On the Applicability of Several Methods for Camera Calibration

Tomislav Pribanić, Peter Sturm, Mario Cifrek

ON THE APPLICABILITY OF SEVERAL METHODS FOR CAMERA CALIBRATION

Tomislav Pribanic*, Peter Sturm** and Mario Cifrek*

*Faculty of electrical engineering and computing, University of Zagreb, Unska 3, HR-10000
**INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 Montbonnot St Martin, France

tomislav.pribanic@fer.hr

Abstract: This paper presents four alternative ways of initializing camera parameters using essentially the same calibration tools (orthogonal wands) as nowadays popular 3D kinematic systems do. However, the key idea presented here is to sweep the volume with an orthogonal pair or triad of wands instead of a single one. The proposed methods exploit the orthogonality of the used wands and set up familiar linear constraints on certain entities of projective geometry. Extracted initial camera parameters values are closer to the refined ones, which should generally assure faster and safer convergence during the refinement procedure. Even without refinement, sometimes not necessary, reconstruction results using our initial sets are better than using commonly obtained initial values. Besides, the entire calibration procedure is shortened since the usual two calibration phases become one.

Introduction

Using so called 3D kinematic systems is one convenient way for biomechanical analysis [1] since, generally, it does not constrain subject movement. It is based on processing images acquired by cameras [2]. A necessary step before the actual 3D reconstruction is camera calibration. The process of camera calibration came across many stages of improvement during the last few decades. One aspect of improvement is witnessed by the particular calibration tool that a user normally uses to calibrate cameras of a 3D kinematic system. It went from, traditionally, manipulation of cumbersome 3D cages to nowadays sweeping the volume with only a single wand of known length (so called wand dance). Prior to the wand dance part usually two (three) orthogonal wands (with some markers) are placed within the calibration volume (Figure 1), mainly for two reasons: to determine a spatial coordinate system according to user needs and, most likely, to initialize camera parameters, which are later on refined as a result of the wand dance. The closeness of the initial solution to the final/refined one highly determines speed of convergence and in a large number of cases determines whether there will be any convergence or none at all [4]. This paper investigates alternative ways to initialize camera parameters, using a presently popular calibration tool – a set of orthogonal wands.

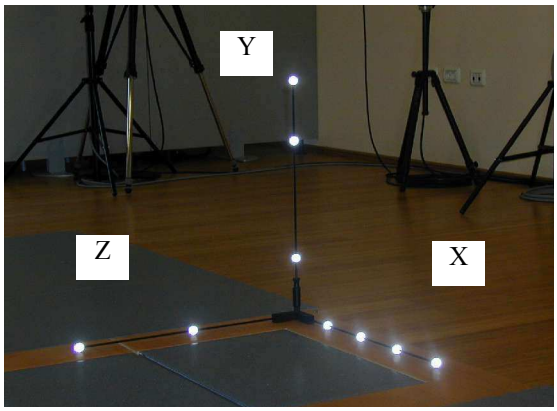

Figure 1. Image of the orthogonal triad with attached markers. Markers distances [cm] with respect to the triad origin are: X-axis 15, 30, 45, 60; Y-axis 15, 45, 60; Z-axis 30, 60.

Equipment and Methods

Camera calibration and/or metric (Euclidean) reconstruction is closely related to certain geometric entities from projective geometry [3], such as lines at infinity, circular points (2D space) and plane at infinity, absolute conic, absolute dual quadric (3D space). The idea of the proposed methods is to sweep the calibration volume (wand dance) with two or three orthogonal wands instead of one wand only. In fact, such an approach effectively eliminates the need to also undertake the usual calibration phase of placing on the floor and imaging a triad of orthogonal wands (Figure 1), prior to the wand dance itself. The proposed approach will allow us to set up constraints on some of the mentioned geometric entities, compute them and ultimately obtain the camera parameters from them. Four different calibration methods were investigated.

Method 1a

1. Perform the wand dance using two orthogonal wands, each having at least two (three) markers whose relative distances are known.
2. For every frame of one camera, find the vanishing point of the wands' directions from the known distance ratios between the markers.
3. Use the constraint that wands are perpendicular to one another, to form an equation system on the image of the absolute conic (IAC) ω, built from equations of this type ($\mathbf{v}_1$ and $\mathbf{v}_2$ are vanishing points

associated with two perpendicular wands):

$$v_1^T \cdot \omega \cdot v_2 = 0$$

4. Solve the above overdetermined equation system and apply the Cholesky decomposition on the computed ω to retrieve the camera's internal parameters.
5. Repeat steps 2-4 for each camera.
6. Chose one camera as reference, form pairs with it and all other cameras, and then compute the fundamental matrix for each pair.
7. Knowing the internal camera parameters and fundamental matrices, compute essential matrices and decompose them to obtain the cameras' relative orientation (external parameters).
8. Having both internal and external camera parameters for a given camera pair, obtain the 3D metric reconstruction of the wand dance and calculate the wand lengths. Use the information about true wand lengths to compute the scale factor needed to transform the metric reconstruction to Euclidean.
9. Refine the above calculated initial cameras parameters enforcing the known wands lengths and/or orthogonality of wands. This step is left for future work and is given here only for completeness.

Method 1b

1. Perform the wand dance using two orthogonal wands, each with at least two markers and whose relative distances are known.
2. For every frame of one camera, compute the plane homography **H** between the plane in space (formed by the two wands) and the image plane.
3. Use **H** to find the images of the so-called circular points. Based on fact that these also lie on the IAC ω, form a system of linear equations of this type ($\mathbf{h}_1$ and $\mathbf{h}_2$ are the first and second column of **H**):

$$(h_1 \pm i \cdot h_2)^T \cdot \omega \cdot (h_1 \pm i \cdot h_2) = 0 \qquad (1)$$

4. Solve the overdetermined equation system and apply the Cholesky decomposition on the computed ω in order to retrieve the camera's internal parameters.
5. Repeat steps 2-4 for every camera.
6. Follow steps 6 to 9 described in Method 1a.

Method 2a

1. Perform the wand dance using the three orthogonal wands, each having at least two (three) markers with known relative distances.
2. Chose one camera as reference, form pairs with it and all other cameras and then compute the fundamental matrix for each pair. For every camera pair, compute the pair of canonical projection matrices $\mathbf{P}_1$ and $\mathbf{P}_2$ from the fundamental matrix:

$$P_1 = [I|0] \quad P_2 = \left[[e_2]_x + F + e_2 \cdot v^T | \lambda \cdot e_2 \right] \quad (2)$$

Here, $\mathbf{e}_2$ is the epipole of the second camera, **F** is the fundamental matrix, **v** and λ are an arbitrary vector and scalar, respectively. Using these projection matrices, compute a projective 3D reconstruction of the wand dance. This differs from the true one by an unknown projective transformation $\mathbf{H}_{4x4}$.

3. For every such 3D reconstruction compute the coordinates of the three planes formed by the wands of the orthogonal triad. The perpendicularity of these planes gives linear constraints on the absolute dual quadric $\mathbf{Q}_\infty$:

$$\pi_1^T \cdot Q_\infty \cdot \pi_2 = 0$$

where $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ are two orthogonal planes.

4. Solve the system of above equations, i.e. compute the absolute dual quadric $\mathbf{Q}_\infty$. Find the projective transformation $\mathbf{H}_{4x4}$ that will put the computed $\mathbf{Q}_\infty$, back into its canonical position. Apply the same $\mathbf{H}_{4x4}$ on the 3D projective reconstruction of wand dance to obtain a metric reconstruction of it. Similarly, apply the transformation $\mathbf{H}_{4x4}$ on the initially computed canonical projection matrices to obtain projection matrices coherent with the metric 3D reconstruction.
5. Use the information about true wand lengths to compute the scale factor needed to transform the metric reconstruction to Euclidean.
6. Repeat steps 3 to 5 for all camera pairs.
7. Follow step 9 described in Method 1a.

Method 2b

1. Perform the wand dance using two orthogonal wands, each having at least two (three) markers with known relative distances.
2. For every frame of one camera find the vanishing points of the wands directions using the known distance ratios (markers on wands).
3. Choose one camera as reference, form pairs with all other cameras and compute the associated fundamental matrices.
4. For some camera pair, compute the canonical projection matrices, like in step 2 of method 2a. Perform a projective 3D reconstruction of wand dance markers, wand positions and vanishing points (points at infinity). Based on the latter, compute the plane at infinity $\boldsymbol{\pi}_\infty$ as well and find the homography $\mathbf{H}_{4x4}$ that puts to its canonical position. Apply $\mathbf{H}_{4x4}$ on the reconstructed marker positions, to obtain an affine reconstruction of them.
5. Consider the fact that points at infinity are effectively representing directions of lines, in our case orthogonal lines. The perpendicularity of wands allows to form constraints on the absolute conic $\boldsymbol{\Omega}_\infty$:

$$V_1^T \cdot \Omega_\infty \cdot V_2 = 0$$

Here, $\mathbf{V}_1$ and $\mathbf{V}_2$ are positions in 3D affine space of vanishing points (points at infinity) of the orthogonal wand pair.

6. Solve the overdetermined system of above equations, i.e. compute the absolute conic $\boldsymbol{\Omega}_\infty$. Find the

projective transformation $\mathbf{H}_{4x4}$ that will put the computed $\boldsymbol{\Omega}_{\infty}$, back into its canonical position. Apply the same $\mathbf{H}_{4x4}$ on the 3D affine reconstruction of the wand dance which is in turn equivalent to obtaining a metric reconstruction of it. Similarly, apply both transformation matrices, from projective to affine and from affine to metric, on the initially computed canonical projective matrices to obtain projection matrices that correspond to a metric 3D reconstruction.

7. Use the information about true wand lengths to compute the scale factor needed to transform the metric reconstruction to Euclidean.
8. Repeat steps 4 to 7 for all camera pairs.
9. Follow step 9 described in Method 1a.

A popular commercially available system, Smart by eMotion company [5], was used to test the proposed methods for camera parameter initialization. The system version used (version 1.10, Build 2.39) consists of 9 cameras (50 Hz). It is a so-called optoelectronic system which actually reconstructs positions of passive retro-reflective markers, attached to the subject's points of interest. Markers are illuminated by stroboscopic IR sources of light attached to the cameras. The Smart system is installed in the Biomechanic laboratory of Peharec Polyclinic in Pula, Croatia [6]. Its software features offer the capability to export 2D image data of image sequences, initial camera parameters (obtained from putting the triad on the floor as explained in the introduction) and refined ones.

Results and Discussion

Let us first consider results for the initialization of internal parameters (Table 1, Table 2, Table 3, Table 4, Table 5 and Table 6) for all 9 cameras of the system. Table 2 shows internal parameters which are outputs of Smart's refinement routines. Although even those values are not perfectly accurate, it can be considered in the present context as the ones which are very close to the true ones, i.e. values the optimization should converge to. Hence, an initialization method which produces parameters closest to Smart's refined ones (Table 2) can be, generally speaking, considered as the method of choice. Namely, the closer we are, at the beginning of the iteration procedure, with initial values to expected refined (optimal) ones, the better chance we have to avoid all potential problems characteristic to convergence of refinement algorithms. A closer comparison of Smart's initialization values (Table 1) and initialization values of the proposed methods (Table 3, Table 4, Table 5 Table 6) with the Smart's refined ones (Table 2) clearly indicates that method 1a (Table 3) provides values very close to Smart's refined ones. Furthermore, it seems that the other three proposed methods (Table 4, Table 5 and Table 6) are about equally successful in providing estimates as close as possible to the refined ones.

Common to all proposed methods is that initial estimates are closer to Smart's refined values (Table 2) than Smart initial's ones (Table 1). The superiority of method 1a is demonstrated by the 3D reconstruction accuracy of wand lengths (Table 7), using a linear camera model for different camera pairs (36 possible pairs for 9 cameras). Table 7 shows that the reconstruction accuracy of method 1a is basically the same as the one using Smart's refined results (3rd and 4th column in Table 7). Besides let us bear in mind that Smart's refined projection matrices are optimized on the external parameters as well, which is definitely not the case for the shown algorithm (optimization of proposed methods is part of future work) and shown results of method 1a.

Table 1: Internal camera parameters. Initial values given by the Smart system.

Cam.	**Focal length [pixels]**		**Skew factor**	**Principal point [pixels]**	
	X	**Y**			
1	751,9	387,0	12,7	400,0	175,3
2	755,6	381,0	18,3	330,3	233,5
3	676,6	343,3	7,4	272,3	200,5
4	704,1	367,4	6,4	320,7	171,3
5	765,2	399,0	5,0	335,9	170,2
6	746,8	386,4	19,0	376,9	215,3
7	691,0	354,6	9,2	285,3	164,5
8	672,7	343,3	14,4	297,2	142,7
9	672,8	348,2	10,3	292,1	156,2

Table 2: Internal camera parameters. Final optimized values given by the Smart system.

Cam.	**Focal length [pixels]**		**Skew factor**	**Principal point [pixels]**	
	X	**Y**			
1	727,7	375,9	0,0	349,4	153,7
2	723,8	374,6	0,0	304,4	145,4
3	723,9	375,2	0,0	290,3	138,4
4	724,3	374,7	0,0	325,4	140,0
5	724,2	375,1	0,0	347,7	137,0
6	724,8	375,0	0,0	349,9	143,4
7	719,2	371,8	0,0	328,9	134,7
8	730,1	377,2	0,0	350,7	133,4
9	715,9	370,5	0,0	345,0	138,6

Table 3: Internal camera parameters. Initial values given by method 1a.

Cam.	**Focal length [pixels]**		**Skew factor**	**Principal point [pixels]**	
	X	Y			
1	724.82	376.17	0.02	376.90	127.97
2	749.74	390.04	0.67	321.94	135.95
3	743.38	390.71	-0.49	267.50	132.06
4	728.30	376.46	-0.25	346.85	133.46
5	715.60	373.75	-1.46	332.82	145.18
6	734.55	380.96	0.17	381.45	136.57
7	715.34	372.93	-2.26	274.41	126.25
8	704.35	370.23	0.87	307.96	122.16
9	723.27	373.40	0.48	341.93	136.22

Table 4: Internal camera parameters. Initial values given by method 1b.

Cam.	Focal length [pixels]		Skew factor	Principal point [pixels]	
	X	Y			
1	683.91	354.36	0	379.76	128.79
2	689.64	357.33	0	322.63	135.05
3	669.16	346.71	0	260.05	138.78
4	699.19	362.27	0	343.82	138.23
5	695.17	360.19	0	368.39	141
6	709.06	367.39	0	391.62	140.19
7	640.2	331.71	0	312.66	172.37
8	688.78	356.88	0	362.46	131.42
9	676.49	350.52	0	357.15	143.89

Table 5: Internal camera parameters. Initial values given by method 2a.

Cam.	Focal length [pixels]		Skew factor	Principal point [pixels]	
	X	Y			
1	702.17	360	10.12	354.64	118.81
2	769.14	400.99	3.24	303.5	137.59
3	725.01	373.57	5.33	249.41	165.19
4	729.18	376.74	1.24	359.99	132.02
5	792.97	406.88	20.98	331.1	123.5
6	709.51	371.12	16.55	382.62	139.25
7	717.49	371.11	0.81	229.15	122.72
8	747.16	386.62	5.24	327.04	128.71
9	805.36	415.67	14.58	309.63	136.89

Table 6: Internal camera parameters. Initial values given by method 2b.

Cam.	Focal length [pixels]		Skew factor	Principal point [pixels]	
	X	Y			
1	681.09	351.90	1.36	360.18	138.47
2	626.55	323.14	2.45	376.80	151.26
3	728.38	383.59	7.58	316.53	140.14
4	784.64	408.48	8.18	245.79	154.05
5	704.46	364.10	4.46	371.56	132.72
6	695.91	363.57	1.83	360.04	125.76
7	721.62	381.06	16.88	317.97	80.85
8	692.93	362.10	8.85	347.91	106.75
9	780.96	407.29	5.95	226.52	136.23

The reconstruction results of method 1b are not as good as for 1a (4th and 5th column in Table 7), however they are still better than with Smart's initial values (2nd column of Table 7). Methods 2a and 2b (last two columns of Table 7) perform worst among the proposed methods, primarily since for some camera pairs we end up with projection matrices which yield practically unacceptable reconstruction accuracy (shaded cells in Table 7). The logical question emerges immediately what could be the possible reason that method 1a performs that exceptionally well compared to other methods? Through numerous system calibration and studying distinct calculation steps of each method it is believed that the answer to the above question is largely of practical (implementation) nature. Namely, method 1b assumes the calculation of the planar homography **H** between a plane in space, defined by a pair of orthogonal wands, and the image plane. This requires a minimum of 4 pairs of non collinear points. In practice, due to noise, mainly caused by lens imperfection, the computation of **H** will be less impaired if we have large number of corresponding point pairs, presumably equally distributed in image planes.

Table 7: Mean error [mm] between reconstructed and true wand lengths; **SI** using Smart's initial data; **SF** using Smart's final optimized data; **1a**, **1b**, **2a** and **2b** using method 1a, 1b, 2a and 2b respectively.

Pair	SI	SF	1a	1b	2a	2b
12	29.26	9.54	14.27	11.99	6.68	32.43
13	15.97	7.74	10.36	7.33	13.05	5.16
14	14.59	6.54	4.53	7.84	10.37	21.68
15	14.01	6.17	5.23	8.03	25.84	4.71
16	19.83	8.81	6.81	6.74	9.43	6.87
17	12.17	10.74	4.20	13.04	17.02	11.17
18	14.03	6.61	5.43	7.57	795.66	6.65
19	13.53	9.45	5.31	10.02	36.37	670.53
23	20.79	6.20	7.73	9.06	15.92	15.85
24	17.36	4.48	4.79	6.72	4.51	15.03
25	18.26	4.38	4.50	8.37	23.46	24.19
26	25.11	6.14	8.75	7.18	5.99	12.08
27	15.45	8.62	4.58	16.80	23.55	15.71
28	17.12	4.47	6.44	7.75	15.20	135.56
29	35.36	6.55	5.37	8.06	634.62	16.68
34	16.20	4.96	3.61	7.20	4.26	9.95
35	38.67	13.40	5.73	11.71	210.52	9.99
36	23.71	6.45	7.72	6.17	12.41	6.54
37	15.29	11.20	11.11	52.41	7.46	6.63
38	15.77	5.43	6.22	9.82	13.50	5.15
39	13.69	5.60	3.65	8.50	33.26	11.22
45	15.62	4.29	5.10	5.94	7.98	13.92
46	24.66	5.86	5.59	8.21	10.66	15.64
47	14.28	7.17	3.57	15.72	4.27	29.03
48	17.74	5.85	6.02	6.67	4.12	19.29
49	12.42	4.68	3.47	7.46	8.26	19.98
56	23.89	5.44	5.15	8.26	10.17	4.32
57	19.00	11.97	5.38	14.68	101.77	17.02
58	14.70	4.45	6.10	7.04	13.18	12.00
59	9.56	4.34	6.67	10.08	9.04	16.56
67	20.03	8.63	4.83	9.16	10.40	17.98
68	23.76	6.93	5.76	7.28	12.03	5.23
69	20.07	5.51	4.21	9.17	31.55	19.32
78	14.42	7.85	5.95	11.12	11.29	10.51
79	11.23	7.86	3.68	9.24	12.22	13.07
89	15.20	4.54	11.80	7.35	6.45	17.16
Average	18.41	6.91	6.10	10.27	60.35 (19.4)[1]	35.41 (13.8)[1]

[1]Mean values considering no shaded cells.

Unfortunately, in our case neither a large number of point pairs nor their equal distribution is possible, because to compute **H** we use only the few markers available on our wands. Consequently, the images of circular points computed via **H** will not provide optimal sets of constraints for the system that we use to find the IAC. This issue comes particularly into effect in those frames where the apparent angle between wands, in the image plane, is small. Computing with such data will have a particularly bad effect on the final results and frequently even cause the IAC matrix $\boldsymbol{\omega}$ to not be positive definite and therefore prevent its Cholesky factorization. We have found empirically that for given Smart's cameras it proves to be beneficial to exclude all those frames where the apparent angle between wands in the image plane is less than 80°. This rather strict threshold causes that we end up with even less usable calibration data (redundancy) to form the final system of equations (1) for the IAC computation.

Different experiments revealed a positive effect if we enforce during the calculation the conditions of the skew factor being zero and of a known aspect ratio. Assumptions about knowledge of those two values are quite justified given the quality of the used cameras. At first we have tried out so-called soft constraints by adding two more equations to the already large overdetermined equation, gearing appropriate IAC elements towards their known value. As intuitively expected, soft constraining did not bring any significant improvement, since adding two more equations to the overdetermined system of several thousands of equations (solved by the least squares method) can hardly do much difference. Alternatively, one could assign different weights to the added equations, but that has not been tried out yet. Therefore we were left with the option of hard constraints where the equation system to compute IAC was set up right from the start to accommodate the fact of skew being zero and known aspect ratio. Hard constraining did bring an improvement in accuracy and thus results given for the method 1b (Table 4) explicitly shows a zero skew factor.

On the other hand, the desired equal distribution of points in the image plane is considerably easier to achieve with method 1a. Nevertheless, even here we face at least two potential problems: the possibility of the wand being (almost) perpendicular or parallel to the image plane during the imaging. Both problems backed up with image noise can seriously draw the solution in the wrong direction. The first occurrence is relatively easy to threshold out simply by discarding all those frames where the distance between wand end markers is less than some value. Similarly, the second problem could be taken care of by neglecting all those frames where the magnitude of vanishing points is larger than some value. However, in this case we have undertaken another strategy. The calculation starts by first including vanishing points around the image center and gradually considers more and more data around the image center in the calculation. For each given set of vanishing points data we end up eventually with one set of IAC matrices, i.e. the cameras' internal parameters. How to resolve the ambiguity which one to chose? Let us recall the fact that working with good quality cameras it is reasonable to assume a zero value for the skew parameter. Therefore we chose the set of internal parameters that has the smallest computed skew parameter, justifiably expecting that this set is very close to the true parameter values. At least close enough that an appropriate optimization routine, using the chosen set as initial one, would ultimately bring us to true values.

Methods 2a and 2b have a common feature: one of the calculation steps requires a 3D projective reconstruction based on a pair of canonical projection matrices, derived from the fundamental matrix. It can be shown that the pair of projection matrices uniquely determine the fundamental matrix. Unfortunately, for a given fundamental matrix we can find various pairs of projection matrices all corresponding to the same fundamental one. Various pairs of camera projection matrices represent various projective reconstructions which differ among themselves by the unknown spatial transformation **H**. Just as a reminder we are usually interested in finding pairs of projection matrices that correspond to a Euclidean reconstruction. Since solely based on the fundamental matrix, we cannot accomplish that aim directly and it is customary to start with the mentioned canonical pair of projection matrices. In practice, it means that according to (2) we choose an arbitrary vector **v** and scalar λ. In accordance with the above mentioned statements different choices for $(\mathbf{v}, \lambda)$ will yield different projection matrices, all leading to the same fundamental one. In this paper we have chosen unity values for components of vector **v** and scalar λ. The obtained pair of projection matrices has made possible, in accordance with specifically proposed steps of the methods 2a and 2b, to obtain a metric (Euclidean) reconstruction of space (last two columns of Table 7)). However, experimenting with different values for $(\mathbf{v}, \lambda)$ in (2) gave different reconstruction errors for the same camera pair. That is quite expectable since for different $(\mathbf{v}, \lambda)$ we are starting our way to Euclidean reconstruction from different 3D projective reconstructions. And in the presence of noise, evidently, an error of computing spatial transformation **H**, that will enable us transformation form projective to Euclidean, depends on the chosen $(\mathbf{v}, \lambda)$. This dependency certainly does not go in favor of either method 2a or 2b. Unless perhaps we can find out such values for $(\mathbf{v}, \lambda)$ that will minimize the error of computing transformation **H**, i.e. error of the Euclidean reconstruction (last two columns of Table 7). Existence analysis of such pair $(\mathbf{v}, \lambda)$ has been left for the future work. The transformation from projective to Euclidean space is equivalent to finding a pair of projection matrices that corresponds to a Euclidean reconstruction. The internal parameters found using methods 2a and 2b (Table 5 Table 6) are computed in a manner that for every camera we have decomposed such projection matrix (reconstruction pair member) which contributed to wand length reconstruction with the smallest mean error. For instance, to find the first camera's parameters using method 2b, camera pair *15* was considered (fifth row in last column of Table 7). That is

in agreement with the reasoning where a 3D reconstruction with smallest error should resemble camera parameter values closest to the true ones.

It has been demonstrated that certain thresholding has positive effect on methods 1a and 1b implementations. Similar approach has been tried also for other two methods 2a and 2b. For example, in case of method 2a it would be perhaps beneficial to do thresholding using angles between lines and/or planes in space before actual formation of system of equation to compute absolute dual quadric. However, since the space we start with to form mentioned system of equation is of projective character using such metric information as threshold is not meaningful. Furthermore, considering that image distortion is one of the most serious obstacles for successful implementation we could take into calculation only those image points of certain planes in space which lie around the image center. The drawback of this potential idea is that is not easy to come up with enough number of such frames where image points of some plane in space are around image center of both cameras. Therefore, it appears that only thresholding left is to exclude those data where image distance between end markers are not greater then some values. Although this strategy could be applied for the either of the proposed methods. Evidently in case of method 2a such thresholding is not as efficient as similar thresholding for some other methods. For instance in case of method 1a, gradual consideration of larger number of vanishing points and using the magnitude of skew factor. In case of method 2b situation with thresholding is slightly better then with method 2a. Here we are capable to discard all those vanishing points whose coordinates are of larger value (in all likelihood wand in space is almost parallel to image plane). That surely guaranties more accurate computation of 3D affine space (in accordance with proposed method steps), which explains why results of method 2b (last row of Table 7) are better then results of method 2a (row before the last one in Table 7). One more thing that methods 2a and 2b have in common is they in essence calibrate two cameras at the same time, indirectly through reconstruction (computation) of 3D Euclidean space. Let us remind ourselves that accuracy of 3D reconstruction depends also on cameras spatial configuration (angle between optical axes, ratio of the base line and depth of the points etc.). Naturally we obtain for the same camera, but paired with different others, very different reconstruction errors, even up to the hardly expectable magnitudes (shaded cells in Table 7). For completeness lets just say that all proposed methods work equally well with synthetic data (noise free) outputting perfectly accurate results in terms of camera parameters and reconstructions as well. It simply proves its correct theoretical foundation explained in preceding section.

Conclusion

We have taken theoretically known methods for camera calibration [3] and investigated their applicability in practice in case of typical 3D kinematic systems. Using commonly present calibration tools of many 3D kinematic systems an idea was proposed to perform wand dance with two (three) orthogonal wands instead of a single one. Such a configuration allowed us to use a number of well known projective geometry entities. Specifically, we have showed four possible methods which can give us more accurate and reliable initial camera parameter estimates, assuring faster and saver convergence and most likely better accuracy after the refinement (future work). Besides, using either of the proposed methods two typical phases of calibration (imaging the orthogonal triad of wands before the wand dance) successfully boil down to single one – only the wand dance. In some instances parameter refinement is not necessary. Even without refinement, our initial values give better reconstruction results (tested on reconstruction of known lengths) than using Smart's initials. Among the proposed methods and without consideration of a refinement, method 1a appears to be the method of choice.

References

[1] ALLARD P., STOKES I., and BLANCHI J-P. (1995): Three Dimensional Analysis of Human Movement. Human Kinetics, Champaign.

[2] ATKINSON K.B. (1996): Close Range Photogram. and Machine Vision. Whittles Publishing, Caithness, Scotland.

[3] HARTLEY R. and ZISSERMAN A. (2000): Multiple View Geometry in Computer Vision, Cambridge University Press.

[4] PRESS W.H., TEUKOLSKY S.A., VETTERLING W.T., and FLANNERY B.P. (1997): Numerical Recipes in C. Cambridge University Press.

[5] eMotion/Smart **http://www.emotion3d.com**

[6] http://www.peharec.com/index.html