N3M: Natural 3D Markers for Real-Time Object Detection and Pose Estimation

Stefan Hinterstoisser Selim Benhimane Nassir Navab
Department of Computer Science, Technical University of Munich
Boltzmannstr. 3, 85748 Garching, Germany
hinterst@in.tum.de, selim.benhimane@in.tum.de, navab@cs.tum.edu

Abstract

In this paper, a new approach for object detection and pose estimation is introduced. The contribution consists in the conception of entities permitting stable detection and reliable pose estimation of a given object. Thanks to a well-defined off-line learning phase, we design local and minimal subsets of feature points that have, at the same time, distinctive photometric and geometric properties. We call these entities Natural 3D Markers (N3Ms). Constraints on the selection and the distribution of the subsets coupled with a multi-level validation approach result in a detection at high frame rates and allow us to determine the precise pose of the object. The method is robust against noise, partial occlusions, background clutter and illumination changes. The experiments show its superiority to existing standard methods. The validation was carried out using simulated ground truth data. Excellent results on real data demonstrated the usefulness of this approach for many computer vision applications.

1. Introduction

For many years, artificial 2D and 3D markers have been successfully used in many vision-based applications: camera calibration [25], augmented reality [4, 9] and robotic vision [3, 13] just to name a few. These markers are generally designed in a way that allows them to be easily detected with very simple image processing operations. For some applications, their geometry is carefully chosen in order to avoid degenerate pose estimation.

On the other hand using natural features for vision problems is a more recent development. Some early work related to feature extraction was done in [6, 7, 23]. Today it is generally agreed upon that in order to detect and to match these features for image retrieval and object recognition, region detectors [8, 14, 15, 26] and enhanced feature descriptors [12, 18, 20] should be considered. Recently, the challenge has become the improvement of the efficiency of these region detectors and the feature descriptors in order to be used in real-time applications [1, 11], mainly by adding off-line learning which makes it possible to reduce the run-time computations. Until now, mostly photometric properties have been learned. Surprisingly, very few approaches considered incorporating the 3D models of the objects during the learning process [11, 16, 17].

The strength of artificial 3D markers in providing intrinsically stable detection and reliable pose estimation has not been yet replaced with the proposed markerless methods. This made the use of robust algorithms, such as RANSAC [5], inevitable during run-time in order to make the result given by the photometric properties consistent with the geometry of the object. The goal of this paper is to define in a first approach entities attached to the considered object that have, at the same time, distinctive photometric and geometric properties. We call these entities *Natural 3D Markers (N3Ms)* since the off-line learning step that selects these entities makes the detection and the pose estimation fast and straightforward. Inspired by the "visual vocabulary" consisting of 2D Words proposed by [19, 22] for recognition and classification tasks, we define quasi-optimal configurations of feature subsets that build up a "visual 3D vocabulary" to obtain a more abstract description of the 3D object for the use of object detection and pose estimation. However, being a 3D entity, the N3Ms could also play the same role as 3D markers. The remainder of the paper is structured as follows: In the second section, we state the problem and relate our contribution to the current state of the

Figure 1. 3D markers on a laparoscope (left) and a possible N3M on an industrial box (right)

978-1-4244-1631-8/07/$25.00 ©2007 IEEE

art. In the third section, we describe the learning phase of the algorithm for the selection of Natural 3D Markers on an object (or in a scene) given its appearance (from 2D images) and its geometry (from a 3D CAD model or reconstructed model). We also present the multi-level approach that uses these entities to simultaneously detect and estimate the pose during the run-time phase. In the experiments section, we compare our method to two popular methods using realistic simulations with ground truth. Promising experimental results in real world conditions are also presented and the robustness of the algorithm against noise, partial occlusions, background clutter and illumination changes is shown.

2. Related Work

Typically, markerless object detection and pose estimation start with features extraction [2, 6, 7, 23]. This step used to be followed by matching algorithms based on similarity measures such as the Normalized Cross-Correlation (NCC) [27] or on the dot product of edge directions [24]. Such algorithms work well when the object motion is limited to a translation in a plane parallel to the image plane. Other methods based on affine invariant regions determined around feature points were proposed [8, 14, 15, 26] in order to obtain invariance to out-of-plane rotations and translations. Unfortunately, for real-time applications, these algorithms are too slow.

Recently, more efficient algorithms (based exclusively on feature points and descriptors or classifiers) were introduced most notably SIFT [12] and Randomized Trees [11]. These algorithms work well for generic motions and are less sensitive to noise, wide baseline viewpoint changes and partial occlusion. The SIFT method describes the region around a feature point by computing weighted gradient histograms. These gradient histograms are collected in a normalized vector that is used in a nearest neighbor matching process. The advantage of SIFT is that it tolerates significant local deformations. However, it is still sensitive to large viewpoint changes and despite attempts to improve its speed [1], it remains quite slow. In contrast to the vector based approaches, Randomized Trees consist of decision trees based on series of pixel intensity comparisons. The trees need to be learned offline and the pixels involved in the comparisons are chosen randomly in a window around the feature points. In addition to their simplicity, Randomized Trees are very fast and work well for large viewpoint changes.

Despite their efficiency in matching, these approaches still need a subsequent method that rejects all falsely established point correspondences - called outliers. This is mostly done by robust methods, such as RANSAC [5], that enforce the matched feature points to be consistent with the object model and geometry. With these approaches the appearance (photometric properties) and the model of the object (geometrics properties) are considered sequentially. We propose a unified approach that makes use of N3Ms, where an offline learning stage permits to take into consideration both the photometric and the geometric properties simultaneously during the detection and the pose estimation.

3. Natural 3D Markers

Our goal is to detect and estimate the pose of a given 3D object in a stable and precise way. The proposed method is carefully tailored in order to deal with the usual limitations of the standard approaches: noise, illumination changes, severely oblique viewpoints, partial occlusions and background clutter. The contribution lies on the learning of minimal point sets, called Natural 3D Markers (N3Ms) that fulfill the requirements for stable pose estimation and that are therefore able to replace artificial 3D markers.

An N3M is a set of 4 or 5 close feature points with distinctive photometric and geometric properties: the feature points that are selected should be able to be extracted under multiple viewpoints, various illumination changes and noise conditions. In addition, the feature points forming an N3M are grouped in a way that guarantees their visibility from at least one common viewpoint, their adequacy to perform a geometric consistency check to validate the feature point matching and a non-singular configuration during the pose estimation (despite their locality). Theoretically, detecting a single N3M on an object is sufficient to determine its pose.

3.1. Learning Stage

In this section, we describe how the feature points are selected in a way that ensures distinctive photometric properties and an equal distribution over the object's visible surface. We also propose a process for grouping these features into entities guaranteeing non-singularity during pose estimation and robustness to partial occlusions.

3.1.1 Preprocessing Stage

Feature selection: The first step consists in learning feature points that can be detected under multiple viewpoints, illumination changes and noise. Harris Corner points [7] turn out to have a good mixture between illumination invariance, fast computation and invariance to large viewpoint changes [21]. Note that other point detectors could also be used. In order to take the most stable points, we synthetically render the textured 3D model of the considered object under different random transformations, add noise to it and extract Harris corner points. Since the transformations are known, we can compute for each physical point the repeatability of extraction. A set of points with high repeatability is temporarily selected for further processing.

Equal Distribution: If all feature points were clustered in one region, the detection would not be possible as soon as this region becomes occluded. Therefore, we need to guarantee as far as possible that the feature points are equally distributed over the surface of the object. A trade-off between the equal distribution and the repeatability should be considered. Since every object can be approximated as piecewise planar, we make sure that the number of the points extracted on each plane is proportional to the ratio between the area of the plane and the overall surface area of the object. This does not avoid having clustered point clouds in one specific part of a plane but ensures that the points are fairly distributed among the different planes of the objects.

Visibility Set: In the final preprocessing step we have to compute a visibility set for each 3D feature point. Such a visibility set contains all viewpoints, from which the 3D feature point is visible. For this reason we define an approximated triangulated sphere around the object, where each triangle vertex stands for one specific viewpoint, and shoot rays from these viewpoints to each 3D feature point. If a ray from a certain viewpoint intersects the object on the 3D feature point first, this viewpoint is inserted into the visibility set of the 3D feature point.

3.1.2 Learning Natural 3D Markers

An N3M is a set of 3D *coordinate points* defining a local coordinate system and one 3D *checker point* expressed in this local coordinate system permitting us to check the N3M's point configuration on geometric consistency. Consequently, we distinguish two possible cases: planar (defined with 3 coordinate points) and non-planar (defined with 4 coordinate points) N3Ms. See Figure 2 for an illustration.

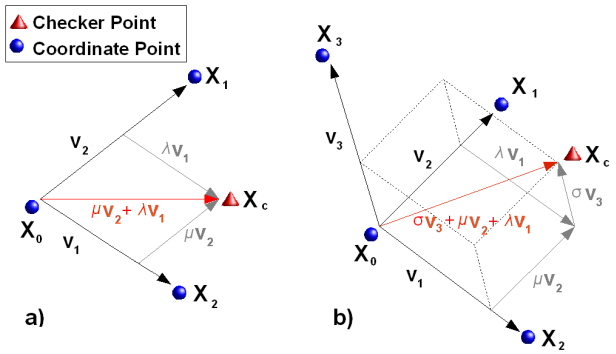

Figure 2. Planar N3M on the left side and a non planar N3M on the right side

Creating all potential N3Ms: Since an N3M only contributes to detection and pose estimation if all its points are extracted and correctly matched, the points should be located in the same local neighborhood. This increases the probability that an N3M is also detected under self- or partial occlusion of the object. We use Algorithm 1 in order to create all potential N3Ms. Note that this algorithm allows that one feature point belongs to multiple N3Ms. This is called *connectivity*. If the N3Ms were constructed such that one feature point belonged to a single N3M, the rest of the feature points of that N3M could not be used, as soon as one feature point of an N3M was not extracted or badly matched. With connectivity, we therefore increase the probability that a correctly matched feature point belongs to at least one N3M, for which all other feature points are also correctly matched. An example for connectivity is shown in Figure 3.

Algorithm 1 Calculate set G of all potential N3Ms

Require: *extracted feature points* $\mathbf{X}_i$
  $G \leftarrow \{\}$
  for all $\mathbf{X}_i$ **do**
    create all possible quadruplets Q_{ik} *including* $\mathbf{X}_i$ *in a local neighborhood of* $\mathbf{X}_i$*;*
    for all Q_{ik} **do**
      if *the points of* Q_{ik} *are all on the same plane* **then**
        1. $S_{ik} \leftarrow Q_{ik}$
        2. label an arbitrary point $\in S_{ik}$ *as checker point*
      else
        1. $S_{ik} \leftarrow Q_{ik} \cup \{\mathbf{X}_j\}$*, where* $\mathbf{X}_j$ *is another neighbor*
        2. label $\mathbf{X}_j$ *as checker point*
      end if
      if *the intersection of the visibility set of the feature points forming* S_{ik} *is not the empty set* **then**
        $G \leftarrow G \cup \{S_{ik}\}$
      end if
    end for
  end for

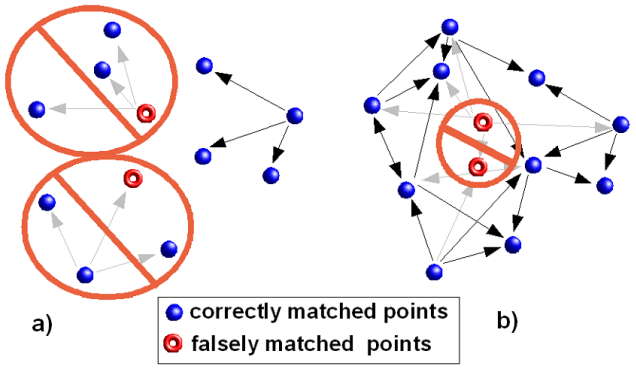

Figure 3. Connectivity as shown in b) avoids loosing correctly matched feature points as seen in a).

Removing ill-conditioned N3Ms: We know that point configurations that are close to collinear or located in a very small neighborhood lead to unstable detection and pose estimation results. In order to exclude these cases, we apply a *tube-collinearity* test. Three points are tube collinear, if one of these three points is located within a tube of radius d_t whose axis is the line connecting the two other points. See Figure 4 for an illustration. To remove all N3Ms that

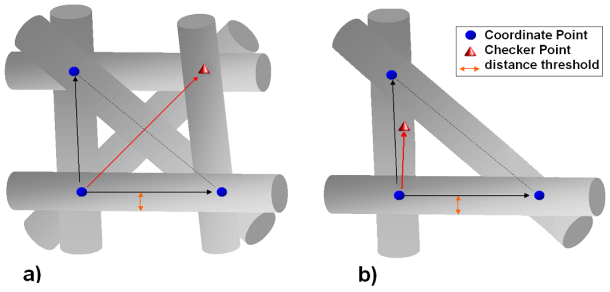

Figure 4. Tube-collinearity test for a planar N3M. If more than 2 points are lying within one gray tube then the N3M is rejected. In Figure a) one sees an accepted N3M whereas the N3M in Figure b) is rejected.

are close to degenerate point configurations, we exclude all N3Ms that contain tube collinear points. For this purpose we compute a quality value for every N3M using the value:

$$\prod_{ij}\left(1-\exp\left(-\frac{1}{2}\left(\frac{d_{ij}}{d_t}\right)^2\right)\right) \tag{1}$$

where d_{ij} is the distance from the ith point to the jth line formed by two other points of the N3M. This quality measure is normalized between 0 (ill-conditioned) and 1 (well-conditioned). The N3Ms with a quality value below a certain threshold are discarded. Since each formed set - obtained by this algorithm - is both local and well-conditioned, we can theoretically use it for stable pose estimation of the object, once it is detected.

3.1.3 Single Point Classifiers

The final learning step consists in learning a point classifier for the feature points forming one or multiple N3Ms. We choose to use the Randomized Trees [11] for the reasons explained above. Note that other classifiers could also be used. In addition, for each N3M $\{\mathbf{X}_i,\ i\in\{0,1,2,3,c\}\}$, we store the 3D coordinate system origin $\mathbf{X}_0$, the local coordinate axes $\mathbf{V}_i=\mathbf{X}_i-\mathbf{X}_0, i\in\{1,2,3\}$, and the coordinates $(\lambda,\mu,\sigma)^\top$ of the checker point $\mathbf{X}_c$ expressed in the local coordinate system $\{\mathbf{X}_0,\mathbf{V}_1,\mathbf{V}_2,\mathbf{V}_3\}$:

$$\mathbf{X}_c=\mathbf{X}_0+\lambda\mathbf{V}_1+\mu\mathbf{V}_2+\sigma\mathbf{V}_3$$

In case of planar N3Ms, $\mathbf{X}_3$ and $\mathbf{V}_3$ do not exist and $\sigma=0$.

3.2. Run Time Stage

During the run-time, in each acquired image, the feature points are extracted and the preliminary one-to-one 2D-3D correspondences are obtained using the point classifier. Only points participating in forming complete N3Ms are considered in the matching. The other feature points are discarded. In order to remove falsely matched points and to compute the pose, we use a two-step algorithm.

3.2.1 Step 1: Self Verification of the N3Ms

Each N3M can be *self-verified* independently of other N3Ms. In fact, given the relative position of the checker point with respect to the local coordinate points, we introduce a score function that tells us whether a subset of points of the N3M is correctly matched or not. Let $\mathbf{v}_i,\ i\in\{1,2,3\}$ be the real 2D coordinate axes and $\mathbf{x}_0$, $\mathbf{x}_c$ be the real coordinate origin and the real checker point after projection in the image. Since the N3Ms are local, every projection matrix $\mathbf{P}$ can be approximated by a linear fronto-parallel projection matrix $\widetilde{\mathbf{P}}$ that preserves parallelism. Thus, we have:

$$\mathbf{x}_c=\mathbf{P}\mathbf{X}_c\approx\widetilde{\mathbf{P}}\mathbf{X}_c\approx\mathbf{x}_0+\lambda\mathbf{v}_1+\mu\mathbf{v}_2+\sigma\mathbf{v}_3 \tag{2}$$

Now let $\mathbf{v}_i^*,\ i\in\{1,2,3\}$ be the 2D coordinate axes and $\mathbf{x}_0^*$, $\mathbf{x}_c^*$ be the coordinate origin and the checker point as 'detected' in the image. The score function:

$$f=\|\mathbf{x}_c^*-\mathbf{x}_0^*-\lambda\mathbf{v}_1^*-\mu\mathbf{v}_2^*-\sigma\mathbf{v}_3^*\| \tag{3}$$

returns a low score in case of correctly matched N3M and a high score if one of the feature points is falsely matched. The proposed score function is similar to Geometric Hashing [10]. It permits to remove most of the falsely matched N3Ms. Some very special configurations remain and need the second step of the algorithm to be automatically removed.

3.2.2 Step 2: Voting scheme

Given the high percentage of correctly matched N3Ms after the first step, we exclude the incorrectly matched N3Ms by proposing the following voting scheme: if the pose provided by one N3M is confirmed (or voted for) by a certain number of other N3Ms, the correspondences of this N3M are added to the set of correspondences for global pose estimation. Experimentally, we found that the voting by two other N3Ms is enough to ensure precise detection and pose estimation. The voting process is shown in Figure 5. Alternatively, for planar N3Ms, one could also compute a similarity measure (e.g. NCC) between the area of the current image enclosed by the 2D feature points and the texture of the model enclosed by the corresponding N3M. Due to the

non degenerate point configurations of an N3M, the similarity measure can easily be computed after mapping the current image area to the corresponding model texture. This similarity based voting enables an N3M to be totally verified by itself. The complete two-step algorithm is summarized in Algorithm 2.

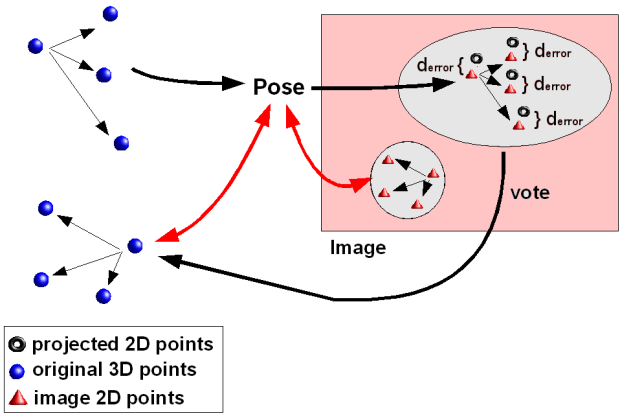

Figure 5. N3Ms vote for each other's validity

Algorithm 2 Calculate the Pose of an Object with N3Ms

Require: *trained Natural 3D Markers* $N3M_i$
  $S \leftarrow \{\}, F \leftarrow \{\}$
  extract the feature points X_i *in the current image*
  for all X_i **do**
    classify X_i *and establish 2D-3D correspondences*
  end for
  for all $N3M_i$ **do**
    if $N3M_i$ has all member points matched **then**
      if $f_s(N3M_i) < t_s$ **then**
        $S \leftarrow S \cup \{N3M_i\}$
      end if
    end if
  end for
  for all $N3M_i \in S$ **do**
    if *m-N3Ms of* S *vote for* $N3M_i$ **or** *NCC(*$N3M_i$*) is high* **then**
      $F \leftarrow F \cup \{N3M_i\}$
    end if
  end for
  compute the pose with all points of all $N3M_i \in F$

4. Experimental Validation

Since automatic recognition of 3D subsets of feature points using the N3Ms is new, we compare our overall matching/pose estimation performance to the most common alternative approaches for automatic 2D/3D matching and pose estimation. To evaluate the validity of our approach, we performed several experiments on synthetic images with ground truth and on real images comparing our method to the standard matching and pose estimation methods using SIFT and Randomized Trees followed by RANSAC [5] (in order to remove potential outliers). The synthetic images are created by rendering a textured 3D model on a highly cluttered background under 500 random poses. For each pose, we simulate 80 different occlusions of the object by a textured pattern. The sized of the occluded region increases from 0% to 95% of the global surface of the object in the image. Thus, we obtain for each pose and for each degree of partial occlusion one synthetic image on which we run the standard Randomized Trees and SIFT (both combined with RANSAC running with a maximum of 1000 iterations) and the N3Ms approach. The recovered pose parameters of each method are then compared to the underlying ground truth data. A pose estimation is considered successful, if the error of the estimated rotation is less than 5 degrees and the error of the estimated translation is less than 5 centimeters along each axis. For each degree of partial occlusion, we count the number of correctly recovered poses. In Figure 6, we display the results. We see that the N3Ms approach and SIFT

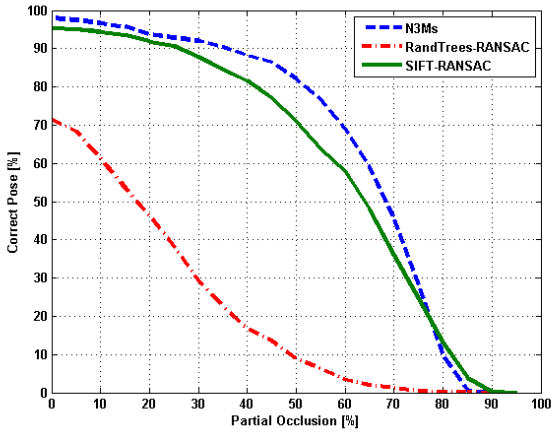

Figure 6. Natural 3D Markers versus Randomized Trees and SIFT

combined with RANSAC clearly outperform Randomized Trees combined with RANSAC. This is due to the fact that we have many outliers in the synthetic images compared to the number of inliers because of the highly cluttered background and because of partial occlusion. Since outlier elimination in our approach is not dependent on the overall number of inliers, N3Ms are very robust to incorrectly matched feature points. The better pose estimation performance of SIFT combined with RANSAC compared to Randomized Trees combined with RANSAC is mainly explained by the fact that the nearest neighbor matching used by SIFT is a natural barrier for falsely matched feature points and therefore produces less outliers for a highly cluttered background with partial occlusion than the classification with Randomized Trees, where the natural barrier is weaker and most

feature points are assigned to one class.

In Figure 6, we can also see that the results of our approach are slightly better than the ones obtained with SIFT combined with RANSAC. However, from the efficiency point of view, the frame rate of the (non optimized version) of the N3Ms approach is about 10 fps on a 1.0GHz Intel Centrino notebook with 512MB memory. While, on the same hardware, SIFT is running with 1.5 fps and the Randomized Trees with 12 fps. Consequently, if we take into account both the correct results obtained and the computational efficiency, our approach performs better than the two others. This was also confirmed with the real world examples. See Figure 7 for some excerpts.

5. Discussion

The method presented is a first attempt towards incorporating the 3D models of the objects during the learning process in order to design Natural 3D Markers for detection and pose estimation. Compared to methods like Randomized Trees that need a training step for the detection, our method greatly improves the detection rate and the pose estimation results thanks to its additional training of the N3Ms configurations. We found that this approach works remarkably well for pose estimation even under partial occlusion and background clutter. In addition, even the non-optimized version achieves quite high frame rates.

Future work addresses the following points: First, we wish to add different point descriptors and matching methods to the N3Ms in order to make them even more robust to view point changes. Second, we want to add different score functions to the N3Ms in order to exclude all outliers in the self verification step. In addition, we want to speed up our system to use a depth first strategy instead of a breadth first strategy such that it does not search first for all N3Ms before it votes for each N3M. Finally, we want to investigate an alternative voting (pose clustering) process that simplifies the algorithm even more.

6. Conclusion

We have presented a new idea for the automatic learning of 3D sets of feature points for pose estimation. We call these point sets 'Natural 3D Markers', because they define a 3D entity enabling detection and self verification as well as pose estimation. The contribution lies in the learning of such stable and non degenerate feature points sets, the geometric consistency check for these entities and in the multi-level approach for the final pose estimation. Our method has been successfully tested on synthetic images and on real world sequences. Since automatic recognition of 3D subsets of feature points is new, we compared our overall matching/pose estimation performance to the most common alternative approaches for automatic 2D/3D matching and pose estimation. If we take into account at the same time the detection rate, the pose estimation precision and the computational efficiency, our approach outperforms the existing popular alternative methods, namely SIFT or the Randomized Trees followed by RANSAC. This is even more noticeable in the case of partial occlusions and background clutter.

References

[1] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. *European Conf. on Computer Vision*, 2006.

[2] J. F. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[3] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3):313–326, 1992.

[4] M. Fiala. Artag, a fiducial marker system using digital techniques. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 590–596, 2005.

[5] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[6] W. Forstner. A framework for low-level feature extraction. In *European Conf. on Computer Vision*, pages 383–394, 1994.

[7] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. of the 4th Alvey Vision Conf.*, pages 147–151, 1988.

[8] T. Kadir, A. Zisserman, and M. Brady. An affine invariant salient region detector. *European Conf. on Computer Vision*, 2004.

[9] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proc. of the 2nd Int. Workshop on Augmented Reality*, 1999.

[10] Y. Lambdan and H. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. *IEEE Int. Conf. on Computer Vision*, pages 238–249, 1988.

[11] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2006.

[12] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60(2):91–110, 2004.

[13] E. Malis, F. Chaumette, and S. Boudet. 2 1/2 d visual servoing. *IEEE Trans. on Robotics and Automation*, 15(2):234–246, 1999.

[14] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. *British Machine Vision Conf.*, 2002.

[15] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *Int. Journal of Computer Vision*, 60(1):63–86, 2004.

[16] H. Najafi, Y. Genc, and N. Navab. Fusion of 3d and appearance models for fast object detection and pose estimation. *IEEE Asian Conf. on Computer Vision*, pages 415–426, 2006.

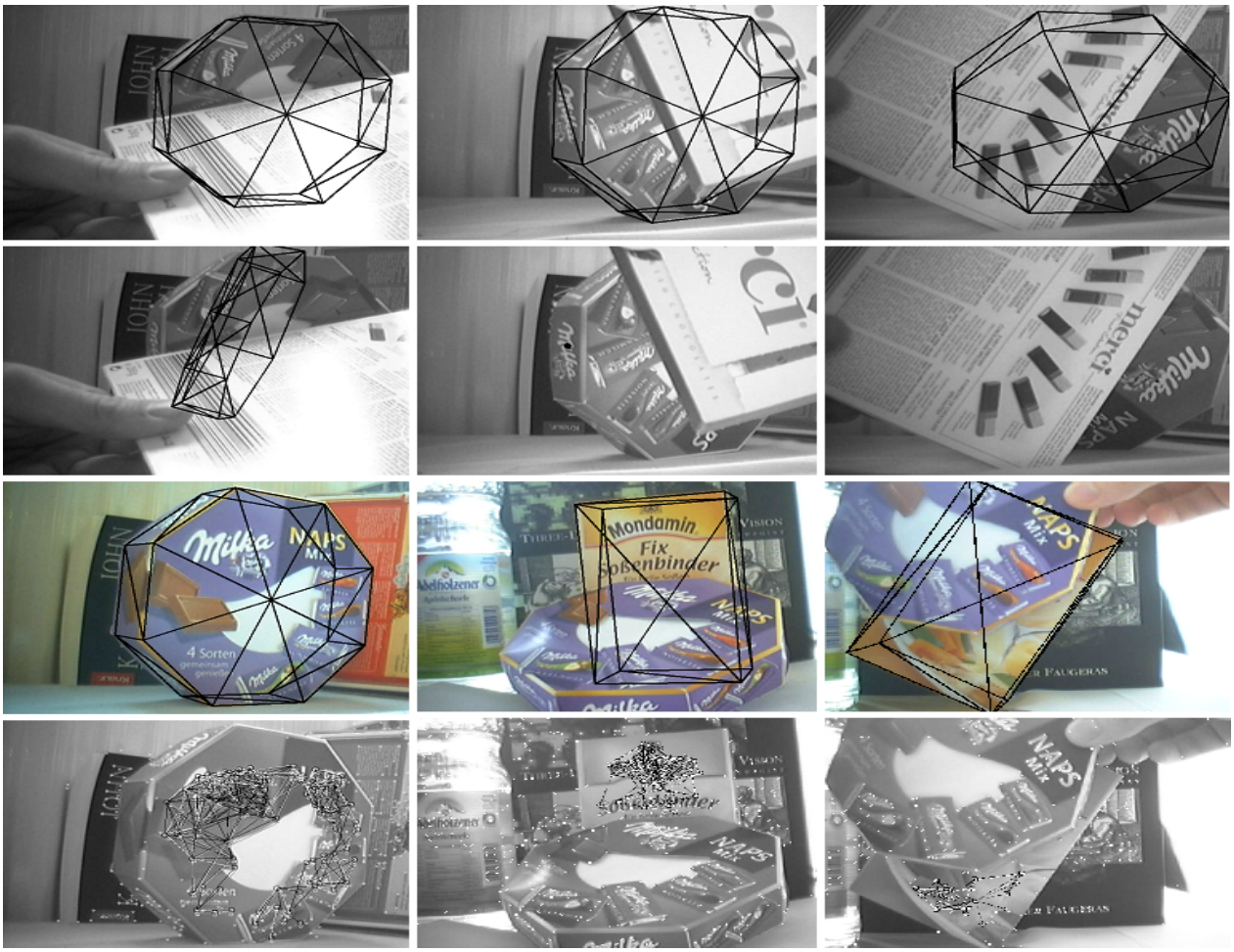

Figure 7. The first row shows real world examples for pose estimation with N3Ms under partial occlusion. The second row shows the same images tested with Randomized Trees combined with RANSAC. The third row shows results of pose estimation with N3Ms with different models. The fourth row shows the corresponding N3Ms detected and used for pose estimation.

[17] M. Özuysal, V. Lepetit, F. Fleuret, and P. Fua. Feature harvesting for tracking-by-detection. *eccv*, pages 592–605, 2006.

[18] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *IEEE Int. Conf. on Computer Vision*, pages 754–760, 1998.

[19] R.Fergus, L.Fei-Fei, P.Perona, and A.Zisserman. Learning object categories from google's image search. *IEEE Int. Conf. on Computer Vision*, 2005.

[20] C. Schmid and R. Mohr. Local grayvalue invariants for image. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(5):530–535, 1997.

[21] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *Int. Journal of Computer Vision*, 37(2):151–172, 2000.

[22] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. *IEEE Int. Conf. on Computer Vision*, 2003.

[23] S. M. Smith and J. M. Brady. Susan - a new approach to low level image processing. *Int. Journal of Computer Vision*, 23:45–78, 1997.

[24] C. Steger. Similarity measures for occlusion, clutter, and illumination invariant object recognition. In B. Radig and S. Florczyk, editors, *Pattern Recognition*, volume 2191 of *Lecture Notes in Computer Science*, pages 148–154, Berlin, 2001. Springer-Verlag.

[25] R. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, 1987.

[26] T. Tuytelaars and L. van Gool. *Matching Widely Separated Views Based on Affine Invariant Regions*. Kluwer Academic Publishers, 2004.

[27] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. Technical Report 2273, INRIA, 1994.