CVPR 2010

# Gradient Response Maps for Real-Time Detection of Texture-Less Objects

Stefan Hinterstoisser, Cedric Cagniart, *Student Members, IEEE,* Slobodan Ilic, Peter Sturm,
Nassir Navab, Pascal Fua, *Members, IEEE,* and Vincent Lepetit

**Abstract**—We present a method for real-time 3D object instance detection that does not require a time consuming training stage, and can handle untextured objects. At its core, our approach is a novel image representation for template matching designed to be robust to small image transformations. This robustness is based on spread image gradient orientations and allows us to test only a small subset of all possible pixel locations when parsing the image, and to represent a 3D object with a limited set of templates. In addition, we demonstrate that if a dense depth sensor is available we can extend our approach for an even better performance taking also 3D surface normal orientations into account. We show how to take advantage of the architecture of modern computers to build an efficient but very discriminant representation of the input images that can be used to consider thousands of templates in real-time. We demonstrate in many experiments on real data that our method is much faster and more robust with respect to background clutter than current state-of-the-art methods.

**Index Terms**—Computer Vision, Real-Time Detection and Object Recognition, Tracking, Multi-Modality Template Matching

REAL-TIME object instance detection and learning are two important and challenging tasks in Computer Vision. Among the application fields that drive development in this area, robotics especially has a strong need for computationally efficient approaches, as autonomous systems continuously have to adapt to a changing and unknown environment, and to learn and recognize new objects.

For such time-critical applications, real-time template matching is an attractive solution because new objects can be easily learned and matched online, in contrast to statistical-learning techniques that require many training samples and are often too computationally intensive for real-time performance [1], [2], [3], [4], [5]. The reason for this inefficiency is that those learning approaches aim at detecting unseen objects from certain object classes instead of detecting *a priori* known object instances from multiple viewpoints. The latter is tried to be achieved in classical template 好 matching where generalization is not performed on the object class but on the viewpoint sampling. While this is considered as an easier task, it does not make the problem trivial, as the data still exhibit significant changes in viewpoint, in illumination and in occlusion between the training and the runtime sequence.

前者对象泛化，后者视点泛化

看起来简单，因为数据在视角、光照以及训练和运行时序之间的遮挡方面仍显示出显著变化

When the object is textured enough for keypoints to be found and recognized on the basis of their appearance, this difficulty has been successfully addressed by defining patch descriptors that can be computed quickly and used to characterize the object [6]. However, this kind of approach will fail on texture-less objects such as those of Fig. 1, whose appearance is often dominated by their projected contours.

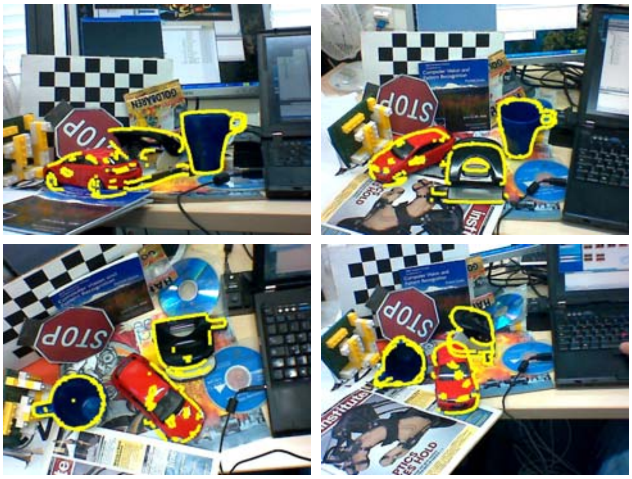


Fig. 1: Our method can detect texture-less 3D objects in real-time under different poses over heavily cluttered background using gradient orientation.

To overcome this problem, we propose a novel approach based on real-time template recognition for rigid 3D object instances, where the templates can both be built and matched very quickly. We will show that this makes it very easy and virtually instantaneous to learn new incoming objects by simply adding new templates to the database while maintaining reliable real-time recognition.

几乎是传入和瞬间完成学习

However, we also wish to keep the efficiency and robustness of statistical methods, as they learn how to reject unpromising image locations very quickly and

- S. Hinterstoisser, C. Cagniart, S. Ilic and N. Navab are with the Department of Computer Aided Medical Procedures (CAMP), Technische Universität München, Garching bei München, Germany, 85478.
E-mail: {hinterst,cagniart,Slobodan.Ilic,navab}@in.tum.de
- V. Lepetit and P. Fua are with the Computer Vision Lab (CVLAB), Ecole Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland.
E-mail: {vincent.lepetit,pascal.fua}@epfl.ch
- P. Sturm is with the STEEP Team, INRIA Grenoble–Rhône-Alpes, 38334 Saint-Ismier Cedex, France
E-mail: Peter.Sturm@inrialpes.fr

这个representation具有小平移和变形不变性

tend to be very robust, because they can generalize well from the training set. We therefore propose a new image representation that holds local image statistics and is fast to compute. It is designed to be invariant to small translations and deformations of the templates, which has been shown to be a key factor to generalization to different view-points of the same object [6]. In addition, it allows us to quickly parse the image by skipping many locations without loss of reliability.

Our approach is related to recent and efficient template matching methods [7], [8] which consider only images and their gradients to detect objects. As such, they work even when the object is not textured enough to use feature point techniques, and learn new objects virtually instantaneously. In addition, they can directly provide a coarse estimation of the object pose which is especially important for robots which have to interact with their environment. However, similar to previous template matching approaches [9], [10], [11], [12], they suffer severe degradation of performance or even failure in the presence of strong background clutter such as the one displayed in Fig. 1.

We therefore propose a new approach that addresses this issue while being much faster for larger templates. Instead of making the templates invariant to small deformations and translations by considering dominant orientations only as in [7], we build a representation of the input images which has similar invariance properties but consider all gradient orientations in local image neighborhoods. Together with a novel similarity measure, this prevents problems due to too strong gradients in the background as illustrated by Fig. 1. 防止了由于图一所示的背景中梯度太大而导致的问题

To avoid slowing down detection when using this finer method, we have to make careful considerations about how modern CPUs work. A naive implementation would result in many "memory cache misses", which slow down the computations, and we thus show how to structure our image representation in memory to prevent these and to additionally exploit heavy SSE parallelization. We consider this as an important contribution: Because of the nature of the hardware improvements, it is not guaranteed anymore that legacy code will run faster on the new versions of CPUs [13]. This is particularly true for Computer Vision, which algorithms are often computationally expensive. It is now required to take the CPU architecture into account, which is not an easy task.

For the case where a dense depth sensor is available we describe an extension of our method where additional depth data is used to further increase the robustness by simultaneously leveraging the information of the 2D image gradients and 3D surface normals. We propose a method that robustly computes 3D surface normals from dense depth maps in real-time, making sure to preserve depth discontinuities on occluding contours and to smooth out discretization noise of the sensor. The 3D normals are then used together with the image gradients and in a similar way.

In the remainder of the paper, we first discuss related work before we explain our approach. We then discuss the theoretical complexity of our approach. We finally present experiments and quantitative evaluations for challenging scenes.

## 1 Related Work

Template Matching has played an important role in tracking-by-detection applications for many years. This is due to its simplicity and its capability to handle different types of objects. It neither needs a large training set nor a time-consuming training stage, and can handle low-textured or texture-less objects, which are, for example, difficult to detect with feature points-based methods [6], [14]. Unfortunately, this increased robustness often comes at the cost of an increased computational load that makes naïve template matching inappropriate for real-time applications. So far, several works have attempted to reduce this complexity.

An early approach to Template Matching [12] and its extension [11] include the use of the Chamfer distance between the template and the input image contours as a dissimilarity measure. For instance, Gavrila and Philomin [11] introduced a coarse-to-fine approach in shape and parameter space using Chamfer Matching [9] on the Distance Transform of a binary edge image. The Chamfer Matching minimizes a generalized distance between two sets of edge points. Although being fast when using the Distance Transform (DT), the disadvantage of the Chamfer Transform is its sensitivity to outliers which often result from occlusions.

Another common measure on binary edge images is the Hausdorff distance [15]. It measures the maximum of all distances from each edge point in the image to its nearest neighbor in the template. However, it is sensitive to occlusions and clutter. Huttenlocher *et al.* [10] tried to avoid that shortcoming by introducing a generalized Hausdorff distance which only computes the maximum of the $k$-th largest distances between the image and the model edges and the $l$-th largest distances between the model and the image edges. This makes the method robust against a certain percentage of occlusions and clutter. Unfortunately, a prior estimate of the background clutter in the image is required but not always available. Additionally, computing the Hausdorff distance is computationally expensive and prevents its real-time application when many templates are used.

Both Chamfer Matching and the Hausdorff distance can easily be modified to take the orientation of edge points into account. This drastically reduces the number of false positives as shown in [12], but unfortunately also increases the computational load.

[16] is also based on the Distance Transform, however, it is invariant to scale changes and robust enough against planar perspective distortions to do real-time matching. Unfortunately, it is restricted to objects with closed contours, which are not always available.

All these methods use binary edge images obtained with a contour extraction algorithm, using the Canny detector [17] for example, and they are very sensitive to illumination changes, noise and blur. For instance, if the image contrast is lowered, the number of extracted edge pixels progressively decreases which has the same effect as increasing the amount of occlusion.

The method proposed in [18] tries to overcome these limitations by considering the image gradients in contrast to the image contours. It relies on the dot product as a similarity measure between the template gradients and those in the image. Unfortunately, this measure rapidly declines with the distance to the object location, or when the object appearance is even slightly distorted. As a result, the similarity measure must be evaluated densely and with many templates to handle appearance variations, making the method computationally costly. Using image pyramids provides some speed improvements, however, fine but important structures tend to be lost if one does not carefully sample the scale space.

Contrary to the above mentioned methods, there are also approaches addressing the general visual recognition problem: they are based on statistical learning and aim at detecting object categories rather than *a priori* known object instances. While they are better at category generalization, they are usually much slower during learning and runtime which makes them unsuitable for online applications.

For example, Amit *et al.* [19] proposed a coarse to fine approach that spreads gradient orientations in local neighborhoods. The amount of spreading is learned for each object part in an initial stage. While this approach — used for license plate reading — achieves high recognition rates, it is not real-time capable.

Histogram of Gradients (HoG) [1] is another related and very popular method. It statistically describes the distribution of intensity gradients in localized portions of the image. The approach is computed on a dense grid with uniform intervals and uses overlapping local histogram normalization for better performance. It has proven to give reliable results but tends to be slow due to the computational complexity.

Ferrari *et al.* [4] provided a learning based method that recognizes objects via a Hough-style voting scheme with a non-rigid shape matcher on object boundaries of a binary edge image. The approach applies statistical methods to learn the model from few images that are only constrained within a bounding box around the object. While giving very good classification results, the approach is neither appropriate for object tracking in real-time due to its expensive computation nor is it precise enough to return the accurate pose of the object. Additionally, it is sensitive to the results of the binary edge detector, an issue that we discussed before.

Kalal *et al.* [20] very recently developed an online learning-based approach. They showed how a classifier can be trained online in real-time, with a training set generated automatically. However, as we will see in the experiments, this approach is only suitable for smooth background transitions and not appropriate to detect known objects over unknown backgrounds.

Opposite to the above mentioned learning based methods, there are also approaches that are specifically trained on different viewpoints. As with our template-based approach, they can detect objects under different poses but typically require a large amount of training data and a long offline training phase. For example, in [5], [21], [22], one or several classifiers are trained to detect faces or cars under various views.

More recent approaches for 3D object detection are related to object class recognition. Stark *et al.* [23] rely on 3D CAD models and generate a training set by rendering them from different viewpoints. Liebelt and Schmid [24] combine a geometric shape and pose prior with natural images. Su *et al.* [25] use a dense, multiview representation of the viewing sphere combined with a part-based probabilistic representation. While these approaches are able to generalize to the object class they are not real-time capable and require expensive training.

From the related works which also take into account depth data there are mainly approaches related to pedestrian detection [26], [27], [28], [29]. They use three kinds of cues: image intensity, depth and motion (optical flow). The most recent approach of Enzweiler *et. al* [26] builds part-based models of pedestrians in order to handle occlusions caused by other objects and not only self occlusions modeled in other approaches [27], [29]. Besides pedestrian detection, there has been an approach to object classification, pose estimation and reconstruction introduced by [30]. The training data set is composed of depth and image intensities while the object classes are detected using the modified Hough transform. While being quite effective in real applications these approaches still require exhaustive training using large training data sets. This is usually prohibited in robotic applications where the robot has to explore an unknown environment and learn new objects online.

As mentioned in the introduction, we recently proposed a method to detect texture-less 3D object instances from different viewpoints based on templates [7]. Each object is represented as a set of templates, relying on local dominant gradient orientations to build a representation of the input images and the templates. Extracting the dominant orientations is useful to tolerate small translations and deformations.

It is fast to perform and most of the time discriminant enough to avoid generating too many false positive detections.

However, we noticed that this approach degrades significantly when the gradient orientations are disturbed by stronger gradients of different orientations coming from background clutter in the input images. In practice, this often happens in the neighborhood of the silhouette of an object, which is unfortunate as the silhouette is a very important cue especially for texture-less objects. The method we propose in this paper does not suffer from this problem while running at the same speed. Additionally, we show how to extend our approach to handle 3D surface normals at the same time if a dense depth sensor like the Kinect is available. As we will see this increases the robustness significantly.

## 2 Proposed Approach

In this section, we describe our template representation and show how a new representation of the input image can be built and used to parse the image to quickly find objects. We will start by deriving our similarity measure, emphasizing the contribution of each aspect of it. We also show how we implement our approach to efficiently use modern processor architectures. Additionally, we demonstrate how to integrate depth data to increase robustness if a dense depth sensor is available.

### 2.1 Similarity Measure

非相似度测量

Our unoptimized similarity measure can be seen as the measure defined by Steger in [18] modified to be robust to small translations and deformations. Steger suggests to use:

$$\mathcal{E}_{\text{Steger}}(\mathcal{I},\mathcal{T},c)=\sum_{r\in\mathcal{P}}\left|\cos(\text{ori}(\mathcal{O},r)-\text{ori}(\mathcal{I},c+r))\right| \ , \tag{1}$$

where $\text{ori}(\mathcal{O},r)$ is the gradient orientation in radians at location $r$ in a reference image $\mathcal{O}$ of an object to detect. Similarly, $\text{ori}(\mathcal{I},c+r)$ is the gradient orientation at $c$ shifted by $r$ in the input image $\mathcal{I}$. We use a list, denoted by $\mathcal{P}$, to define the locations $r$ to be considered in $\mathcal{O}$. This way we can deal with arbitrarily shaped objects efficiently. A template $\mathcal{T}$ is therefore defined as a pair $\mathcal{T}=(\mathcal{O},\mathcal{P})$.

Each template $\mathcal{T}$ is created by extracting a small set of its most discriminant gradient orientations from the corresponding reference image as shown in Fig. 2 and by storing their locations. To extract the most discriminative gradients we consider the strength of their norms. In this selection process, we also take the location of the gradients into account to avoid an accumulation of gradient orientations in one local area of the object while the rest of the object is not sufficiently described. If a dense depth sensor is available, we can extend our approach with 3D surface normals as shown on the right side of Fig. 2.

考虑梯度的位置，避免同一个位置区域有太多个同一方向梯度的累积，让剩余位置没有得到充分描述

强壮有区分度的梯度主要分布在图像边缘上，用ri表示梯度

如果有密集数据的深度信息，可以计算法线方向，用rk表示法线

梯度和法线结合，梯度计算容易出现在物体轮廓上，而表面法线计算则出现在物体内部。两者互补，所以将两个modality结合起来

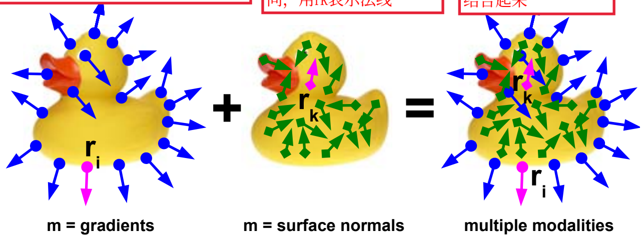


Fig. 2: A toy duck with different modalities. **Left:** Strong and discriminative image gradients are mainly found on the contour. The gradient location $r_i$ is displayed in pink. **Middle:** If a dense depth sensor is available we can also make use of surface 3D normals which are mainly found on the body of the duck. The normal location $r_k$ is displayed in pink. **Right:** The combination of 2D image gradients and 3D surface normals leads to an increased robustness (see Section 2.9). This is due to the complementarity of the visual cues: gradients are usually found on the object contour while surface normals are found on the object interior

Considering only the gradient orientations and not their norms makes the measure robust to contrast changes, and taking the absolute value of the cosine allows it to correctly handle object occluding boundaries: It will not be affected if the object is over a dark background, or a bright background.

The similarity measure of Eq. (1) is very robust to background clutter, but not to small shifts and deformations. A common solution is to first quantize the orientations and to use local histograms like in SIFT [6] or HoG [1]. However this can be unstable when strong gradients appear in the background. In DOT [7], we kept the dominant orientations of a region. This was faster than building histograms but suffers from the same instability. Another option is to apply Gaussian convolution to the orientations like in DAISY [31], but this would be too slow for our purpose.

We therefore propose a more efficient solution. We introduce a similarity measure that, for each gradient orientation on the object, searches in a neighborhood of the associated gradient location for the most similar o[illegible]t image. This can be formalized as:

在相关梯度位置的邻域内搜索输入图像中最相似的方向

$$\mathcal{E}(\mathcal{I},\mathcal{T},c)=\sum_{r\in\mathcal{P}}\left(\max_{t\in\mathcal{R}(c+r)}\left|\cos(\text{ori}(\mathcal{O},r)-\text{ori}(\mathcal{I},t))\right|\right) \ , \tag{2}$$

where $\mathcal{R}(c+r)=\left[c+r-\frac{T}{2},c+r+\frac{T}{2}\right]\times\left[c+r-\frac{T}{2},c+r+\frac{T}{2}\right]$ defines the neighborhood of size $T$ centered on location $c+r$ in the input image. Thus, for each gradient we align the local neighborhood exactly to the associated gradient location whereas in DOT, the gradient orientation is adjusted only to some regular grid. We show below how to compute this measure efficiently.

## 2.2 Computing the Gradient Orientations

方向

Before we continue with our approach, we shortly discuss why we use gradient orientations and how we extract them easily.

We chose to consider image gradients because they proved to be more discriminant than other forms of representations [6], [18] and are robust to illumination change and noise. Additionally, image gradients are often the only reliable image cue when it comes to texture-less objects. Considering only the orientation of the gradients and not their norms makes the measure robust to contrast changes, and taking the absolute value of cosine between them allows it to correctly handle object occluding boundaries: It will not be affected if the object is over a dark background, or a bright background.

比其他形式的representation更有辨别力

设计无纹理对象时候，图像梯度通常是唯一可靠的线索

To increase robustness, we compute the orientation of the gradients on each color channel of our input image separately and for each image location use the gradient orientation of the channel whose magnitude is largest as done in [1] for example. Given an RGB color image $\mathcal{I}$, we compute the gradient orientation map $\mathcal{I}_\mathcal{G}(x)$ at location $x$ with

$$\mathcal{I}_\mathcal{G}(x) = \text{ori}(\hat{\mathcal{C}}(x)) \qquad (3)$$

where

$$\hat{\mathcal{C}}(x) = \underset{\mathcal{C}\in\{R,G,B\}}{\arg\max} \left\| \frac{\partial \mathcal{C}}{\partial x} \right\| \qquad (4)$$

and $R, G, B$ are the RGB channels of the corresponding color image.

In order to quantize the gradient orientation map we omit the gradient direction, consider only the gradient orientation and divide the orientation space into $n_0$ equal spacings as shown in Fig. 3. To make the quantization robust to noise, we assign to each location the gradient whose quantized orientation occurs most often in a $3 \times 3$ neighborhood. We also keep only the gradients whose norms are larger than a small threshold. The whole unoptimized process takes about 31ms on the CPU for a VGA image.

把3X3邻域内的梯度也考虑进来，只保存梯度值大于设定阈值的数

## 2.3 Spreading the Orientations

In order to avoid evaluating the max operator in Eq. (2) every time a new template must be evaluated against an image location, we first introduce a new binary representation — denoted by $\mathcal{J}$ — of the gradients around each image location. We will then use this representation together with lookup tables to efficiently precompute these maximal values.

01二进制表示法：J

The computation of $\mathcal{J}$ is depicted in Fig. 4. We first quantize orientations into a small number of $n_o$ values as done in previous approaches [6], [1], [7]. This allows us to "spread" the gradient orientations $\text{ori}(\mathcal{I}, t)$ of the input image $\mathcal{I}$ around their locations to obtain a new representation of the original image.

For efficiency, we encode the possible combinations of orientations spread to a given image location $m$ using a binary string: Each individual bit of this string corresponds to one quantized orientation, and is set to 1 if this orientation is present in the neighborhood of $m$. The strings for all the image locations form the image $\mathcal{J}$ on the right part of Fig. 4. These strings will be used as indices to access lookup tables for fast precomputation of the similarity measure, as it is described in the next subsection.

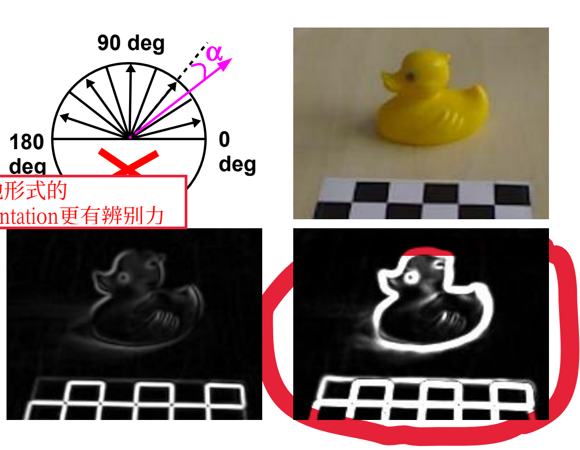


Fig. 3: **Upper Left**: Quantizing the gradient orientations: the pink orientation is closest to the second bin. **Upper right:** A toy duck with a calibration pattern. **Lower Left:** The gradient image computed on a gray value image. The object contour is hardly visible. **Lower right:** Gradients computed with our method. Details of the object contours are clearly visible.

$\mathcal{J}$ can be computed very efficiently: We first compute a map for each quantized orientation, whose values are set to 1 if the corresponding pixel location in the input image has this orientation, and 0 if it does not. $\mathcal{J}$ is then obtained by shifting these maps over the range of $\left[-\frac{T}{2}, +\frac{T}{2}\right] \times \left[-\frac{T}{2}, +\frac{T}{2}\right]$ and merging all shifted versions with an OR operation.

## 2.4 Precomputing Response Maps

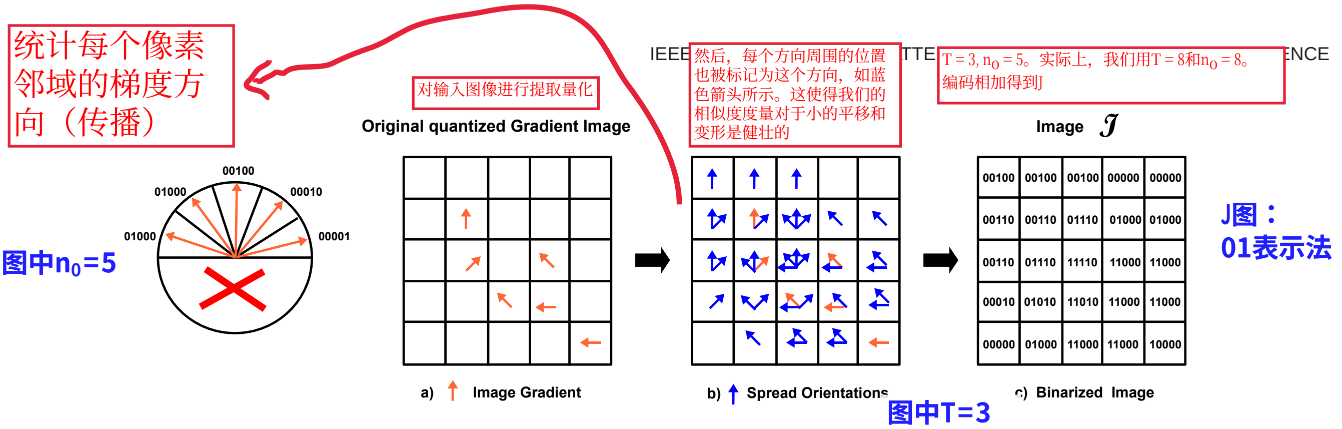
As shown in Fig. 5, $\mathcal{J}$ is used together with lookup tables to precompute the value of the max operation in Eq. (2) for each location and each possible orientation $\text{ori}(\mathcal{O}, r)$ in the template. We store the results into 2D maps $\mathcal{S}_i$. Then, to evaluate the similarity function, we will just have to sum values read from these $\mathcal{S}_i$s.

We use a lookup table $\tau_i$ for each of the $n_o$ quantized orientations, computed offline as:

$$\tau_i[\mathcal{L}] = \max_{l\in\mathcal{L}} |\cos(i-l)| , \qquad (5)$$

where

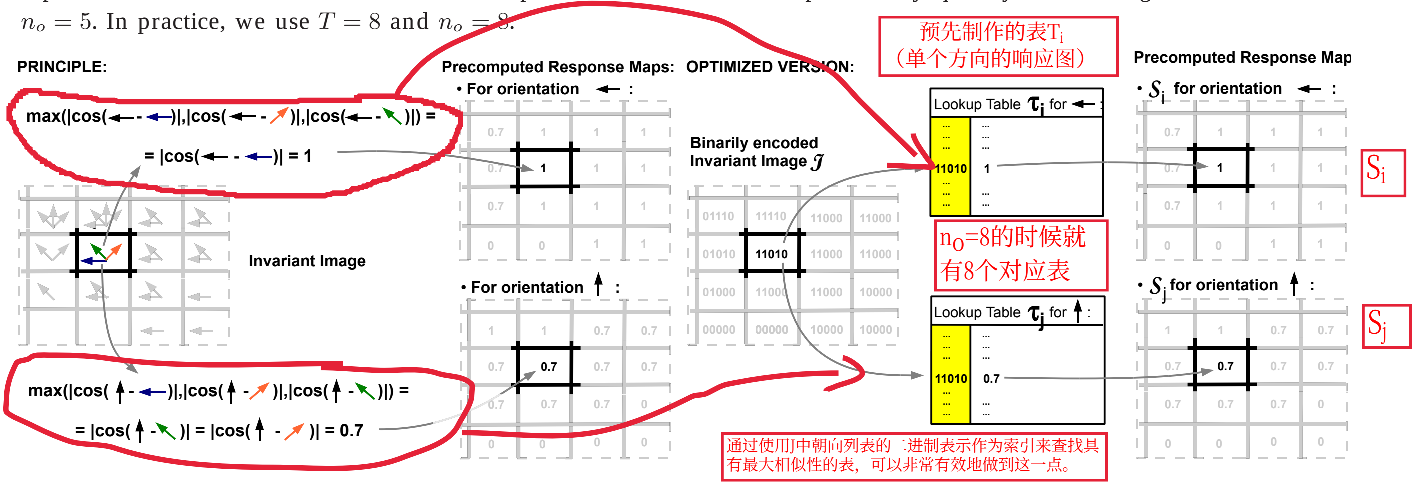
- $i$ is the index of the quantized orientations. To keep the notations simple, we also use $i$ to represent the corresponding angle in radians;
- $\mathcal{L}$ is a list of orientations appearing in a local neighborhood of a gradient with orientation $i$ as described in Section 2.3. In practice, we use the integer value corresponding to the binary representation of $\mathcal{L}$ as an index to the element in the lookup table.



然后，每个方向周围的位置也被标记为这个方向，如蓝色箭头所示。这使得我们的相似度度量对于小的平移和变形是健壮的

T = 3, $n_O$ = 5。实际上，我们用T = 8和$n_O$ = 8。编码相加得到J

Fig. 4: Spreading the gradient orientations. **Left:** The gradient orientations and their binary code. We do not consider the direction of the gradients. **a)** The gradient orientations in the input image, shown in orange, are first extracted and quantized. **b)** Then, the locations around each orientation are also labeled with this orientation, as shown by the blue arrows. This allows our similarity measure to be robust to small translations and deformations. **c)** $\mathcal{J}$ is an efficient representation of the orientations after this operation, and can be computed very quickly. For this figure, $T = 3$ and $n_o = 5$. In practice, we use $T = 8$ and $n_o = 8$.


PRINCIPLE:
max(|cos(← - ←)|,|cos(← - ↗)|,|cos(← - ↖)|) =
= |cos(← - ←)| = 1
Invariant Image
max(|cos(↑ - ←)|,|cos(↑ - ↗)|,|cos(↑ - ↖)|) =
= |cos(↑ - ↖)| = |cos(↑ - ↗)| = 0.7
Precomputed Response Maps:
• For orientation ← :
• For orientation ↑ :
OPTIMIZED VERSION:
Binarily encoded Invariant Image 𝒥
01110 11110 11000 11000
01010 11010 11000 11000
01000 11000 11000 10000
00000 00000 10000 10000
预先制作的表$T_i$（单个方向的响应图）
Lookup Table $\tau_i$ for ← :
11010 1
$n_O$=8的时候就有8个对应表
Lookup Table $\tau_j$ for ↑ :
11010 0.7
Precomputed Response Map
• $\mathcal{S}_i$ for orientation ← :
• $\mathcal{S}_j$ for orientation ↑ :
$S_i$
$S_j$

通过使用J中朝向列表的二进制表示作为索引来查找具有最大相似性的表，可以非常有效地做到这一点。

Fig. 5: Precomputing the Response Maps $\mathcal{S}_i$. **Left:** There is one response map for each quantized orientation. They store the maximal similarity between their corresponding orientation and the orientations $\text{ori}_j$ already stored in the "Invariant Image". **Right:** This can be done very efficiently by using the binary representation of the list of orientations in $\mathcal{J}$ as an index to lookup tables of the maximal similarities.

For each orientation $i$ we can now compute the value at each location $c$ of the response map $\mathcal{S}_i$ as:

$$\mathcal{S}_i(c) = \tau_i[\mathcal{J}(c)] \, . \tag{6}$$

Finally, the similarity measure of Eq. (2) can be evaluated as:

$$\mathcal{E}(\mathcal{I}, \mathcal{T}, c) = \sum_{r \in \mathcal{P}} \mathcal{S}_{\text{ori}(\mathcal{O}, r)}(c + r) \tag{7}$$

相识度公式变形为

Since the maps $\mathcal{S}_i$ are shared between the templates, matching several templates against the input image can be done very fast once the maps are computed.

## 2.5 Linearizing the Memory for Parallelization

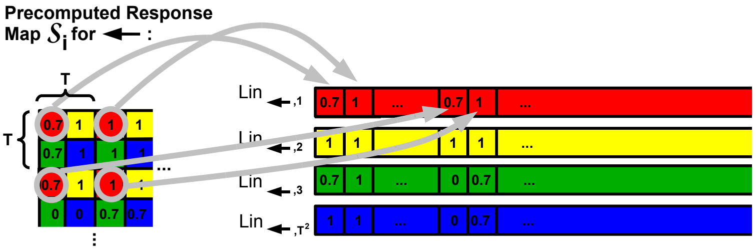
Thanks to Eq. (7), we can match a template against the whole input image by only adding the values in the response maps $\mathcal{S}_i$. However, one of the advantages of spreading the orientations as was done in Section 2.3 is that it is sufficient to do the evaluation only every $T^{\text{th}}$ pixel without reducing the recognition performance. If we want to exploit this property efficiently, we have to take into account the architecture of modern computers.

要有效利用缓存

Modern processors do not only read one data value at a time from the main memory but several ones simultaneously, called a *cache line*. Accessing the memory at random places results in a *cache miss* and slows down the computations. On the other hand, accessing several values from the same cache line is very cheap. As a consequence, storing data in the same order as they are read speeds up the computations significantly. In addition, this allows parallelization: For instance, if 8-bit values are used as it is the case for our $\mathcal{S}_i$ maps, SSE instructions can perform operations on 16 values in parallel. On multicore processors or on the GPU, even more operations can be performed simultaneously. For example, the NVIDIA Quadro GTX 590 can perform 1024 operations in parallel.

Therefore, as shown in Fig. 6, we store the precomputed response maps $\mathcal{S}_i$ into memory in a cache-friendly way: We restructure each response map so that the values of one row that are $T$ pixels apart on the $x$ axis are now stored next to each other in memory. We continue with the row which is $T$ pixels apart on the $y$ axis once we finished with the current one.

第i个map的存储方式



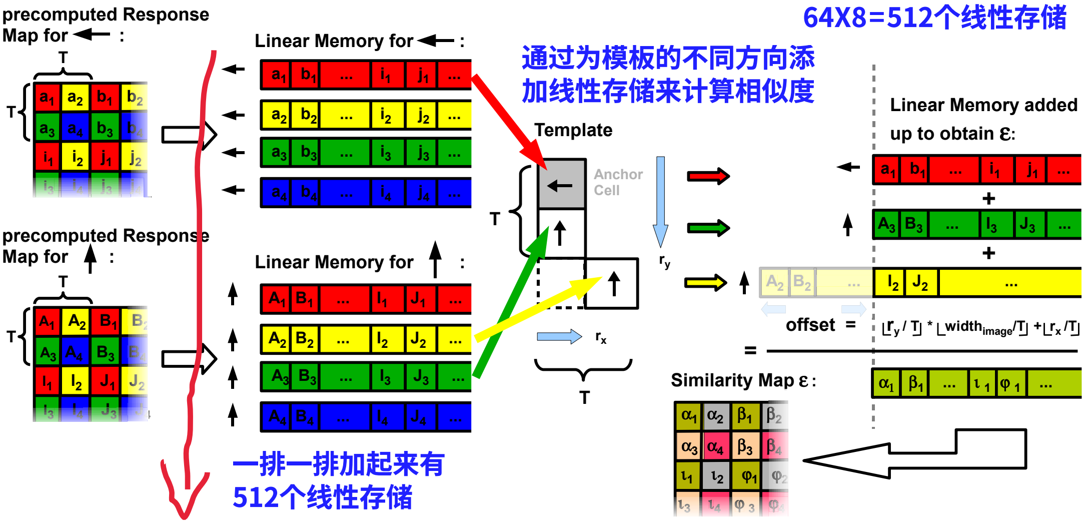
→ T²

T=2，线性存储数目为2^2;
T=8,每个响应图的线性存储数目为8^2=64

Fig. 6: Restructuring the way the response images $\mathcal{S}_i$ are stored in memory. The values of one image row that are $T$ pixels apart on the $x$ axis are stored next to each other in memory. Since we have $T^2$ such linear memories per response map, and $n_o$ quantized orientations, we end up with $T^2 \cdot n_o$ different linear memories.

算上8个分区，有
64X8=512个线性存储

输入图像



通过为模板的不同方向添加线性存储来计算相似度

一排一排加起来有
512个线性存储

Fig. 7: Using the linear memories. We can compute the similarity measure over the input image for a given template by adding up the linear memories for the different orientations of the template (the orientations are visualized by black arrows pointing in different directions), shifted by an offset depending on the relative locations $(r_x, r_y)^\top$ in the template with respect to the anchor point. Performing these additions with parallel SSE instructions further speeds up the computation. In the final similarity map $\mathcal{E}$ only each $T$th pixel has to be parsed to find the object.

Finally, as described in Fig. 7, computing the similarity measure for a given template at each sampled image location can be done by adding the linearized memories with an appropriate offset computed from the locations $r$ in the templates.

## 2.6 Extension to Dense Depth Sensors

In addition to color images, recent commodity hardware like the Kinect allows to capture dense depth maps in real-time. If these depth maps are aligned to the color images, we can make use of them to further increase the robustness of our approach as we have recently shown in [32].

Similar to the image cue, we decided to use quantized surface normals computed from a dense depth map in our template representation as shown in Fig. 8. They allow us to represent both close and far objects while fine structures are preserved.

In the following, we propose a method for the fast and robust estimation of surface normals in a dense range image. Around each pixel location $x$, we consider the first order Taylor expansion of the depth function $\mathcal{D}(x)$:

快速计算法线，对于每个像素点x,对深度函数D(x)的一阶泰勒展开

$$\mathcal{D}(x + dx) - \mathcal{D}(x) = dx^\top \nabla\mathcal{D} + h.o.t. \tag{8}$$

Within a patch defined around $x$, each pixel offset $dx$ yields an equation that constrains the value of $\nabla\mathcal{D}$, allowing to estimate an optimal gradient $\widehat{\nabla\mathcal{D}}$ in a least-square sense. This depth gradient corresponds to a 3D plane going through three points $X, X_1$ and $X_2$:

$$X = \vec{v}(x)\mathcal{D}(x), \tag{9}$$

$$X_1 = \vec{v}(x + [1, 0]^\top)(\mathcal{D}(x) + [1, 0]\widehat{\nabla\mathcal{D}}), \tag{10}$$

$$X_2 = \vec{v}(x + [0, 1]^\top)(\mathcal{D}(x) + [0, 1]\widehat{\nabla\mathcal{D}}). \tag{11}$$

where $\vec{v}(x)$ is the vector along the line of sight that goes through pixel $x$ and is computed from the internal parameters of the depth sensor. The normal to the surface at the 3D point that projects on $x$ can be estimated as the normalized cross-product of $X_1 - X$ and $X_2 - X$.

However this would not be robust around occluding contours, where the first order approximation of Eq. (8) no longer holds. Inspired by bilateral filtering, we ignore the contributions of pixels whose depth difference with the central pixel is above a threshold. In practice, this approach effectively smooths out quantization noise on the surface, while still pro-

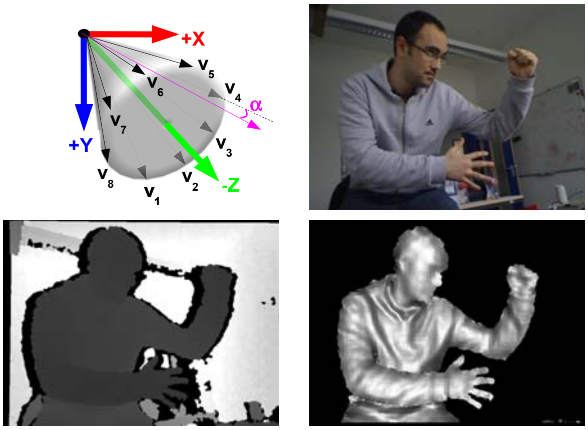


Fig. 8: **Upper Left**: Quantizing the surface normals: the pink surface normal is closest to the precomputed surface normal $v_4$. It is therefore put into the same bin as $v_4$. **Upper right**: A person standing in an office room. **Lower Left**: The corresponding depth image. **Lower right**: Surface normals computed with our approach. Details are clearly visible and depth discontinuities are well handled. We removed the [illegible]bility reasons.

联合相似度测度就是图像梯度
测度和曲面法线测度的总和

viding meaningful surface normal estimates around strong depth discontinuities. Our similarity measure is then defined as the dot product of the normalized surface normals, instead of the cosine difference for the image gradients in Eq.(2). We otherwise apply the same technique we apply to the image gradients. The combined similarity measure is simply the sum of the measure for the image gradients and the one for the surface normals.

To make use of our framework we have to quantize the 3D surface normals into $n_0$ bins. This is done by measuring the angles between the computed normals and a set of $n_0$ precomputed vectors. These vectors are arranged in a circular cone shape originating from the peak of the cone pointing towards the camera. To make the quantization robust to noise, we assign to each location the quantized value that occurs most often in a $3 \times 3$ neighborhood. The whole process is very efficient and needs only 14ms on the CPU and less than 1ms on the GPU.

### 2.7 Computation Time Study

In this section we compare the numbers of operations required by the original method from [18] and the method we propose.

The time required by $\mathcal{E}_{\text{Steger}}$ from [18] to evaluate $R$ templates over an $M \times N$ image is $M \cdot N \cdot R \cdot G \cdot (S+A)$, where $G$ the average number of gradients in a template, $S$ the time to evaluate the similarity function between two gradient orientations and, $A$ the time to add two values.

Changing $\mathcal{E}_{\text{Steger}}$ to Eq. (2) and making use of $\mathcal{J}$ leads to a computation time of $M \cdot N \cdot T^2 \cdot O + \frac{M \cdot N}{T^2} \cdot R \cdot G \cdot (L+A)$, where $L$ is the time needed for accessing once the lookup tables $\tau_i$ and $O$ is the time to OR two values together. The first term corresponds to the time needed to compute $\mathcal{J}$, the second one to the time needed to actually compute Eq. (2).

Precomputing the response maps $\mathcal{S}_i$ further changes the complexity of our approach to $M \cdot N \cdot (T^2 \cdot O + n_o \cdot L) + \frac{M \cdot N}{T^2} \cdot R \cdot G \cdot A$.

Linearizing our memory allows the additional use of parallel SSE instructions. In order to run 16 operations in parallel, we approximate the response values in the lookup tables using bytes. The final complexity of our algorithm is then $M \cdot N \cdot (T^2 \cdot O + (n_o+1) \cdot L) + \frac{M \cdot N}{16T^2} \cdot R \cdot G \cdot A$.

In practice we use $T = 8$, $M = 480$, $N = 640$, $R > 1000$, $G \approx 100$ and $n_o = 8$. If we assume for simplicity that $L \approx A \approx O \approx 1$ time unit, this leads to a speed improvement compared to the original energy formulation $\mathcal{E}_{\text{Steger}}$ of a factor $T^2 \cdot 16(1+S)$ if we assume that the number of templates $R$ is large. Note that we did not incorporate the cache friendliness of our approach since it is very hard to model. Still, since [18] evaluates the similarity of two orientations with the normalized dot product of the two corresponding gradients, $S$ can be set to 3 and we obtain a theoretical gain in speed of at least a factor of 4096.

### 2.8 Experimental Validation

LINE-2D
LINE-3D
LINE-MOD

We compared our approach, which we call LINE (for LINEearizing the memory), to DOT [7], HOG [1], TLD [20] and Steger method [18]. For these experiments we used three different variations of LINE: LINE-2D that uses the image gradients only, LINE-3D that uses the surface normals only and LINE-MOD, for *multimodal*, which uses both.

DOT is a representative for fast template matching while HOG and Steger stand for slow but very robust template matching. In contrast to them, TLD represents very recent insights in online learning.

Instead of gray value gradients, we used the color gradient method of Section 2.2 for DOT, HOG and Steger, which resulted in an enhancement of recognition. Moreover, we used the author's implementations for DOT and TLD. For the approach of Steger we used our own implementation with 4 pyramid levels. For HOG, we also used our own optimized implementation and replaced the Support Vector Machine mentioned in the original work of HOG by a nearest neighbor search. In this way, we can use it as a robust representation and quickly learn new templates as with the other methods.

The experiments were performed on one processor of a standard notebook with an Intel Centrino Processor Core2Duo with 2.4 GHz and 3 GB of RAM. For obtaining the image and the depth data we used the Primesense$^{(tm)}$ PSDK 5.0 device.

### 2.9 Robustness

We used six sequences made of over 2000 real images each. Each sequence presents illumination and large

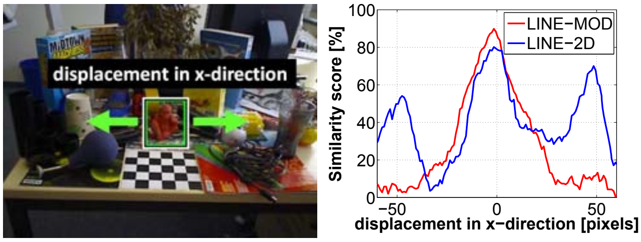


Fig. 9: Combining many modalities results in a more discriminative response function. Here we compare LINE-MOD against LINE-2D on the shown image. We plot the response function of both methods with respect to the true location of the monkey. One can see that the response of LINE-MOD exhibits a single and discriminative peak whereas LINE-2D has several peaks which are of comparable height. This is one explanation why LINE-MOD works better and produces fewer false positives.

viewpoint changes over heavily cluttered background. Ground truth is obtained with a calibration pattern attached to each scene that enables us to know the actual location of the object. The templates were learned over homogeneous background.

We consider the object to be correctly detected if the location given back is within a fixed radius of the ground truth position.

As we can see in the left columns of Fig. 11 and Fig. 12, LINE-2D mostly outperforms all other image-based approaches. The only exception is the method of Steger which gives similar results. This is because our approach and the one of Steger use similar score functions. However the advantage of our method in terms of computation times is very clear from Fig. 10.

The reason for the weak detection results of TLD is that while this method works well under smooth background transition, it is not suitable to detect known objects over unknown backgrounds.

If a dense depth sensor is available we can further increase the robustness without becoming slower at runtime. This is depicted in the left columns of Fig. 11 and Fig. 12 where LINE-MOD always outperforms all the other approaches and shows only few false positives. We believe that this is due to the complementarity of the object features that compensate for the weaknesses of each other. The depth cue alone often performs not very well.

The superiority of LINE-MOD becomes more obvious in Table 1: If we set the threshold for each approach to allow for 97% true positive rate and only evaluate the hypothesis with the largest response, we obtain for LINE-MOD a high detection rate with a very small false positive rate. This is in contrast to LINE-2D, where the true positive rate is often over 90%, but the false positive rate is not negligible. The true positive rate is computed as the ratio of correct detections and the number of images; similarly the false positive rate is the ratio of the number of incorrect detections and the number of images.

One reason for this high robustness is the good separability of the multimodal approach as shown in the middle of Figs. 11 and 12: In contrast to LINE-2D where we have a significant overlap between true and false positives, LINE-MOD separates at a specific threshold — about 80 in our implementation — almost all true positives well from almost all false positives. This has several advantages. First, we will detect almost all instances of the object by setting the threshold to this specific value. Second, we also know that almost every returned template with a similarity score above this specific value is a true positive. Third, the threshold is always around the same value which supports the conclusion that it might also work well for other objects.

### 2.10 Speed

Learning new templates only requires extracting and storing the image features (and if used the depth features), which is almost instantaneous. Therefore, we concentrate on runtime performance.

The runtimes given in Fig. 10 show that the general LINE approach is real-time and can parse a VGA image with over 3000 templates with about 10 fps on the CPU. The small difference of computation times between LINE-MOD and LINE-2D and LINE-3D comes from the slightly slower preprocessing step of LINE-MOD that includes the two preprocessing steps of LINE-2D and LINE-3D.

DOT is initially faster than LINE but becomes slower as the number of templates increases. This is because the runtime of LINE is independent of the template size whereas the runtime of DOT is not. Therefore, to handle larger objects DOT has to use larger templates which makes the approach slower once the number of templates increases.

Our implementation of Steger *et al.* is approximately 100 times slower than our LINE-MOD method. Note that we use 4 pyramid levels for more efficiency which is one of the reasons for the different speed improvement given in Section 2.7, where we assumed no image pyramid.

TLD uses a tree classifier similar to [33] which is the reason why the timings stay relatively equal with respect to the number of templates. Since this paper is concerned with detection, for this experiment we consider only the detection component of TLD, and not the tracking component.

### 2.11 Occlusion

We also tested the robustness of LINE-2D and LINE-MOD with respect to occlusion. We added synthetic noise and illumination changes to the images, incrementally occluded the six different objects of Section 2.9 and measured the corresponding response values. As expected, the similarity measure used by LINE-2D and LINE-MOD behave linearly in the percentage of occlusion as reported in Fig. 10. This is a

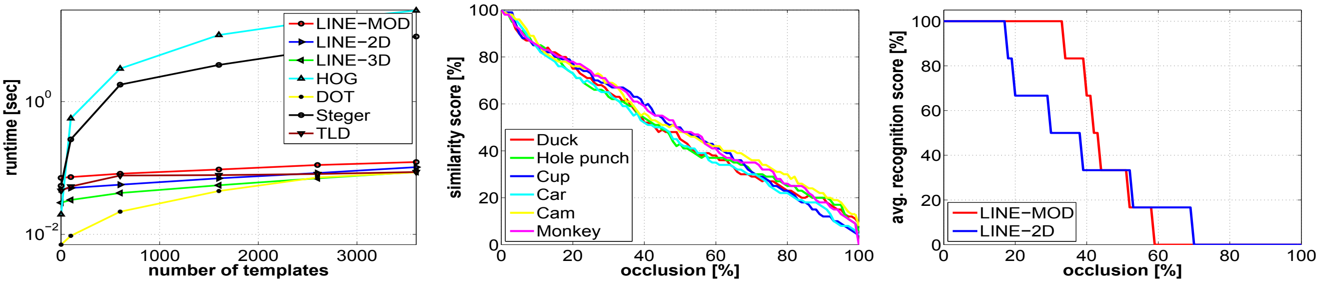


Fig. 10: **Left:** Our new approach runs in real-time and can parse a 640×480 image with over 3000 templates at about 10 fps. **Middle:** Our new approach is linear with respect to occlusion. **Right:** Average recognition score for the six objects of Section2.9 with respect to occlusion.

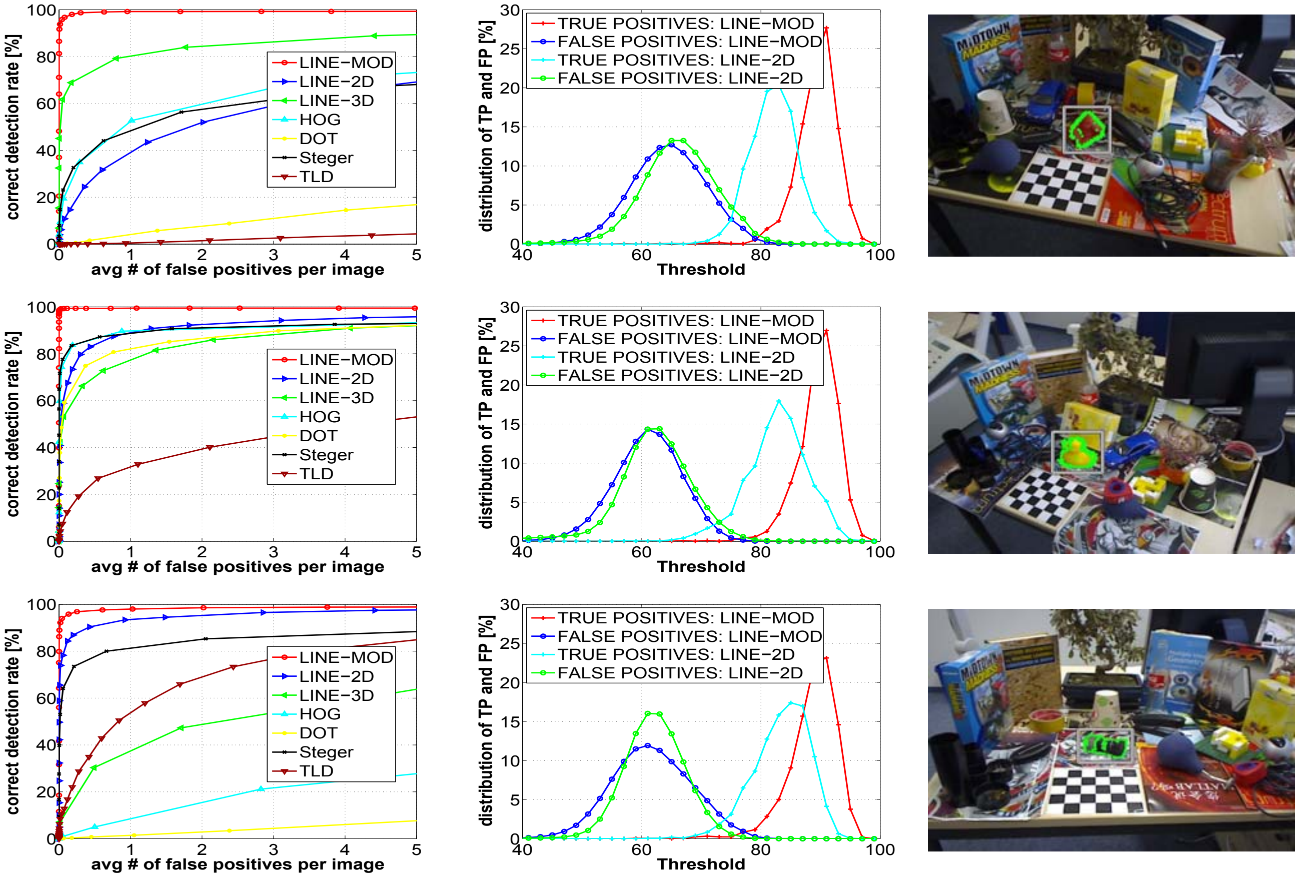


Fig. 11: Comparison of LINE-2D, which is based on gradients, LINE-3D, which is based on normals and LINE-MOD, which uses both cues, to DOT [7], HOG [1], Steger [18] and TLD [20] on real 3D objects. Each row corresponds to a different sequence (made of over 2000 images each) on heavily cluttered background: A monkey, a duck and a camera. The approaches were trained on a homogeneous background. **Left:** Percentage of true positives plotted against the average percentage of false positives. LINE-2D outperforms all other image-based approaches when considering the combination of robustness and speed. If a dense depth sensor is available, we can extend LINE to 3D surface normals resulting in LINE-3D and LINE-MOD. LINE-MOD provides about the same recognition rates for all objects while the other approaches have a much larger variance depending on the object type. LINE-MOD outperforms the other approaches in most cases. **Middle:** The distribution of true and false positives plotted against the threshold. In case of LINE-MOD they are better separable from each other than in the case of LINE-2D. **Right:** One sample image of the corresponding sequence shown with the object detected by LINE-MOD.

desirable property since it allows detection of partly occluded templates by setting the detection threshold with respect to the tolerated percentage of occlusion.

We also experimented with real scenes where we first learned our six objects in front of a homogeneous background and then added heavy 2D and 3D background clutter. For recognition we incrementally occluded the objects. We define our object as correctly recognized if the template with the highest response is found within a fixed radius of the ground truth object

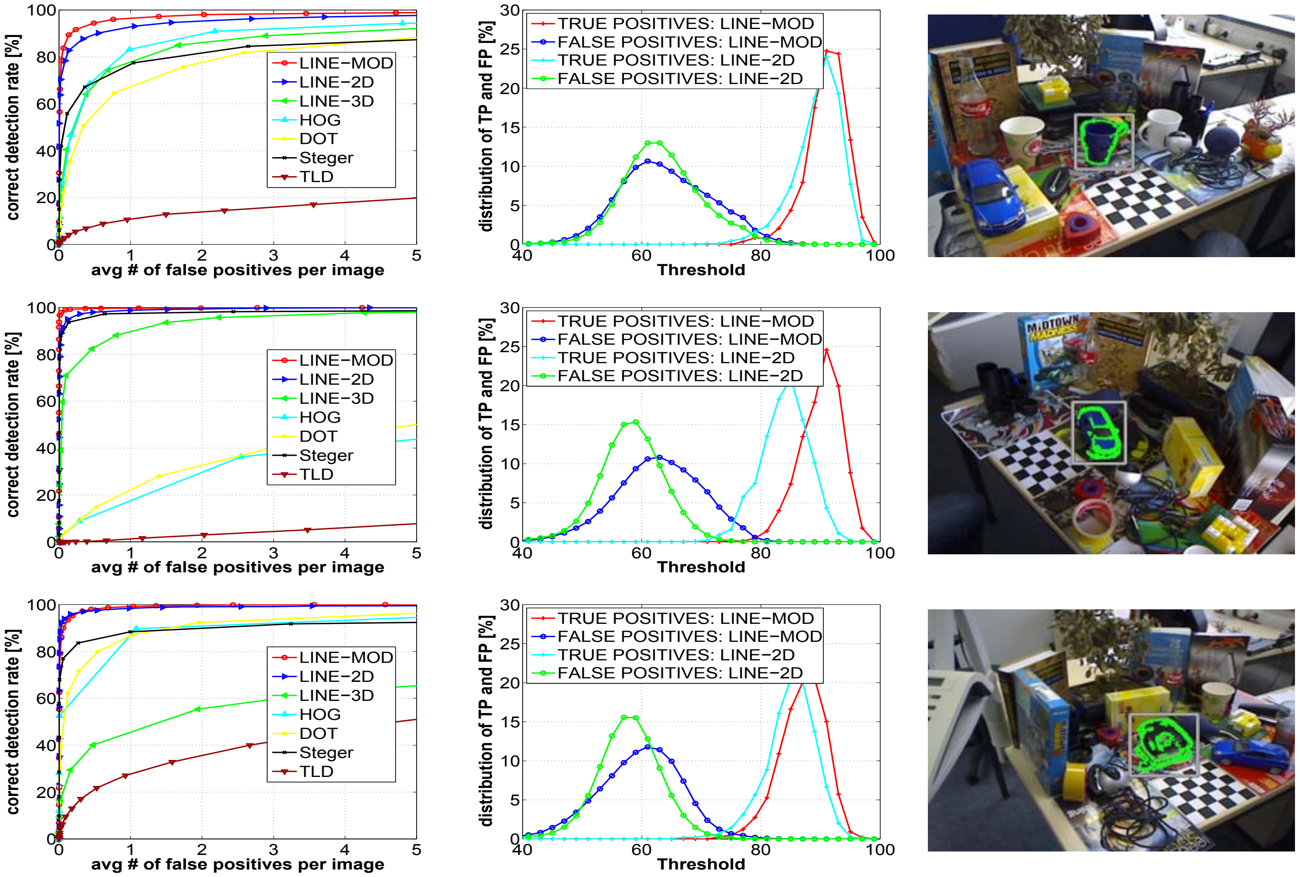
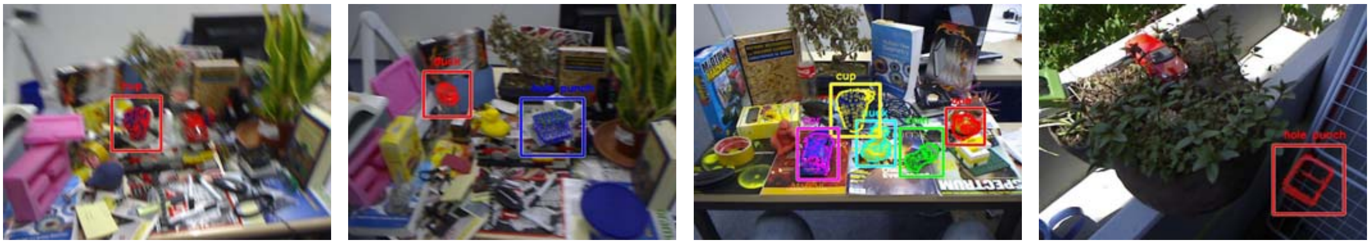


Fig. 12: Same experiments as shown in Fig. 11, for different objects: A cup, a toy car and a hole punch. These values were evaluated on 2000 images for each object.

| Sequence | LINE-MOD | LINE-3D | LINE-2D | HOG | DOT | Steger | TLD |
|---|---|---|---|---|---|---|---|
| Toy-Monkey (2164 pics) | 97.9%–0.3% | 86.1%–13.8% | 50.8%–49.1% | 51.8%–48.2% | 8.6%–91.4% | 69.6%–30.3% | 0.8%–99.1% |
| Camera (2173 pics) | 97.5%–0.3% | 61.9%–38.1% | 92.8%–6.7% | 18.2%–81.8% | 1.9%–98.0% | 96.9%–0.4% | 53.3%–46.6% |
| Toy-Car (2162 pics) | 97.7%–0.0% | 95.6%–2.5% | 96.9%–0.4% | 44.1%–55.9% | 34.0%–66.0% | 83.6%–16.3% | 0.1%–98.9% |
| Cup (2193 pics) | 96.8%–0.5% | 88.3%–10.6% | 92.8%–6.0% | 81.1%–18.8% | 64.1%–35.8% | 90.2%–8.9% | 10.4%–89.6% |
| Toy-Duck (2223 pics) | 97.9%–0.0% | 89.0%–10.0% | 91.7%–8.0% | 87.6%–12.4% | 78.2%–21.8% | 92.2%–7.6% | 28.0%–71.9% |
| Hole punch (2184 pics) | 97.0%–0.2% | 70.0%–30.0% | 96.4%–0.9% | 92.6%–7.4% | 87.7%–12.0% | 90.3%–9.7% | 26.5%–73.4% |

TABLE 1: True and false positive rates for different thresholds on the similarity measure of different methods. In some cases no hypotheses were given back so the sum of true and false positives can be lower than 100%. LINE-2D outperforms all other image-based approaches when taking into account the combination of performance rate and speed. If a dense depth sensor is available, our LINE-MOD approach obtains very high recognition rates at the cost of almost no false positives, and outperforms all the other approaches.



(a) (b) (c) (d)

Fig. 13: Typical failure cases. Motion blur can produce **(a)** false negative: the red car is not detected and **(b)** false positive: the duck is detected on the background. Similar structures can also produce false positives: **(c)** The monkey statue is detected on a bowl, and **(d)** the templates for the hole punch seen under some viewpoints are not discriminative and one is here detected on a structure with orthogonal lines.

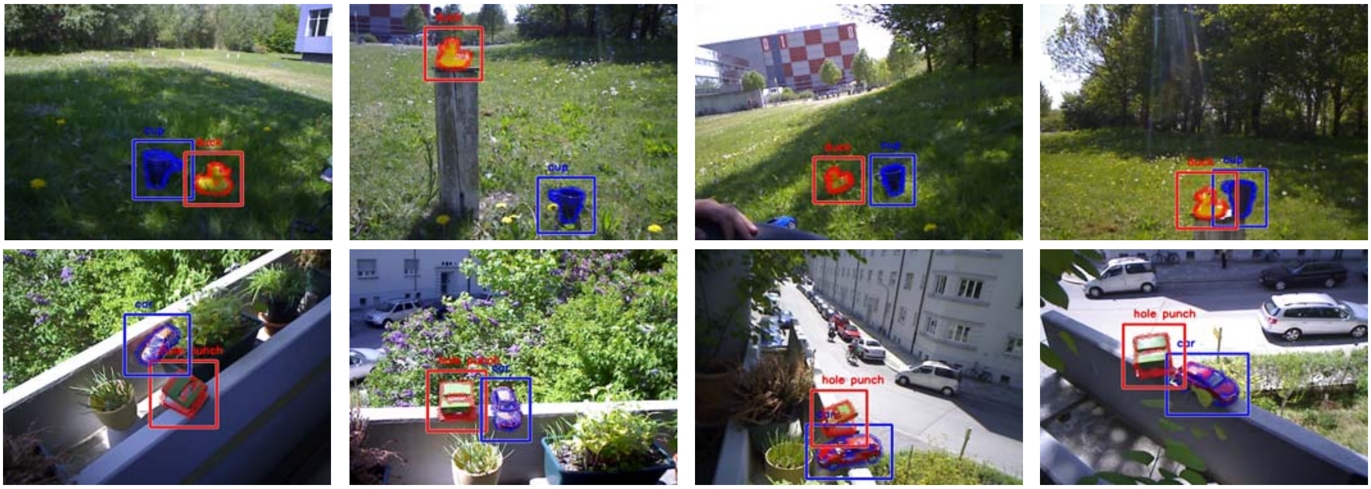


Fig. 14: Different texture-less 3D objects are detected with LINE-2D in real-time under different poses in difficult outdoor scenes with partial occlusion, illumination changes and strong background clutter.

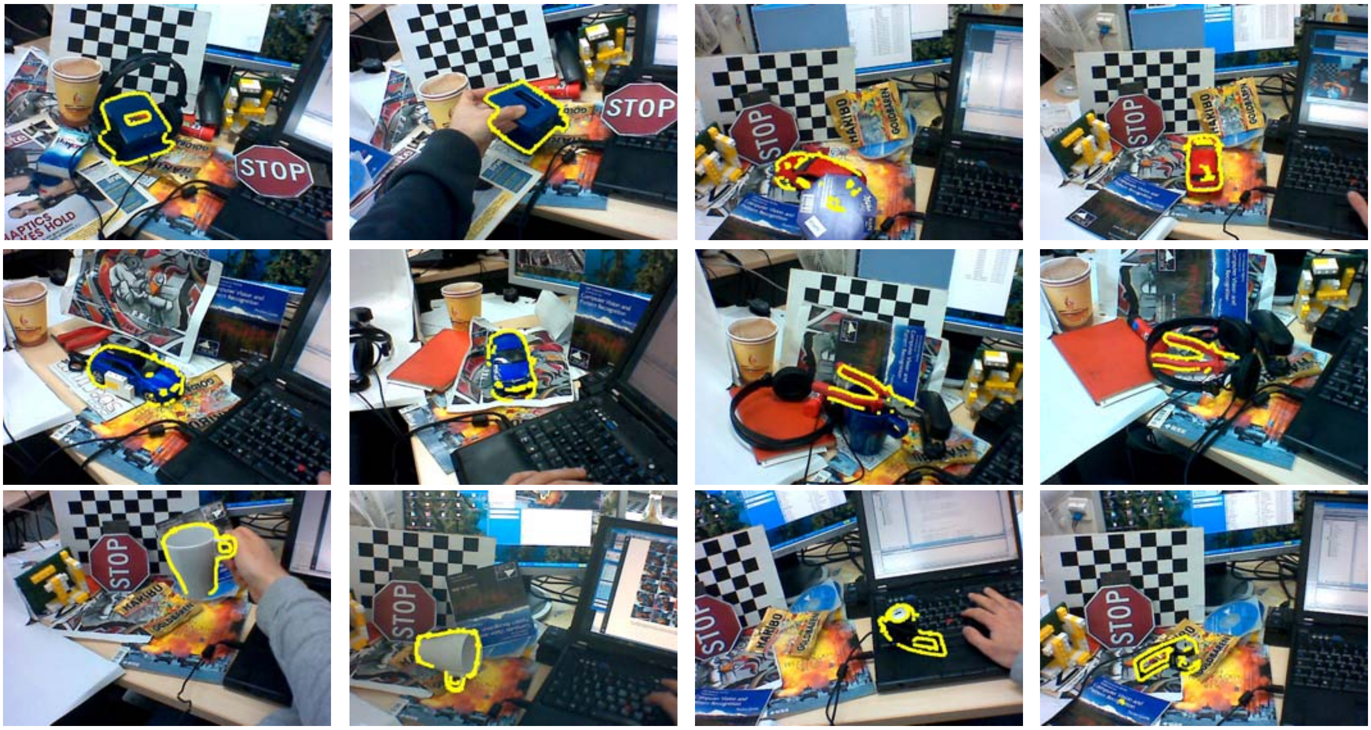


Fig. 15: Different texture-less 3D objects are detected with LINE-2D in real-time under different poses on heavily cluttered background with partial occlusion.

location. The average recognition result is displayed in Fig. 10: With 20% occlusion for LINE-2D and with over 30% occlusion for LINE-MOD we are still able to recognize objects.

## 2.12 Number of Templates

We discuss here the average number of templates needed to detect an arbitrary object from a large number of viewpoints. In our implementation, approximately 2000 templates are needed to detect an object with 360 degree tilt rotation, 90 degree inclination rotation and in-plane rotations of ± 80 degree—tilt and inclination cover the half-sphere of Fig. 17. With the number of templates given here, the detection works for scale changes in the range of [1.0; 2.0].

尺度变化为1~2

## 2.13 Examples

Figs. 14, 15 and 16 show the output of our methods on texture-less objects in different heavy cluttered inside and outside scenes. The objects are detected under partial occlusion, drastic pose and illumination changes. In Figs 14 and 15, we only use gradient features, whereas in Fig. 16, we also use 3D normal features. Note that we could not apply LINE-MOD

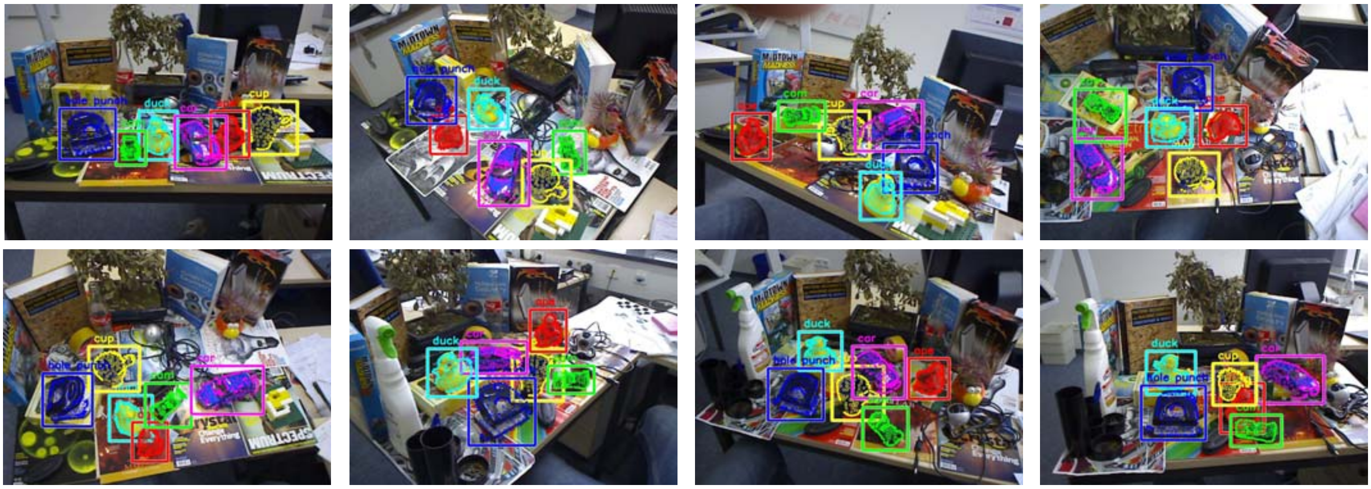


Fig. 16: Different texture-less 3D objects detected simultaneously in real-time by our LINE-MOD method under different poses on heavily cluttered background with partial occlusion.

2000张模
板制作

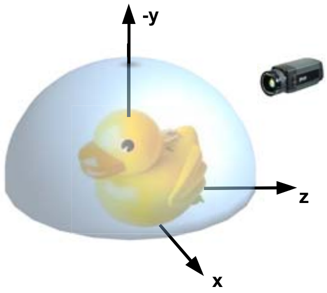


Fig. 17: An arbitrary object can be detected using approximately 2000 templates. The half-sphere represents the detection range in terms of tilt and inclination rotations. Additionally, in-plane rotations of ± 80 degree and scale changes in the range from $[1.0, 2.0]$ can be handled.

outside since the Primesense device was not able to produce a depth map under strong sunlight.

### 2.14 Failure Cases

Fig. 13 shows the limitations of our method. It tends to produce false positives and false negatives in case of motion blur. False positives and false negatives can also be produced when some templates are not discriminative enough.

## 3 Conclusion

We presented a new method that is able to detect 3D texture-less objects in real-time under heavily background clutter, illumination changes and noise. We also showed that if a dense depth sensor is available, 3D surface normals can be robustly and efficiently computed and used together with 2D gradients to further increase the recognition performance. We demonstrated how to take advantage of the architecture of modern computers to build a fast but very discriminant representation of the input images that can be used to consider thousands of arbitrarily sized and arbitrarily shaped templates in real-time. Additionally, we have shown that our approach outperforms state-of-the-art methods with respect to the combination of recognition rate and speed especially in heavily cluttered environments.

## Acknowledgments

The authors thank Stefan Holzer and Kurt Konolige for the useful discussions and their valuable suggestions. This project was funded by the BMBF project AVILUSplus (01IM08002). P. Sturm is grateful to the Alexander-von-Humboldt Foundation for a Research Fellowship supporting his sabbatical at TU München.

## References

[1] N. Dalal and B. Triggs, “Histograms of Oriented Gradients for Human Detection,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[2] R. Fergus, P. Perona, and A. Zisserman, “Weakly Supervised Scale-Invariant Learning of Models for Visual Recognition,” *International Journal of Computer Vision*, 2006.

[3] A. Bosch, A. Zisserman, and X. Munoz, “Image Classification Using Random Forests,” in *IEEE International Conference on Computer Vision*, 2007.

[4] V. Ferrari, F. Jurie, and C. Schmid, “From Images to Shape Models for Object Detection,” *International Journal of Computer Vision*, 2009.

[5] P. Viola and M. Jones, “Fast Multi-View Face Detection,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[6] D. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *International Journal of Computer Vision*, vol. 20, no. 2, pp. 91–110, 2004.

[7] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab, “Dominant Orientation Templates for Real-Time Detection of Texture-Less Objects,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[8] M. Muja, R. Rusu, G. Bradski, and D. Lowe, “Rein - a Fast, Robust, Scalable Recognition Infrastructure,” in *International Conference on Robotics and Automation*, 2011.

[9] G. Borgefors, “Hierarchical Chamfer Matching: a Parametric Edge Matching Algorithm,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1988.

[10] D. Huttenlocher, G. Klanderman, and W. Rucklidge, “Comparing Images Using the Hausdorff Distance,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1993.

[11] D. Gavrila and V. Philomin, "Real-Time Object Detection for "smart" Vehicles," in *IEEE International Conference on Computer Vision*, 1999.

[12] C. F. Olson and D. P. Huttenlocher, "Automatic Target Recognition by Matching Oriented Edge Pixels," *IEEE Transactions on Image Processing*, vol. 6, 1997.

[13] G. Blake, R. Dreslinski, and T. Mudge, "A Survey of Multicore Processors," *IEEE Signal Processing Magazine*, vol. 26, November 2009.

[14] S. Hinterstoisser, V. Lepetit, S. Benhimane, P. Fua, and N. Navab, "Learning Real-Time Perspective Patch Rectification," *International Journal of Computer Vision*, 2011.

[15] W. Rucklidge, "Efficiently Locating Objects Using the Hausdorff Distance," *International Journal of Computer Vision*, 1997.

[16] S. Holzer, S. Hinterstoisser, S. Ilic, and N. Navab, "Distance Transform Templates for Object Detection and Pose Estimation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[17] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, 1986.

[18] C. Steger, "Occlusion Clutter, and Illumination Invariant Object Recognition," in *International Archives of Photogrammetry and Remote Sensing*, 2002.

[19] Y. Amit, D. Geman, and X. Fan, "A Coarse-To-Fine Strategy for Multi-Class Shape Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004.

[20] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N Learning: Bootstrapping Binary Classifiers by Structural Constraints," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[21] C. Huang, H. Ai, Y. Li, and S. Lao, "Vector Boosting for Rotation Invariant Multi-View Face Detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[22] M. Ozuysal, V. Lepetit, and P. Fua, "Pose Estimation for Category Specific Multiview Object Localization," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2009.

[23] M. Stark, M. Goesele, and B. Schiele, "Back to the Future: Learning Shape Models from 3D Cad Data," in *British Machine Vision Conference*, 2010.

[24] J. Liebelt and C. Schmid, "Multi-View Object Class Detection With a 3D Geometric Model," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[25] H. Su, M. Sun, L. Fei-Fei, and S. Savarese, "Learning a Dense Multi-View Representation for Detection, Viewpoint Classification and Synthesis of Object Categories," in *IEEE International Conference on Computer Vision*, 2009.

[26] M. Enzweiler, A. Eigenstetter, B. Schiele, and D. M. Gavrila, "Multi-Cue Pedestrian Classification With Partial Occlusion Handling," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[27] A. Ess, B. Leibe, and L. J. V. Gool, "Depth and Appearance for Mobile Scene Analysis," in *IEEE International Conference on Computer Vision*, 2007.

[28] D. M. Gavrila and S. Munder, "Multi-Cue Pedestrian Detection and Tracking from a Moving Vehicle," *International Journal of Computer Vision*, 2007.

[29] C. Wojek, S. Walk, and B. Schiele, "Multi-Cue Onboard Pedestrian Detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[30] M. Sun, G. R. Bradski, B.-X. Xu, and S. Savarese, "Depth-Encoded Hough Voting for Joint Object Detection and Shape Recovery," in *European Conference on Computer Vision*, 2010.

[31] E. Tola, V. Lepetit, and P. Fua, "Daisy: an Efficient Dense Descriptor Applied to Wide Baseline Stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.

[32] S. Hinterstoisser, C. Cagniart, S. Holzer, S. Ilic, K. Konolige, N. Navab, and V. Lepetit, "Multimodal Templates for Real-Time Detection of Texture-Less Objects in Heavily Cluttered Scenes," in *IEEE International Conference on Computer Vision*, 2011, submitted.

[33] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua, "Fast Keypoint Online Learning and Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.

**Stefan Hinterstoißer** is a PhD student at the CAMP at the Technische Universität München in Germany where he is part of the Computer Vision group. His current research interestes include real-time object detection and pose estimation of generic 2D/3D objects. He is especially interested in improving the recognition and detection of almost texture-less objects as they are often found in human environments.

**Cedric Cagniart** Cedric Cagniart is currently working towards his PhD degree at the Technische Universität München in Germany and the INRIA Rhône-Alpes in France. His research activities include markerless motion tracking and spatiotemporal modeling of deformable surfaces from visual data for computer animation and computer vision.

**Slobodan Ilic** is leading the Computer Vision Group of CAMP at TUM since 2009. From June 2006 he was a senior researcher at Deutsche Telekom Laboratories in Berlin. Before that he was a postdoctoral fellow at CVLab, EPFL, Switzerland, where he received his PhD in 2005. His research interests include: deformable surface modeling and tracking, 3D reconstruction, real-time object detection and tracking. He was recently an Area Chair for ICCV 2011 and is regularly a part of the Program Committee for all major computer vision conferences. Besides active academic involvement Slobodan has strong relations to industry and supervises a number of PhD students supported by industry.

**Peter Sturm** obtained MSc degrees from INPG (National Polytechnical Institute of Grenoble, France) and the University of Karlsruhe, both in 1994, and a PhD degree from INPG in 1997. His PhD thesis was awarded the SPECIF award (given to one French PhD thesis in Computer Science per year). After a two-year post-doc at Reading University, he joined INRIA on a permanent research position as Charg de Recherche in 1999. Since 2006, he is Directeur de Recherche (the INRIA equivalent of Professor). In 2009/10 he spent a one-year sabbatical at CAMP, TU Munich.

**Nassir Navab** is a full professor and director of the institute for Computer Aided Medical Procedures (CAMP) at Technical University of Munich (TUM) with a secondary faculty appointment at its Medical School. He is also acting as Chief Scientific Officer for SurgicEye (http://www.surgiceye.com). In November 2006, he was elected as a member of board of directors of MICCAI society. He has served on the Steering Committee of the IEEE Symposium on Mixed and Augmented Reality between 2001 and 2008. He is the author of hundreds of peer reviewed scientific papers and over 40 US and international patents. He has received Siemens Inventor of the Year award in 2001, and SMIT technology award in 2010 and co-authored many awarded papers in prestigious conferences.

**Pascal Fua** received an engineering degree from Ecole Polytechnique, Paris, in 1984 and the Ph.D. degree in Computer Science from the University of Orsay in 1989. He joined EPFL (Swiss Federal Institute of Technology) in 1996 where he is now a Professor in the School of Computer and Communication Science. His research interests include shape modeling and motion recovery from images, analysis of microscopy images, and Augmented Reality. He has (co)authored over 150 publications in refereed journals and conferences. He has been an associate editor of IEEE journal Transactions for Pattern Analysis and Machine Intelligence and has often been a program committee member, area chair, and program chair of major vision conferences.

**Vincent Lepetit** received the PhD degree in Computer Vision in 2001 from the University of Nancy, France. He then joined the Virtual Reality Lab at EPFL (Swiss Federal Institute of Technology) as a post-doctoral fellow and became a founding member of the Computer Vision Laboratory. His research interests are feature point matching, 3D camera tracking, and object recognition.