Robust multi-hypothesis 3D object pose tracking

Georgios Chliveros[1], Maria Pateraki[1], and Panos Trahanias[1,2]

[1] Foundation for Research and Technology Hellas, Institute of Computer Science,
Heraklion, Crete, GR-70013
[2] Dept. of Computer Science, University of Crete, Heraklion, GR-71409

Abstract. This paper tackles the problem of 3D object pose tracking from monocular cameras. Data association is performed via a variant of the Iterative Closest Point algorithm, thus making it robust to noise and other artifacts. We re-initialise the hypothesis space based on the resulting re-projection error between hypothesised models and observed image objects. This is performed through a non-linear minimisation step after correspondences are found. The use of multi-hypotheses and correspondences refinement, lead to a robust framework. Experimental results with benchmark image sequences indicate the effectiveness of our framework.

Keywords: Robot Vision, Object Tracking, Relative Pose Estimation

1   Introduction

Tracking of 3D objects from a monocular camera is important for numerous applications, including robotic control, grasping and manipulation. Several advances have been made but remains a difficult problem due to issues raised from both a theoretical and a practical perspective. Various approaches have been suggested (see survey in [1]), with most commonly employed being those of 'model-based' methods (see survey in [2]).

Early work by Harris [3] utilised a 3D CAD *model* which is projected onto the image frame and registered to the image extracted contour. A similar approach is employed in [4], where the model takes the form of a parametrised 3D object. Since these early works, great effort has been taken to improving upon the correspondence between the object's model and image contours. For example, Drummond and Cipolla [5] use binary space partition trees to determine the model contour (edge points) and subsequently perform a 1D search for corresponding points along the normal of the projected contour on the image. In [6], registration is based on the iterative closest point (ICP), but it considers the re-projection error as a non-linear minimization problem that is solved via the Levenberg–Marquardt (LM) algorithm. Non-linear error minimisation is also utilised in [4, 7] but with a combined 1D point search: the former identifies

correspondences and then performs LM minimisation, whilst the latter locates correspondences by incorporating minimisation of a geodesic active contour objective function using gradient descent.

Even though the aforementioned approaches do consider refinement of the estimated object pose, they do not consider it as a means to *re-initialisation*: i.e. evaluating new model hypotheses. Inasmuch most tracking algorithms assume that a given 'start' or previous estimate of the object pose is sufficient for algorithm initialisation / re-initialisation. Alas, for erroneous pose initialisation values the tracker either does not converge to the true pose, or loses track of the object after some time in long image sequences.

To compensate for initialisation / re-initialisation issues, multiple pose hypotheses have been considered. An initialisation procedure is offered in [4], but it relies on motion segmentation, based on the displacement of extracted image features (contour / edges) between consecutive frames. Vacchetti *et al.* [8] employ a limited number of low level hypotheses and the tracking problem is solved via 'local' bundle adjustment. The incorporation of multi-hypotheses has recently been given particular attention when formulated within probabilistic frameworks (e.g. in [9, 10]). In [11], a Sequential Monte Carlo (SMC) framework has been suggested, where a greater number of pose hypotheses (i.e. particles) is generated and maintained. Unfortunately the search space may be too large to converge at a good initialisation pose and within reasonable time limits. Re-initialisation is considered for establishing and generating a higher number of hypotheses (particles), when degenerate pose estimates occur based on the *effective particle size* defined in [12]. Unfortunately, SMC frameworks are computationally expensive. For example in [11], roughly a single pose estimate per second can be provided.

In this paper we propose the application of image extracted features (Section 2) in a multiple hypotheses 3D object tracking (MH3DOT) framework (Section 3) from known 3D models. However, for finding correspondences in dissimilar model and image point feature sets, we utilise the least median of squares error (Section 3.1). To compensate for re-initialisation issues we further perform a short term adjustment over given correspondence sets via non-linear minimisation (Section 3.2). We apply our MH3DOT method on benchmark sequences and illustrate that it is sufficiently robust for tracking over long time period and for a number of challenges in visual tracking systems (Section 4).

2 Extracted image features

To extract image features (contours and edges) we extend the 2D tracking approach of [13, 14] to apply for multiple objects, according to which foreground coloured pixels are identified based on their colour and grouped together based on their colour histograms. Location and shape of each tracked colour pixels group is maintained by a set of pixel hypotheses which are initially sampled from the observed histograms and are propagated from frame t to $t + 1$ according to linear object dynamics computed by a Kalman filter. The distribution of the propagated pixel hypotheses provides a representation for the uncertainty in

Fig. 1. Extracted image features. From left to right: the original image; the result of thresholding operations; pixel probabilities after labelling; resulting point features, where the contour is shown in yellow and the internal edge is depicted in green.

both the 2D position and the shape of the tracked object. There are no explicit assumptions about the objects motion, shapes and dynamics.

Based on said templates and histograms a number of colour classes $c_i, i \in \mathbb{N}^*$ are formed. This type of primitives can be used in an off-line training process thus building a classifier based on the Bayes rule. The posterior probability for each pixel $I_{n,m}$ with color c to belong to a color class c_i is computed according to Bayes rule $P(c_i|I_{n,m}) = (P(c_i)/P(I_{n,m})) P(I_{n,m}|c_i)$, where $P(I_{n,m}) = \sum_j P(I_{n,m}|c_j)P(c_j)$. The prior probabilities of foreground pixels having $P(I_{n,m})$ specific colour c and $P(c_i)$ specific colour class, are computed via off-line training. $P(I_{n,m}|c_i)$ is the likelihood of colour c foreground regions for specific colour class.

The algorithm handles the issue of assigning a pixel in more than one color classes / objects, by assigning to the class with the highest probability. Using multi-level thresholding operations [15] and standard connected components labeling in the totality of the image [13], we acquire all regions that have high probability of belonging to the tracked object. A further query within contour's pixel regions reveals internal edges of the object (see example of Fig. 1).

3 Methodology for tracking

In our methodology the model and image point features correspondences are found via a '*correspondence process*'. This is performed using an Iterative Closest Point variant, which makes for robustness to noise and other artifacts. Hypotheses are formed from rendered 3D models and are re-initialised by incorporating a non-linear minimisation step over a short term window. This is the 'interpretation process', which adjusts and bounds the pose error.

3.1 Correspondence Process

Given the (intrinsic) camera calibration matrix $\mathbf{K}$ and a projection matrix $\mathbf{P} = [\mathbf{R}|\mathbf{t}]$, then model 3D vectors $\mathbf{m}_i$ can be projected to the image plane; that is $\acute{\mathbf{m}}_i = \mathbf{KPm}_i$, where $\mathbf{m}_i$ are 3D points from model database and $\acute{\mathbf{m}}_i$ represents the 3D-to-2D projection model points. From the rendered model $\acute{\mathbf{m}}_i$, we subsequently extract the projected feature model points $\hat{\mathbf{m}}_i$. In the model
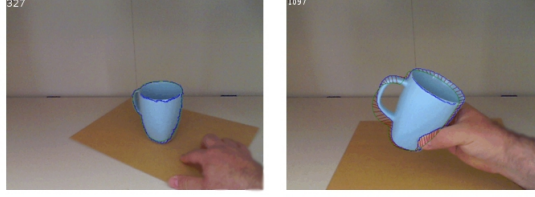
Fig. 2. The TrICP correspondence process: the blue line defines the observed image feature points; the green line illustrates the down-projected model points; the red lines denote the trimmed least squares point correspondences.

feature extraction case, no multi-thresholding and labeling operations are performed, since the rendered down-projection is much simpler. The set of such points $\mathbb{M} = \{\hat{\mathbf{m}}_i\}_1^{n_m}$ is the employed model representation in our work. Observed image feature points are $\mathbb{P} = \{\hat{\mathbf{p}}_j\}_1^{n_p}$ are extracted using the procedure of Section 2.

Iterative Closest Point (ICP) algorithm and its variants have been extensively studied for matching. However, all points from $\mathbb{P}$, $\mathbb{M}$ (or subsets thereof) need be matched due to the iterative least mean squares error. Thus, in the presence of noise and artifacts (e.g. cluttered background) the matching process can rapidly deteriorate. For these reasons we employ ICP with a least median of squares error [16] with a 'trimming' operation. The Trimmed Least Squares ICP (TrICP) [17], allows for the two point sets to contain unequal number of points ($i \neq j$). In our implementation, TrICP calculates the translation and rotation between the feature point sets by 'minimizing' the sum of the least median squared individual Mahalanobis distances [16], defined as

$$\mathbf{d}_{ij}^2 = (\hat{\mathbf{m}}_i - \hat{\mathbf{p}}_j)^{\mathrm{T}}(\mathbf{S}_{m_i} + \mathbf{S}_{p_j})^{-1}(\hat{\mathbf{m}}_i - \hat{\mathbf{p}}_j) \qquad (1)$$

where $\mathbf{S}_{m_i}$ is the covariance, thus the uncertainty, on the position of point feature $\hat{\mathbf{m}}_i$; and respectively for $\mathbf{S}_{p_j}$ of $\hat{\mathbf{p}}_j$.

To improve speed we first employ nearest neighbour search in $\mathbb{P}$, $\mathbb{M}$ using uniform grid structures[1]. The best possible alignment between data / model sets is found by 'sifting' through at most i nearest-neighbour combinations in an attempt to find a subset of size less than j, which yields the lowest sum of ordered $\mathbf{d}_{ij}^2$ values. When performing the least median search two thresholds are employed: (i) max distance to be used between valid point combinations, and (ii) max percentage of points allowed for the 'trimming' operation. An example of the correspondence loop can be found in Fig. 2, where the trimmed least squares point correspondences can also be seen.

Hypotheses initialisation: The starting point for finding correspondences is models generation from a parametrised pose estimate $\mathbf{s} = (t_x, t_y, t_z, \alpha_x, \alpha_y, \alpha_z)$ given

---

[1] Other structures like 'range-trees' would cost with each query $\mathcal{O}(N \log N)$ as opposed to $\mathcal{O}(N)$ in the uniform case.

at the previous frame instance, where t and α are the translation and rotation elements respectively. However, at the start of an image sequence this may not be the case (no prior pose is provided). Thus, we need to have in place an initialisation procedure, so that we generate a representative search space over rotations $(\alpha_x + \delta\alpha_x, \alpha_y + \delta\alpha_y, \alpha_z + \delta\alpha_z)$. The term $\delta\alpha$ can be assigned as dictated by a number of increment steps (N) over the full rotation range $(0, \pi)$ of the corresponding axis. However, this results in at least $3N(N^2 + 1)$ model hypotheses, which will require a great computational effort. In our implementation, rotations over the x, y axes, are constrained by the use of of monitoring the displacement of image extracted observed features, based on the assumption that for the first few frames, neighbouring moved image features represent projected features of the object we wish to initialise for our tracking process. It can be estimated that this constraint reduces the model hypothesis space to approximately N^2. The advantage of this procedure is that there will always exist a pose estimate, reflecting the system's 'best guess' of what the actual pose of the object is.

With respect to the covariance matrix $\mathbf{S}_{p_j}$, this is a 2×2 matrix with predetermined values modelling the observation noise on the camera frame, such that $0 < \sigma_x, \sigma_y < 0.2$; experimentally verified as an appropriate threshold. It should be evident that the higher the σ_x, σ_y values, the higher the uncertainty expected over the x and y axis respectively. The covariance matrix $\mathbf{S}_{m_i}$ is initialised as a unit matrix, and is subsequently updated within the interpretation process of Section 3.2.

3.2 Interpretation Process

The 're-initialisation' problem mentioned in Section 1, can be formulated as a minimisation problem either within small image frames sequence window (local) or a large window (global) bundle adjustment framework. Inspired by [4], we use an 'interpretation' loop because the correspondence sets' found in Section 3.1 process, need to be assessed for their 'validity' (i.e. reduce the number of outliers) as time passes over a sequence of neighbouring frames. Therefore the resulting set of model interpretations, are those that result in the smallest residuals between models and observed point features. Only those with the smallest residual are referenced as potential solutions and maintained over the next few frames. The aforementioned residuals are thus re-evaluated via non-linear minimisation of the re-projection error, between the model(s) and observed feature points. This is expressed as the sum of squares of a large number of nonlinear real-valued functions; i.e. a non-linear least squares problem. Thus, the objective function is formulated as:

$$\hat{\mathbf{s}}_t = \operatorname*{argmin}_{\mathbf{s}} \sum_{i=1}^{n} ||\mathbf{p}_i - f(\mathbf{s}, \mathbf{m}_i)||^2 \tag{2}$$

where $f(\cdot)$ is the function that projects the 3D model points to the image plane, according to $\mathbf{s}$.
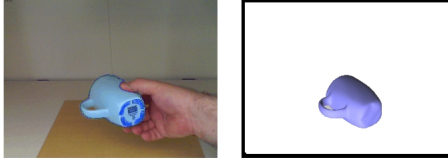
Fig. 3. The interpretation process: an example of pose model best hypothesis after the LM optimisation step takes place.

Assuming an initial pose estimate $\hat{\mathbf{s}}$ equal to the current TrICP pose estimate, found during the correspondence process, the pose is updated iteratively according to $\hat{\mathbf{s}}_t = \hat{\mathbf{s}}_t + \Delta t$, where Δt is given by:

$$\Delta t = -(\mathbf{J}^\mathrm{T}\mathbf{J} + \mu\mathbf{I})^{-1}\mathbf{J}^\mathrm{T}\epsilon_t \qquad (3)$$

and $\mathbf{J}$ is the Jacobian resulting from $f(\cdot)$ computed at $\mathbf{s}_t$, and $\epsilon_t = |f(\hat{\mathbf{s}}_t) - f(\mathbf{s}_t)|$. The scalar μ, computed after every iteration, is a 'dumping term' and controls the behavior of a Levenberg-Marquardt (LM) algorithm. If the updated pose leads to a reduction in the error, the update is accepted and the process repeats with a decreased damping term μ. Otherwise, the damping term is increased, and the process iterates until a value of Δt that decreases the error is found.

An example output of this process can be seen in Fig. 3, 4. In Fig. 3 we illustrate a case where we selected the best hypothesis out of the initial N number of hypothesis generated. The realisation of these N number of hypotheses is illustrated in Fig. 4. At run-time execution we have set the generation of 35 hypotheses when the error is deemed to be large. In Frame 1300 the error was at 0.1689, i.e. greater than the preset value of 0.0025. Of the total number of formed hypotheses, and at Frame 1302 they are reduced to 6. In total, from Frame 1300 to Frame 1304, only three hypotheses survive (from a least squares fitting error of 0.1688 to 0.0022.

In this local framework, the normal equations have a sparse block structure. This is due to the fact that there is a lack of interaction among parameters for different (down-projected) 3D points and camera extracted feature points. This can be exploited to the overall algorithm computational benefit by avoiding storage and operation upon zero elements. Thus a sparse variant of the LM algorithm that takes advantage of the normal equations zeros pattern, greatly reduces the computational effort involved [18].

Hypotheses re-initialisation: Contour and edge points of an object may not provide enough information to uniquely identify the objects pose. This would become more prominent when occluded areas of the object and natural obstructions as well as the object coming in and out of the field of view of the camera.

To remedy for the aforementioned, if in the TrICP correspondence loop the best pose estimate has a large error attached to it, then an LM interpretation process is initiated. For a large LM error a greater number of rendered model hypotheses are generated. Hypotheses are generated by rotating the object model

Frame	No. Hyps	Input	Generated Hyps					
1301	35							
1303	8							
1304	6							
1305	3							

Fig. 4. An initialisation / re-initialisation case with hypotheses maintenance: a large re-projection error initiates a threshold number of hypothesis. For illustration purposes only few of the true number of generated hypotheses is reported.

with respect to a previous frame's pose estimate. The number of frames is dictated by the number of frames LM is allowed to operate upon.

Each of these model hypotheses is assigned a value of the goodness-of-fit with respect to the current image frame. The values are updated from frame to frame based on the minimization of the objective function. Hence, multiple hypotheses are generated only when the error returned by the minimization algorithm exceeds a threshold. The covariance matrix calculated in the LM minimisation step, is assigned for $\mathbf{S}_{m_i}$ in the correspondence process'.

Particle filters perform re-initialisation using Doucet's *effective particle size* N_{eff} [12]. In [11], and to avoid getting trapped in local minima, an additional rule is considered whereas if $N_{\text{eff}} < N_{\text{threshold}}$ then a set max number of pose hypotheses are generated. In our methodology a max number of model hypotheses is generated only when $\epsilon_t < \epsilon_{\text{threshold}}$, i.e. the error returned by the minimization algorithm exceeds a given threshold, or when a max number of LM iterations has been reached. We derive our 'efficient number of hypotheses' N_{hyp} using Chebyshev's inequality for n independent random variables. It follows that, for some mean value $\mu = \max(\mu_i)$ and variance $\sigma = \max(\sigma_i)$ over the observations made for the set of image frames and for pose vector values $\mathbf{s} = \{S_i\}$, the probability within an expected sensitivity k is:

$$\mathrm{P}\left\{\left|\sum_{i=1}^{n}(S_i/n) - \mu \geq k\right|\right\} \leq \sigma/nk^2 \tag{4}$$

Table 1. Average total error in full image sequence taken per frame in image sequence. The total translational error is in mm and rotation in degrees. Reported time is in msec.

Sequence	Challenges	Methodology	Time	Total error					
				X	Y	Z	Roll	Pitch	Yaw
Panda toy	illumination, clutter	BLORT	84	3.8	6.6	3.3	2.1	3.2	1.3
		LM-ICP	31	20.2	32.0	11.7	2.7	6.3	4.7
		ViSP	40	12.7	21.1	13.4	7.5	4.4	3.4
		MH3DOT	132	3.9	5.5	3.1	2.2	2.1	1.9
Coffee box	viewpoint, scale	BLORT	147	1.2	2.3	1.8	3.0	4.2	1.7
		LM-ICP	54	11.3	7.6	4.4	4.1	11.0	6.8
		ViSP	62	6.2	4.7	2.1	3.1	5.1	4.9
		MH3DOT	195	1.4	2.5	1.9	3.2	4.1	1.5
Mug/Cup	viewpoint, clutter	BLORT	150	2.2	1.7	2.9	13.1	11.7	3.5
		LM-ICP	85	13.1	12.5	12.9	6.8	9.6	8.2
		ViSP	98	7.0	5.1	8.6	6.6	23.1	11.0
		MH3DOT	224	1.3	1.9	2.3	2.8	5.6	2.3

For example, if $\sigma = 1$ and we wish to be 95% confident that our estimations are within $k = \epsilon_t = 0.5$ units at some model hypotheses of pose $\mathbf{s}$ then $\sigma/nk^2 = 1/0.25n = 4/n$ and our $N_{\mathrm{hyp}} = 4/0.05 = 80$, where the estimate n is assumed to be *sufficient*. An example of re-initialisation ($N_{\mathrm{hyp}} = 35$) is provided in Fig. 4.

4 Results

In order to evaluate our approach we have used the BoBoT[2] benchmark on tracking sequences which include partial ground truth data. For evaluation over model based with no hypotheses generation and maintenance we applied the ViSP[3] software toolbox and a variant of LM-ICP. To evaluate performance of our MH3DOT and other multiple pose hypotheses methods, we have applied the BLORT software toolbox, which implements a particle filter. The results are summarised in Table 1. It should be noted that both ViSP and BLORT use hardware acceleration. Our implementation does not currently support hardware acceleration. Optimisations are performed in the sense of custom matrix and array operations, based on uBLAS and Lapack libraries. For completeness we also report the computational time required by each method. We note that our method is close to BLORT in terms of computational performance and certainly applicable for on-line applications. A quantitative analysis on the sequences used alongside the aforementioned software solutions, is presented in Table 1.

For our experimental test, we set $n = 100$ for both BLORT (max particles number) and MH3DOT (effective number of hypotheses). In the case of LM-ICP and ViSP, the methods do not employ multiple hypotheses ($n = 1$). It should be noted from the error results of Table 1, that BLORT reports less

---

[2] http://www.iai.uni-bonn.de/~kleind/tracking/, /~martin/tracking.html
[3] http://www.irisa.fr/lagadic/visp/visp.html

Fig. 5. Snapshot images with super imposed pose results from BoBoT's 'cup' image sequence: MH3DOT tracker versus BLORT. The green line is the object pose from MH3DOT and the yellow line corresponds to BLORT.

errors in the 'coffee box' complex background sequence, whilst it shows large errors in the 'mug' simpler background sequence. In contrast, our hybrid method performs consistently in both image sequences. BLORT error results are slightly better than our MH3DOT method in the X and Z translation. However, BLORT severely suffers from increased errors in angular rotations. Furthermore, the pose results of ViSP, and to some extent BLORT, indicate increased error in rotation, which under certain conditions, is not desirable for robot vision applications. We postulate that the superiority of our MH3DOT method may be due to the fact that use of an interpretation process (Section 3.2), constrains this type of pose errors. That is, do not propagate into inferences for subsequent frames.

An example from the 'mug' sequence (three frames), with tracking superimposed, is provided in Fig. 5. From the sequences tested we can conclude that methods with multi-hypothesis generation and maintenance (MH3DOT green line in Fig. 5) track with good accuracy the target objects. We also observed that for the used case Particle filter tracker (BLORT yellow line in Fig. 5) at some point in time they converge to an erroneous result.

As evidenced in Fig. 5, by frame 869 BLORT pose estimation deteriorates, whilst MH3DOT remains consistent. Thus, the target object will no longer be tracked after some time has elapsed within the image sequences. This becomes even worse for single hypothesis implementations. In said cases, tracking fails to recover the target, and the object remains as 'lost' without successful recovery until the end of the sequence. This explains (in-part) the increased errors reported in Table 1. This is not the case with our MH3DOT approach and thus the reported error is smaller.

5   Conclusions

This paper presents a model based approach to tracking the pose of an object in 3D based on 2D derived contours and edges, using a monocular camera. To enhance the performance of our method under occlusions and other artifacts, we have established a generation of multiple hypothesis, in the form of rendered objects. For this purpose, we have formulated an efficient number of hypotheses criterion within our framework's implementation. Experimental results have

demonstrated that our method achieves good pose tracking resolution at a relatively fast frame rate. The results have indicated that our tracking method exhibits better performance over the tested methods with reported parameters.

Further research is required to establish potential cases under which the method may not work robustly. However, in the current challenges posed by the image sequences used, the algorithm has shown to operate robustly even under situations where environmental (e.g. lighting, clutter) and motion conditions (e.g. motion, scale changes) are realistic. Finally, the criterion for efficient number of hypotheses will constitute a topic for further study.

Acknowledgments

References

1. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Computing Surveys **38**(4) (2006) 1–46
2. Lepetit, V., Fua, P.: Monocular model-based 3D tracking of rigid objects: a survey. In: Foundations and Trends in Computer Graphics and Vision. (2005)
3. Harris, C., Stennet, C.: RAPiD – A video-rate object tracker. In: British Machine Vision Conference. (1990) 73–77
4. Koller, D., Daniilidis, K., Nagel, H.: Model-based object tracking in monocular image sequences of road traffic scenes. International Journal of Computer Vision **10** (1993) 257–281
5. Drummond, T., Cipolla, R.: Real-time visual tracking of complex structures. IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002) 932–946
6. Fitzgibbon, A.: Robust registration of 2D and 3D point sets. Image and Vision Computing **21**(13) (2003) 1145–1153
7. Paragios, N., Deriche, R.: Geodesic active contours and level sets for the detection and tracking of moving objects. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(3) (2000) 266–280
8. Vacchetti, L., Lepetit, V., Fua, P.: Stable real-time 3D tracking using online and offline information. IEEE Transactions on Pattern Analysis and Machine Intelligence **26** (2004) 1385–1391
9. Azad, P., Münch, D., Asfour, T., Dillmann, R.: 6-DoF model-based tracking of arbitrarily shaped 3D objects. In: IEEE Int. Conf. on Robotics and Automation. (2011)
10. Puppili, M., Calway, A.: Real time camera tracking using known 3D models and a particle filter. In: IEEE Int. Conf. on Pattern Recognition. (2006)
11. Choi, C., Christensen, H.I.: Robust 3D visual tracking using particle filtering on the special Euclidean group: A combined approach of keypoint and edge features. The International Journal of Robotics Research **31**(4) (2012) 498–519
12. Doucet, A., Godsill, S., Andrieu, C.: On Sequential Monte Carlo sampling methods for Bayesian filtering. Statistics and Computing **10**(3) (2000) 197–208
13. Argyros, A., Lourakis, M.: Real-time tracking of multiple skin-colored objects with a possibly moving camera. In: European Conference on Computer Vision. (2004)

14. Baltzakis, H., Argyros, A.: Propagation of pixel hypotheses for multiple objects tracking. In: Advances in Visual Computing. Volume 5876 of Lecture Notes in Computer Science. (2009) 140–149
15. Liao, P.S., Chen, T.S., Chung, P.C.: A fast algorithm for multi-level thresholding. Journal of Information Science and Engineering **17** (2001) 713–727
16. Rousseeuw, P.J.: Least median of squares regression. Journal of the American Statistical Association **79**(388) (1984) 871–880
17. Chetverikov, D., Stepanov, D., , Krsek, P.: Robust Euclidean alignment of 3D point sets: the trimmed iterative closest point algorithm. Image and Vision Computing **23** (2005) 299–309
18. Lourakis, M.: Sparse non-linear least squares optimization for geometric vision. In: European Conference on Computer Vision. (2010)