Edge-based Template Matching and Tracking for Perspectively Distorted Planar Objects

Andreas Hofhauser and Carsten Steger and Nassir Navab

TU München, Boltzmannstrasse 3, 85748 Garching bei München, Germany
MVTec Software GmbH, Neherstrasse 1, 81675 München, Germany

Abstract. This paper presents a template matching approach to high accuracy detection and tracking of perspectively distorted objects. To this end we propose a robust match metric that allows significant perspective shape changes. Using a coarse-to-fine representation for the detection of the template further increases efficiency. Once an template is detected at interactive frame-rate, we immediately switch to tracking with the same algorithm, enabling detection times of only 20ms. We show in a number of experiments that the presented approach is not only fast, but also very robust and highly accurate in detecting the 3D pose of planar objects or planar subparts of non-planar objects. The approach is used in augmented reality applications that could up to now not be sufficiently solved, because existing approaches either needed extensive training data, like machine learning methods, or relied on interest point extraction, like descriptors-based methods.

1 Introduction

Methods that exhaustively search a template in an image are one of the oldest computer vision algorithms used to detect an object in an image. However, the mainstream vision community has abandoned the idea of an exhaustive search as there are two prejudices that are commonly articulated. First, that an object detection based on template matching is slow and second, that an object detection based on template matching is certainly extremely inefficient for, e.g., perspective distortions where an 8 dimensional search space must be evaluated. In our work, we address these issues and show that with several contributions it is possible to benefit from the robustness and accuracy of template matching even when an object is perspectively distorted. Furthermore, we show in a number of experiments that it is possible to achieve an interactive rate of detection that was only possible with a descriptor-based approach until now. In fact, if the overall search range of the pattern is restricted, real-time detection is possible on current standard PC hardware. Furthermore, as we have an explicit representation of the geometric search space, we can easily restrict it and therefore use the proposed method for high-speed tracking. This is in strong contrast to many other approaches (e.g. [1]), in which a difficult decision has to be made when to switch between a detection and a tracking algorithm. One of the key contributions of any new template matching algorithm is the image metric that is used

to compare the model template with the image content. The design of this metric determines the overall behavior and its evaluation dominates the run-time. One key contribution of the proposed metric is that we explicitly distinguish between model contours that give us point correspondences and contours that suffer from the aperture problem. This allows us to detect, e.g., an assembly part that contains only curved contours. For these kinds of objects, descriptor-based methods, that excel as they allow for perspective shape changes, notoriously fail if the image contains not enough or only a small set of repetitive texture like in Figure 1.

Fig. 1. An object typically encountered in assembly scenarios. A planar sub-part of a non-planar object is taken as model region and the detection results are depicted as the white contours. The different views leads to significant non-linear illumination changes and perspective distortion. The object contains only curved contours, and hence extraction of discriminative point features is a difficult task.

1.1 Related Work

We roughly classify algorithms for pose detection into template matching and descriptor-based methods. In the descriptor-based category, the rough scheme is to first determine discriminative "high-level" features, extract surrounding discriminative descriptors from these feature points, and to establish the correspondence between model and search image by classifying the descriptors. The big advantage of this scheme is that the run-time of the algorithm is independent of the degree of the geometric search space. Recent prominent examples, that fall into this category are [2–5]. While showing outstanding performance in several scenarios, they fail if the object has only highly repetitive texture or only sparse edge information. The feature descriptors overlap in the feature space and are not discriminating anymore.
In the template matching category, we subsume algorithms that perform an explicit search. Here, a similarity measure that is either based on intensities (like

SAD, SSD, NCC and mutual information) or gradient features is evaluated. However, the evaluation of intensity-based metrics is computationally expensive. Additionally, they are typically not invariant against nonlinear illumination changes, clutter, or occlusion.
For the case of feature-based template matching, only a sparse set of features between template and search image is compared. While extremely fast and robust if the object undergoes only rigid transformations, these methods become intractable for a large number of degrees of freedom, e.g., when an object is allowed to deform perspectively. Nevertheless, one approach for feature-based deformable template matching is presented in [6], where the final template is chosen from a learning set while the match metric is evaluated. Because obtaining a learning set and applying a learning step is problematic for our applications, we prefer to not rely on training data except for the original template. In contrast to this, we use a match metric that allows for local perspective deformations, while preserving robustness to illumination changes, partial occlusion and clutter. While we found a match metric with normalized directed edge points in [7, 8] for rigid object detection, and also for articulated object detection in [9], its adaptation to 3D object detection is new. A novelty in the proposed approach is that the search method takes all search results for all parts into account at the same time. Despite the fact that the model is decomposed into sub-parts, the relevant size of the model that is used for the search at the highest pyramid level is not reduced. Hence, the presented method does not suffer the speed limitations of a reduced number of pyramid levels that prior art methods have. This is in contrast to, e.g., a component-based detection like in [9] which could conceptually also be adapted for the perspective object detection. Here small sub-parts must be detected, leading to a lower number of pyramid levels that can be used to speed up the search.

2 Perspective Shape-Based Object Detection

In the following, we detail the perspective shape-based model generation and matching algorithm. The problem that this algorithm solves is particularly difficult, because in contrast to optical flow, tracking, or medical registration, we assume neither temporal nor local coherence. While the location of the objects are determined with the robustness of a template matching method, we avoid the necessity of expanding the full search space as if it was a descriptor-based method.

2.1 Shape Model Generation

For the generation of our model, we decided to rely on the result of a simple contour edge detection. This allows us to represent objects from template images as long as there is any intensity change. Note that in contrast to corners or interest **point** features, we can model objects that contain only curved **contours** (see detection results in Figure 1). Furthermore, directly generating a model

from an untextured CAD format is possible in principle. Our shape model M is composed of an unordered set of edge points

$$M = \left\{ x_i, y_i, d_i^m, c_{ji}, p_j | i = 1 \dots n, j = 1 \dots k \right\}. \tag{1}$$

Here, x and y are the row and column coordinates of the n model points. d^m denotes the gradient direction vector at the respective row and column coordinate of the template. We assume that spatially coherent structures stay the same even after a perspective distortion. Therefore, we cluster the model points with expectation-maximization-based k-means such that every model point belongs to one of k clusters. The indicator matrix c_{ji} maps clusters to model points (entry 0 if not in cluster, else entry 1) that allow us to access the model points of each cluster efficiently at run-time. For the later detection we have to distinguish, whether a cluster of a model can be used as a point feature (that gives us two equations for the x and y location) or only as a contour line feature (that suffers from the aperture problem and gives only one equation). This is a label that we save in for each cluster in p_j. We detect this by analyzing whether a part contains only one dominant gradient direction or gradient directions into various, e.g., perpendicular directions. For this we determine a feature that describes the main gradient directions for each of the j clusters of the model:

$$ClusterDirection_j = \left\| \frac{\sum_{i=1}^{n} \frac{d_i^m}{\|d_i^m\|} c_{ji}}{\sum_{i=1}^{n} c_{ji}} \right\|. \tag{2}$$

If a model part contains only direction vectors in one dominant direction, the length of the resulting $ClusterDirection$ vector has the value one or, in case of noise, slightly below one. In all other cases, e.g., a corner-like shape or directions of opposing gradient directions, the length of $ClusterDirection$ is significantly smaller than one. For a straight edge with a contrast change, the sign change of the gradient polarity gives us one constraint and hence prevents movement of that cluster along the edge and therefore we assign it a point feature label. Because the model generation relies on only one image and the calculation of the clusters is realized efficiently, this step needs, even for models with thousands of points, less than a second. This is an advantage for users of a computer vision system, as an extended offline phase would make the use of a model generation algorithm cumbersome.

2.2 Metric Based on Local Edge Patches

Given the generated model, the task of the perspective shape matching algorithm is to extract instances of that model in new images. Therefore, we adapted the match metric of [8]. This match metric is designed such that it is inherently invariant against nonlinear illumination changes, partial occlusion and clutter. If a location of an object is described by x, y (this is 2D translation only, but the formulas can easily be extended for, e.g., 2D affine transformations), the score

function for rigid objects reads as follows:

$$s(x,y) = \frac{1}{n}\sum_{i=1}^{n} \frac{\langle d_i^m, d^s_{(x+x_i,y+y_i)}\rangle}{\|d_i^m\| \cdot \|d^s_{(x+x_i,y+y_i)}\|}, \tag{3}$$

where d^s is the direction vector in the search image, $\langle\cdot\rangle$ is the dot product and $\|\cdot\|$ is the Euclidean norm. The point set of the model is compared to a dense gradient direction field of the search image. Even with significant nonlinear illumination changes that propagate to the gradient amplitude the gradient direction stays the same. Furthermore, a hysteresis threshold or non-maximum suppression is completely avoided in the search image, resulting in true invariance against arbitrary illumination changes.[1] Partial occlusion, noise, and clutter results in random gradient directions in the search image. These effects lower the maximum of the score function but do not alter its location. Hence, the semantic meaning of the score value is the ratio of matching model points. It is interesting to note that comparing the cosine between the gradients leads to the same result, but calculating this formula with dot products is several orders of magnitudes faster.

The idea of extending this metric for 3D object detection is that we instantiate globally only similarity transformations. By allowing successive small movements of the parts of the model, we implicitly evaluate a much higher class of nonlinear transformations, like perspective distortions. Following this argument, we distinguish between an explicit global score function s_g, which is evaluated for, e.g., affine 2D transformations[2], and a local implicit score function s_l, that allows for local deformations. The global score function s_g is a sum of the contributions of all the clusters.

$$s_g(x,y) = \frac{1}{n}\sum_{j=1}^{k} s_l(x,y,j). \tag{4}$$

We assume that even after a perspective distortion the neighborhood of each model point stays the same and is approximated by a local euclidean transformation. Hence, we instantiate local euclidean transformations T for each cluster and apply it on the model points of that cluster in a small local neighborhood. If the cluster is a point feature we search in a 5×5 pixel window the optimal score. In case of a line feature, we search a 2 pixels in both directions of $ClusterDirection_j$ of the respective cluster. The local score then is the maximum alignment of gradient direction between the locally transformed model points of each cluster and the search image. Accordingly, the proposed local

[1] Homogenous regions and regions that are below a minimum contrast change (e.g., less than 3 gray values) can optionally be discarded, as they give random directions that is due to noise. However, this is not needed conceptually, but gives a small speedup.

[2] For sake of clarity we write formulas only for 2D translation. They can easily be extended for, e.g., 2D rotation, scaling, and anisotropic scaling, as is done in our implementation.

score function s_l is:

$$s_l(x,y,j) = \max_T \sum_{i=1}^{size(j)} \frac{\langle d^m_{c_{ji}}, d^s_{(x+T(x_{c_{ji}}),y+T(y_{c_{ji}}))} \rangle}{\|d^m_{c_{ji}}\| \cdot \|d^s_{(x+T(x_{c_{ji}}),y+T(y_{c_{ji}}))}\|} \tag{5}$$

Here, the function *size* returns the number of elements in cluster j. For the sake of efficiency, we exploit the mapping that was generated in the offline phase for accessing the points in each cluster (the c_{ji} matrix). Furthermore, we cache $T(x_{c_{ji}})$ and $T(y_{c_{ji}})$ since they are independent of x and y.

2.3 Perspective Shape Matching

After defining an efficient score function that tolerates shape changes, we integrated it into a general purpose object detection system. We decided to alter the conventional template matching algorithm such that it copes with perspectively distorted planar 3D objects.

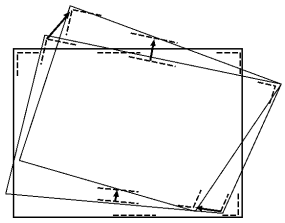

Fig. 2. The schematic depiction of the shape matching algorithm. The original model consists of the rectangle. The first distorted quadrilateral is derived from the parent of the hypothesis. The local displacements T, depicted as arrows bring the warped template to a displaced position and the fitted homography aligns the parts of the model again.

Hence, the perspective shape matching algorithm first extracts an image pyramid of incrementally zoomed down versions of the original search image. At the highest pyramid level, we extract the local maxima of the score s_g function (4). The local maxima of s_g are then tracked through the image pyramid. until either the lowest pyramid level is reached or no match candidate is above a certain score value. While tracking the candidates down the pyramid, a rough alignment was already extracted during evaluation of the current candidate's parent on a higher pyramid level. Therefore, we first use the alignment originating from the candidate's parent to warp the model. Now, starting from this warped candidate the local transformation T that give the maximal score give a locally optimal displacement of each cluster to image. Since we are interested in perspective distortions of the whole objects, we fit a homography with the normalized DLT algorithm [10] to the locations of the original cluster centers to

the displaced cluster centers given T. Depending whether the clusters have been classified as point features or as line features each correspondence give as two or one equation in the DLT matrix[3] Then we iteratively warp the whole model with the extracted update and refine the homography on each pyramid level until the update becomes near to identity or a maximal step of iterations is reached. Up to now, the displacement of each part T is discretized up to pixel resolution. However, as the total pose of the object is determined by the displacements of many clusters, we typically obtain a very precise position of the objects. To reach a high accuracy and precision that is a requirement in many computer vision applications, we extract subpixel-precise edge points in the search image and determine correspondences for each model point. Given the subpixel precise correspondences, the homography is again iteratively updated until convergence. Here, we minimize the distance of the tangent of the model points to the subpixel precise edge point in the image. Hence, each model edge point gives as one equation since it is a line feature.

2.4 Perspective Shape Tracking

Once an template is found in an image sequence, we restrict the search space for the subsequent frames. This assumption is valid in, e.g., many augmented reality scenarios in which the inter-frame rotations can be assumed to be less then 45 degrees and the scaling change of the object be less than 0.8 to 1.2. Once, a track is lost we switch back to detection by expanding the search range to the original size. It is important to note that we use the same algorithm for tracking and detection and only parametrize it differently. Hence, we exploit prior knowledge to speed up the search if it is available. However, our template matching is not restricted to tracking alone, like e.g. [11], but can be used for object detection when needed.

3 Experiments

For the evaluation of the proposed object detection and tracking algorithm, we conducted experiments under various real world conditions.

3.1 Benchmark Test Set

For comparison with other approaches we used sample images from publicly available benchmark datasets[4] (see Figure 3). The graffiti sequence is used for

[3] Typically, the DLT equations are highly overdetermined. We evaluated a solution with the SVD as described in [10] or directly with the eigenvalue decomposition of the normal equation. As there is no noticeable robustness difference, despite the expected quadratic worse condition number for the solution with the eigenvalue decomposition, but the version with the eigenvalue decomposition is faster we use in the following discussion the solution with the normal equations. To give an impression of the difference, in one example sequence the whole object detection with SVD runs in 150 ms, with eigenvalue decomposition in 50 ms.

[4] *http://www.robots.ox.ac.uk/~vgg/data/data-aff.html*

Fig. 3. Benchmark data set with detected template used for evaluations of the method.

instance to evaluate how much perspective distortion a descriptor-based approach can tolerate. The last two depicted images are a challenge for many descriptor-based approaches.

Another interesting comparison is with the phone test sequence provided in [12]. Here, Lukas-Kanade, SIFT and the method of [12] are reported to sometimes loose the object. The proposed algorithm is able to process the sequence at 60 ms without once loosing the object. We think this is due to the fact that we explicitly represent contour information and not just interest point features and because we are able to exhaustively search for the object.

3.2 Industrial Robot Experiments

Fig. 4. Sample images used for different real world experimental evaluations. The fitted edge model points are depicted.

To evaluate the accuracy of the 3D object detection, we equipped a 6 axis industrial robot with a calibrated camera (12 mm lens 640x480 pixel) mounted at its gripper. Then we taught the robot a sequence of different poses where the camera–object distance changes in the range of 30-50 cm and significant latitude changes must be compensated. First, we determined the repeatability of the robot, to prevent a drift during different experiment runs. Therefore, we made the robot drive the sequence several times and determined the pose of the camera at different runs with an industrial calibration grid that is seen by the

camera. The maximal difference between the poses of different runs was 0.0009 distance error between estimated and ground truth translational position and 0.04 degrees angle to bring the rotation from the estimated to the true pose. Then, we manually placed an object at the same place as the calibration grid and used the planar shape matching with a bundle adjustment pose estimation to determine the pose of the object (see Figure 4). The maximal difference to the poses that where extracted with the calibration grid was below 0.01 normalized distance and 0.4 degree angle. It is interesting to note that the remaining pose error is dominated by the depth uncertainty. The maximal errors are measured, when the images suffer severe illumination changes or are out of focus. When these situations are prevented the error of translation is below 0.005 normalized distance and the angle error below 0.2 degree angle.

3.3 Registering an Untextured Building

Fig. 5. Sample images from a longer sequence used in the experimental evaluation. The first image is used for generating the template of the house. Further movies can be viewed on the web site: *http://campar.in.tum.de/Main/AndreasHofhauser.*

To evaluate the method for augmented reality scenarios we used the proposed approach to estimate the position of a building that is imaged by a shaking hand-held camera (see Figure 5). Despite huge scale changes, motion blur and parallax effects the facade of the building is robustly detected and tracked. We think that this is a typical sequence for mobile augmented reality applications, e.g. the remote expert situation. Furthermore, due to the fast model generation, the approach is particularly useful for, e.g., mobile navigation, in which the template for the object detection must be generated instantly.

To sum up the experimental evaluation, we think that the result are very encouraging in terms of speed, robustness and accuracy compared to other methods

that have been published in literature. Applications that up to now relied on interest points can replace their current object detection without the disadvantage of smaller robustness with regards to e.g. attitude change. Further, these applications can benefit of a bigger robustness and accuracy particularly for detecting untextured objects. Since the acquisition of the test sequences was a time consuming task and to stimulate further research and comparisons we will make the test data available upon request.

4 Conclusion

In this paper we presented a single-camera solution for planar 3D template matching and tracking that can be utilized in a wide range of applications. For this, we extended an already existing edge polarity based match metric for tolerating local shape changes. In an extensive evaluation we showed the applicability of the method for several computer vision scenarios.

References

1. Ladikos, A., Benhimane, S., Navab, N.: A real-time tracking system combining template-based and feature-based approaches. In: International Conference on Computer Vision Theory and Applications. (2007)
2. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision (2004)
3. Berg, A., Berg, T., Malik, J.: Shape matching and object recognition using low distortion correspondences. In: Conference on Computer Vision and Pattern Recognition, San Diego, CA. (2005)
4. Pilet, J., Lepetit, V., Fua, P.: Real-time non-rigid surface detection. In: Conference on Computer Vision and Pattern Recognition, San Diego, CA. (2005)
5. Özuysal, M., Fua, P., Lepetit, V.: Fast keypoint recognition in ten lines of code. In: Conference on Computer Vision and Pattern Recognition. (2007)
6. Gavrila, D.M., Philomin, V.: Real-time object detection for "smart" vehicles. In: 7th International Conference on Computer Vision. Volume I. (1999) 87–93
7. Olson, C.F., Huttenlocher, D.P.: Automatic target recognition by matching oriented edge pixels. IEEE Transactions on Image Processing **6** (1997) 103–113
8. Steger, C.: Occlusion, clutter, and illumination invariant object recognition. In Kalliany, R., Leberl, F., eds.: International Archives of Photogrammetry, Remote Sensing, and Spatial Information Sciences. Volume XXXIV, part 3A., Graz (2002) 345–350
9. Ulrich, M., Baumgartner, A., Steger, C.: Automatic hierarchical object decomposition for object recognition. In: International Archives of Photogrammetry and Remote Sensing. Volume XXXIV, part 5. (2002) 99–104
10. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2000)
11. Benhimane, S., Malis, E.: Homography-based 2d visual tracking and servoing. Special Joint Issue IJCV/IJRR on Robot and Vision. Published in The International Journal of Robotics Research **26** (2007) 661–676
12. K.Zimmermann, J.Matas, T.: Tracking by an optimal sequence of linear predictors. Transactions on Pattern Analysis and Machine Intelligence (to appear) (2008)