 10 iterations: LSS_2 and LSS_{10}), and the algebraic method. Furthermore, the behavior of the methods for different number of cameras is studied. The results are reported for the following values of m : $\{2, 3, 64\}$ cameras. The cameras are positioned, at the same altitude (5 meter), on a circle as shown in Fig. 2. The ray of the circle is 8 meter, and the domain of targets in case I is the cube (each side is 10 meter long) centered at the circle's center. When $m = 2$ triangulation is obtained using the two cameras in red in Fig. 2. When $m = 3$ also another camera is used: That in green in Fig. 2 in case I and II, while one among the black and the white one in case III. Finally, when $m = 64$ all the cameras are used.

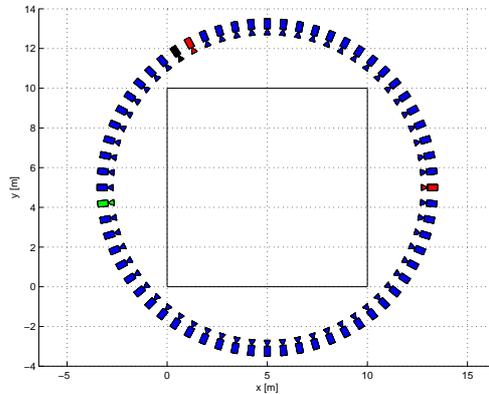


Fig. 2. 64 cameras positioned on a circle. Triangulation methods are compared using 2 (the red ones), 3 (those in red and one among the green, white and black camera), and 64 cameras.

In each considered condition, the following results are reported: the root mean square reconstruction error (RMSE) and the maximum reconstruction error (max) with respect to the correct target position, the root mean square reconstruction error ($RMSE_{Opt}$) and the maximum reconstruction error (max_{Opt}) both with respect to the optimal L_2 reconstruction.

In all the considered cases the measurements are affected by a zero-mean (white) Gaussian noise of covariance $\text{diag}(1, 1)$ pixel, where $\text{diag}(1, 1)$ is the 2×2 diagonal matrix with $(1, 1)$ on the diagonal. This can be considered as an intermediate amount of noise (for high quality motion capture systems it is a large amount of noise).

A. Case I: random points

In this case 100 points are randomly sampled (from a uniform distribution) in all the target domain. 100 reconstructions of each point are obtained from different noisy measurements of the cameras. The results reported in Table I shows that in this case all methods obtain quite good results: iterative methods after 2 steps practically converge to the optimal L_2 solution, but LE and LLS results are quite good too. The results of the algebraic method are acceptable but worse with respect to those of the other methods.

B. Case II: almost singular conditions

In this subsection 50 points randomly sampled along the epipole line between cameras in red in Fig. 2 are considered. 100 reconstructions of each point are obtained from different noisy measurements of the cameras. The results are reported in Table II. This example represents a geometrical singularity condition for the triangulation problem with 2 cameras. So, as expected, all methods cannot provide any useful solution using 2 cameras. Nevertheless, using 3 non aligned cameras the results LLS (and iterative LLS) method are good, and the use of 64 cameras can improve further the results of LLS algorithm. Instead the LE method proves to be unstable in this conditions. Since in non unstable conditions its results are practically the same of LLS (see case I), hereafter only the LLS method is considered. Finally, the algebraic method provides fair results only when using 64 cameras.

TABLE I

2 cameras				
Method	RMSE[mm]	max[mm]	RMSE _{Opt} [μ m]	max _{Opt} [μ m]
L_2 Opt	2.20	4.26	0	0
LE	2.21	4.15	507	916
LLS	2.21	4.15	507	916
LE ₂	2.20	4.26	0.687	1.81
LLS ₂	2.20	4.26	0.686	1.87
LE ₁₀	2.20	4.26	0.687	1.81
LLS ₁₀	2.20	4.26	0.686	1.87
Algebraic	3.60	6.43	2110	3800
3 cameras				
Method	RMSE[mm]	max[mm]	RMSE _{Opt} [μ m]	max _{Opt} [μ m]
L_2 Opt	1.67	3.25	0	0
LE	2.68	4.18	1640	3010
LLS	2.68	4.18	1640	3010
LE ₂	1.67	3.25	1.19	2.71
LLS ₂	1.67	3.25	1.08	2.54
LE ₁₀	1.67	3.25	0.785	1.93
LLS ₁₀	1.67	3.25	0.691	1.88
Algebraic	226	489	226000	489000
64 cameras				
Method	RMSE[mm]	max[mm]	RMSE _{Opt} [μ m]	max _{Opt} [μ m]
L_2 Opt	1.22	1.40	0	0
LE	2.21	2.58	1536	1802
LLS	2.21	2.58	1536	1802
LE ₂	1.22	1.40	0.958	1.45
LLS ₂	1.22	1.40	0.701	1.15
LE ₁₀	1.22	1.40	0.326	0.537
LLS ₁₀	1.22	1.40	0.138	0.300
Algebraic	14.6	19.6	15000	20000

C. Case III: different distances

This is an example of a potentially critical condition for the iterative methods. When the distances between cameras and target are very different the LS solution may be far from the correct target position. Consequently, iterative LLS may converge on a local minimum. In the 3 cameras case, 1 camera is far from the target while 2 are close to it. As shown by the results reported in Table III, actually the iterative LLS converged very close to the optimal L_2 solution in practical all the cases. Differently, the algebraic method is not robust to different distances between cameras and target. This result is in analogy with a previous example provided in [4].

D. Remarks

While the LLS method provides very reliable results in all the considered (non-singular) examples, the algebraic method is very unstable in most of the considered conditions. The LE method often provides results very similar to the LLS method, which however is typically computationally more stable. The iterative LLS method successfully exploits the initial LLS solution converging in all the considered cases very close to the optimal L_2 solution. Furthermore, LLS₂, the solution of the second iteration, is often already fair. Moreover, as expected, increasing the number of cameras, the triangulation error with respect to the correct target position decreases consequently.

Motivated by the above considerations, we suggest that the iterative LLS is a good candidate for real scenario applications. In the following section the adopted triangulation method, iterative LLS, is assumed to practically always converge to the optimal L_2 solution.

IV. CONDITIONS FOR ADEQUATE RECONSTRUCTION

A typical design requirement for a motion capture system is that of reconstructing targets' positions with a certain accuracy (usually in terms of Euclidean distance of the reconstructed point with respect to its correct position). Then, the aim of this section is that of providing conditions for an *adequate reconstruction*, e.g. how many cameras have to be used to make sure that the reconstruction error's standard deviation is lower than a given threshold ϵ .

As shown in the previous section, optimal positioning can be practically obtained using a fast linear triangulation method, the iterative LLS: After a few steps, the LLS reaches the optimal solution in almost all the conditions of practical interest in this framework. Then, hereafter the position of a point reconstructed by some cameras will be assumed to be the optimal (in L_2 norm sense) position.

Let camera measurements be affected by an additive zero mean white Gaussian noise of covariance $\sigma_e^2 I$, and consider a target reconstructed by a set of cameras including camera j . Then the information about the target position ϕ provided by

TABLE II

2 cameras				
Method	RMSE[m]	max[m]	RMSE _{Opt} [m]	max _{Opt} [m]
L_2 Opt	2.91	13.2	0	0
LLS	2.91	13.2	0.70	2.93
LLS ₂	2.91	13.2	1.14	6.05
LLS ₁₀	2.91	13.2	1.14	6.05
Algebraic	2.32	5.17	2250	14300
3 cameras				
Method	RMSE[mm]	max[mm]	RMSE _{Opt} [μ m]	max _{Opt} [μ m]
L_2 Opt	4.11	7.36	0	0
LLS	3.92	7.23	1090	1610
LLS ₂	4.11	7.36	2.06	5.01
LLS ₁₀	4.11	7.36	2.48	6.04
Algebraic	272	604	$2.71 \cdot 10^5$	$6.00 \cdot 10^5$
64 cameras				
Method	RMSE[mm]	max[mm]	RMSE _{Opt} [μ m]	max _{Opt} [μ m]
L_2 Opt	1.08	1.29	0	0
LLS	2.34	2.71	1780	2140
LLS ₂	1.08	1.29	0.793	1.20
LLS ₁₀	1.08	1.29	0.0906	0.204
Algebraic	17.1	44.5	17000	44400

TABLE III

2 cameras				
Method	RMSE[mm]	max[mm]	RMSE _{Opt} [μ m]	max _{Opt} [μ m]
L_2 Opt	2.17	5.16	0	0
LLS	2.82	5.47	1650	2820
LLS ₂	2.17	5.16	0.0254	0.0543
LLS ₁₀	2.17	5.16	0.0254	0.0543
Algebraic	11.3	18.2	12100	20600
3 cameras				
Method	RMSE[mm]	max[mm]	RMSE _{Opt} [μ m]	max _{Opt} [μ m]
L_2 Opt	1.84	4.39	0	0
LLS	2.97	5.21	2180	3700
LLS ₂	1.84	4.39	0.0518	0.114
LLS ₁₀	1.84	4.39	0.0435	0.102
Algebraic	105	330	106000	330000
64 cameras				
Method	RMSE[mm]	max[mm]	RMSE _{Opt} [μ m]	max _{Opt} [μ m]
L_2 Opt	0.104	0.137	0	0
LLS	2.53	3.01	2470	2930
LLS ₂	0.103	0.137	0.102	0.153
LLS ₁₀	0.104	0.137	0.004	0.011
Algebraic	18.9	27.9	18900	27900

camera j can be modeled as follows: $\hat{\phi}_j \sim \mathcal{N}(\phi, \Sigma_j(\phi))$. The variance Σ_j depends on ϕ and on the camera characteristics as follows: $\Sigma_j = \gamma \psi_j \psi_j^\top + \sigma_e^2 d_j^2 \bar{\Psi}_j \bar{\Psi}_j^\top$, where ψ_j is the unit vector of direction $\phi - O_j$, $\bar{\Psi}_j$ is an orthonormal basis of the plane Π_j , and γ is much larger than $\sigma_e^2 d_j^2$. Then, the first principal component of Σ_j practically coincides with ψ_j and has singular value γ . Then Σ_j admits the following PCA representation:

$$\Sigma_j \approx [\psi_j \quad \Psi_j] \text{diag}(\gamma, \sigma_1, \sigma_2) \begin{bmatrix} \psi_j^\top \\ \Psi_j^\top \end{bmatrix}, \quad (4)$$

where $[\psi_j \quad \Psi_j]$ is a unitary 3×3 matrix, and typically $\gamma \gg \sigma_1 \geq \sigma_2 \geq 0$. Notice that $\Psi_j = \bar{\Psi}_j$ only for ϕ positioned on the optical axis of camera j . Then, the target position reconstructed using m cameras' measurements is: $\hat{\phi} \sim \mathcal{N}(\phi, \Sigma_{\hat{\phi}}(\phi))$, where $\Sigma_{\hat{\phi}}(\phi) = \text{inv} \left(\sum_{j=1}^m \Sigma_j^{-1} \right)$. Notice that $\Sigma_{\hat{\phi}}(\phi)$ depends on the point ϕ at which it is evaluated.

Substituting (4) in the above equation: $\Sigma_{\hat{\phi}}(\phi) = \text{inv} \left(\sum_{j=1}^m [\psi_j \quad \Psi_j] \text{inv}(\text{diag}(\gamma, \sigma_1, \sigma_2)) \begin{bmatrix} \psi_j^\top \\ \Psi_j^\top \end{bmatrix} \right)$.

Since $\gamma^{-1} \approx 0$, if ϕ is a properly reconstructed point (in a non-singular configuration, i.e. not all cameras and target aligned): $\Sigma_{\hat{\phi}}(\phi) \approx \text{inv} \left(\sum_{j=1}^m \Psi_j \text{inv}(\text{diag}(\sigma_1, \sigma_2)) \Psi_j^\top \right)$.

The above equation is a very good approximation of the uncertainty in the reconstructed position of target ϕ . The reader is referred to [7] for a detailed experimental validation of such approximation. Then, the goal of the motion capture system

can be formulated as follows:

$$\sqrt{\text{trace}(\Sigma_{\hat{\phi}}(\phi))} < \epsilon, \quad \forall \phi \in \mathcal{D}, \quad (5)$$

where \mathcal{D} is the targets' domain. Since checking the above condition over the entire domain \mathcal{D} can be quite laborious, an upper bound of $\sqrt{\text{trace}(\Sigma_{\hat{\phi}}(\phi))}$ will be derived in the following for some configurations of interest.

Let \bar{d} be the maximum feasible value of d_j , for all j . The value of \bar{d} is typically imposed by the room size or by the camera's maximum visibility distance, i.e. the maximum distance at which a target can be detected by the camera.

Considering the worst case², $\sigma_e^2 \bar{d}^2 \geq \sigma_1 \geq \sigma_2$, and $\Sigma'_{\hat{\phi}}$ is used instead of $\Sigma_{\hat{\phi}}$: $\Sigma'_{\hat{\phi}}(\phi) = \sigma_e^2 \bar{d}^2 \left(\sum_{j=1}^m \Psi_j \Psi_j^\top \right)^{-1}$. Let $\lambda_k(\Xi)$ indicate the k -th eigenvalue of a generic matrix Ξ . Furthermore, for each positive (or negative) definite matrix Ξ let the eigenvalues be ordered in increasing order, i.e. $\lambda_1(\Xi)$ is the smallest eigenvalue of Ξ . Then $\text{trace}(\Sigma'_{\hat{\phi}}(\phi))$ can be computed as follows: $\text{trace}(\Sigma'_{\hat{\phi}}(\phi)) = \sigma_e^2 \bar{d}^2 \sum_k \frac{1}{\lambda_k(\sum_{j=1}^m \Psi_j \Psi_j^\top)}$, where such expression has been obtained noticing that $\{\lambda_k(\Xi)\}_{k=1:3} = \{1/\lambda_k(\Xi^{-1})\}_{k=1:3}$ for a 3×3 positive definite matrix Ξ [5]. Since $[\psi_j \quad \Psi_j]$ is unitary, then $\psi_j \psi_j^\top + \Psi_j \Psi_j^\top = I$, and hence: $\text{trace}(\Sigma'_{\hat{\phi}}(\phi)) = \sigma_e^2 \bar{d}^2 \sum_k \frac{1}{m - \lambda_k(\sum_{j=1}^m \psi_j \psi_j^\top)}$. Defining $M(\phi) = \sum_{j=1}^m \psi_j \psi_j^\top$, then:

$$\text{trace}(\Sigma'_{\hat{\phi}}(\phi)) = \sigma_e^2 \bar{d}^2 \sum_k \frac{1}{m - \lambda_k(M(\phi))} \quad (6)$$

$$\leq \sigma_e^2 \bar{d}^2 \frac{3}{m - \lambda_3(M(\phi))}. \quad (7)$$

Since $\{\psi_j\}$ are unit vectors, then $\sum_{k=1}^m \lambda_k(M(\phi)) = m$.

Example 1: Let cameras be placed along a circle of ray r (similarly to Fig. 2). Cameras are equally spaced along such circle. Without loss of generality the camera circle is assumed to be centered on the origin of the Cartesian axes. This example aims at computing the value of $\text{trace}(\Sigma'_{\hat{\phi}})$ on the origin O . By construction: $\psi_j = \frac{\phi - O_j}{|\phi - O_j|} = \frac{-O_j}{|O_j|} = \frac{-O_j}{r}$.

Then $\lambda_k(M(O)) = \frac{1}{r^2} \lambda_k(\sum_{j=1}^m O_j O_j^\top)$. Since m cameras are distributed uniformly along a circle of ray r , then the eigenvalues of $\sum_{j=1}^m O_j O_j^\top$ are $\{\frac{mr^2}{2}, \frac{mr^2}{2}, 0\}$, where the first two eigenvalues are associated to two orthogonal directions on the plane containing the circle. Hence, $\text{trace}(\Sigma'_{\hat{\phi}}(O)) = \sigma_e^2 \bar{d}^2 (\frac{2}{m} + \frac{2}{m} + \frac{1}{m}) = \sigma_e^2 \bar{d}^2 \frac{5}{m}$. \square

In the following subsections some bounds on $\text{trace}(\Sigma'_{\hat{\phi}})$ (for all points in \mathcal{D}) will be computed assuming that the cameras are positioned as in the above example.

A. \mathcal{D} as a small spherical domain

In this subsection the minimum possible distance of a target from the cameras is set to l . Let Λ be a semi-positive definite matrix and x be a vector, if $\beta_1 \leq \beta_2$ then $(\Lambda + \frac{1}{\beta_1} x x^\top) - (\Lambda + \frac{1}{\beta_2} x x^\top) \geq 0$ and thus $\lambda_k(\Lambda + \frac{1}{\beta_1} x x^\top) \geq \lambda_k(\Lambda + \frac{1}{\beta_2} x x^\top)$. Let $\eta_j = \phi - O_j$, then $\lambda_k(\sum_j \frac{\eta_j \eta_j^\top}{|\eta_j|^2}) \leq \frac{1}{l^2} \lambda_k(\sum_j \eta_j \eta_j^\top)$, and from (6): $\text{trace}(\Sigma'_{\hat{\phi}}(\phi)) \leq \sum_k \frac{\sigma_e^2 \bar{d}^2}{m - \frac{1}{l^2} \lambda_k(\sum_j (\phi - O_j)(\phi - O_j)^\top)}$. Because of the symmetric camera configuration $\sum_{j=1}^m O_j = 0$, thus:

$$\text{trace}(\Sigma'_{\hat{\phi}}(\phi)) \leq \sum_k \frac{\sigma_e^2 \bar{d}^2}{m - \frac{1}{l^2} \lambda_k(m\phi\phi^\top + \sum_{j=1}^m O_j O_j^\top)}.$$

$m\phi\phi^\top$ and $\sum_{j=1}^m O_j O_j^\top$ have the following sets of eigenvalues: $\{m|\phi|^2, 0, 0\}$, and (as shown in Example 1) $\{\frac{mr^2}{2}, \frac{mr^2}{2}, 0\}$, respectively. Finally, taking into account of bounds on the eigenvalues of a sum of matrices [5]:

$$\text{trace}(\Sigma'_{\hat{\phi}}(\phi)) \leq \frac{3\sigma_e^2 \bar{d}^2}{m \left(1 - \frac{r^2/2 + (r-l)^2}{l^2}\right)}. \quad (8)$$

Notice that the above bound can be used only if the targets are close enough to the origin (i.e. far enough from cameras).

B. \mathcal{D} as a planar domain

In this subsection the domain \mathcal{D}_p is the 2D region limited by the camera circle. Nevertheless, because of measurement noise, the reconstructed points are 3D points not restricted to lay on such 2D region, even if typically they are close to it.

Since $\{\psi_j\}$ are unit vectors, then, because of the symmetry of the configuration, the maximum value of $\lambda_3(M(\phi))$ with respect to $\phi \in \mathcal{D}_p$ is $\lambda_3(M(\phi)) = \max(\lambda_3(M(O)), \lambda_3(M(O_j)))$, where $\lambda_3(M(O_j)) = \lambda_3(M(O_{j'})) \quad \forall j' \neq j$. Notice that when $\phi = O_j$ then $M(\phi) = \sum_{j' \neq j} \psi_{j'} \psi_{j'}^\top$.

²In $\Sigma'_{\hat{\phi}}$ the noise level is set at its maximum value for all points in \mathcal{D} and for all cameras, i.e. the information actually provided by each camera about the target position is always greater or equal to that used in $\Sigma'_{\hat{\phi}}$.

Considering $O_1 = [r \ 0 \ 0]^T$, then by geometric considerations the first principal component has to be aligned with the horizontal axis, and the value of the corresponding singular value (which is equal to $\lambda_3(M(O_1))$) is:

$$\lambda_3(M(O_1)) = \sum_{h=1}^{m-1} \frac{1}{2}(1 - \cos(2\pi h/m)) = m/2 .$$

Substituting the above expression in (7), then: $\text{trace} \left(\Sigma'_{\hat{\phi}}(\phi) \right) \leq \sigma_e^2 \bar{d}^2 \frac{6}{m} .$

C. \mathcal{D} as a semi-spherical domain

Let \mathcal{D}_s be the semi-sphere of ray r centered in the origin and with positive z coordinate.

From geometrical considerations (aiming at maximizing the variance captured by the first principal component of $M(\phi)$), the maximum value of $\lambda_3(M(\phi))$ with respect to $\phi \in \mathcal{D}_s$ is $\lambda_3(M(\phi)) = \max(\lambda_3(M(O)), \lambda_3(M([0 \ 0 \ r]^T)))$.

The value of the maximum eigenvalue in correspondence of ϕ on the top of the semi-sphere, $\phi = [0 \ 0 \ r]^T$. Because of the symmetry of the configuration the maximum principal component is aligned with the vertical axis, and the value of its associated singular value can be computed as follows: since $\psi_h = [-\cos(2\pi h/m) \ -\sin(2\pi h/m) \ r]^T / \sqrt{2}r$, then (sum of the squares of projections of $\{\psi_h\}$ on the principal component vector), $\lambda_3(M([0 \ 0 \ r]^T)) = \sum_{h=0}^{m-1} \left(\frac{r}{\sqrt{2}r} \right)^2 = m/2$. Substituting this expression in (7), then:

$$\text{trace} \left(\Sigma'_{\hat{\phi}}(\phi) \right) \leq \sigma_e^2 \bar{d}^2 \frac{6}{m} . \quad (9)$$

Notice that this result can be easily extended also to the spherical case.

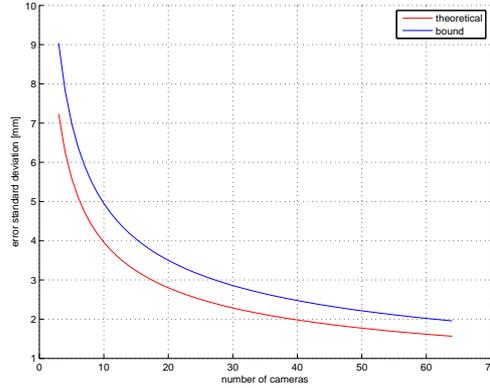


Fig. 3. Maximum value of the standard deviation of the reconstructed error varying the number of cameras placed in circle (Fig. 2). Maximum standard deviation value (red) and its upper bound (blue) computed as in (9).

D. Remarks

Interestingly, the computed bounds decrease with $1/m$ (similarly to the common variance of a mean estimate). Thus, once that an upper bound for $\text{trace} \left(\Sigma'_{\hat{\phi}}(\phi) \right)$, $\forall \phi \in \mathcal{D}$, has been obtained, then it is possible to compute through (5) an upper bound to the minimum number of cameras \hat{m} necessary to have the error standard deviation lower than ϵ . Since (8) can be applied only when the targets are quite close to the origin, (9) practically results to be more useful.

Fig. 3 compares the bound obtained through (9) with the correct values of the error's standard deviation when cameras are disposed on a circle of ray 8 m (as in Fig. 2) and the visibility range of each camera is 10 m. As shown in Fig. 3, both the real variance and the computed upper bound decrease approximatively as $O(1/m)$. Since the constant factors in the $O(1/m)$ notation are different for the upper bound and the correct variance, then the discrepancy between the computed upper bound \hat{m} on the number of necessary cameras and the number of cameras really needed for a certain error level ϵ increases as ϵ becomes smaller.

V. CONCLUSIONS

In the first part of this paper some linear algorithms for geometric triangulation (minimizing the L_2 norm) have been resumed and compared on three cases of interest. Differently from the other linear methods, the iterative LLS method has provided reliable results in all the considered conditions: Actually the solutions obtained (in non-singular geometric conditions) after few steps of the iterative LLS were very close to the correct targets positions.

In the second part of the paper a Gaussian approximation of the reconstruction error has been introduced. The standard deviation of the reconstruction error on a point $\hat{\phi}$ has been explicitly expressed exploiting such approximation. Theoretical upper bounds on the reconstruction error variance on all the target domain have been derived in a possible configuration of the motion capture systems. Finally, the upper bound to the reconstruction error allows to obtain an upper bound on the number of cameras necessary to reconstruct the targets on all the domain with a user defined accuracy.

REFERENCES

- [1] R. Fisher, "CVonline: The Evolving, Distributed, Non-Proprietary, On-Line Compendium of Computer Vision", <http://homepages.inf.ed.ac.uk/rbf/CVonline/>
- [2] R.I. Hartley and P. Sturm, "Triangulation", *Computer Vision and Image Understanding*, vol. 68(2), 1997, pp 146-157.
- [3] R.I. Hartley and F. Kahl, "Optimal Algorithms in Multiview Geometry", *Proc. of the 8th Asian Conf. on Computer Vision*, 2007, pp 13-34.
- [4] F. Kahl and R. Hartley, "Multiple-View Geometry under the L_∞ -Norm", *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(9):1603-1617, Sept. 2008.
- [5] R.A. Horn and C.R. Johnson, "Matrix analysis", Cambridge University Press, 1985.
- [6] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, "An Invitation to 3-D Vision: From Images to Geometric Models", *Interdisciplinary Applied Mathematics: Imaging, Vision, and Graphics*, Vol. 26, Springer, 2004.
- [7] A. Masiero and A. Cenedese, "Reconstruction error in a motion capture system", *arXiv:1203.3230v1*, 2012.
- [8] H. Stewenius, F. Schaffalitzky and D. Nister, "How hard is 3-view triangulation really?", *Proc. of the 10th ICCV*, 2005, pp 686-693.
- [9] B. Triggs, P. McLauchlan, R. Hartley and A. Fitzgibbon, "Bundle Adjustment - A modern Synthesis", *Springer Lecture Notes on Computer Science*, Springer Verlag, 2000.
- [10] A. Willig, "Wireless sensor networks: concept, challenges and approaches", *Elektrotechnik und Informationstechnik*, 2006.