Object Disambiguation for Augmented Reality Applications

Wei-Chen Chiu[1]
walon@mpi-inf.mpg.de
Gregory S. Johnson[2]
gregory.s.johnson@intel.com
Daniel Mcculley[2]
daniel.b.mcculley@intel.com
Oliver Grau[2]
oliver.grau@intel.com
Mario Fritz[1]
mfritz@mpi-inf.mpg.de

[1] Max Planck Institute for Informatics
Saarbrücken, Germany
[2] Intel Corporation

Abstract

The broad deployment of wearable camera technology in the foreseeable future offers new opportunities for augmented reality applications ranging from consumer (e.g. games) to professional (e.g. assistance). In order to span this wide scope of use cases, a markerless object detection and disambiguation technology is needed that is robust and can be easily adapted to new scenarios. Further, standardized benchmarking data and performance metrics are needed to establish the relative success rates of different detection and disambiguation methods designed for augmented reality applications.

Here, we propose a novel object recognition system that fuses state-of-the-art 2D detection with 3D context. We focus on assisting a maintenance worker by providing an augmented reality overlay that identifies and disambiguates potentially repetitive machine parts. In addition, we provide an annotated dataset that can be used to quantify the success rate of a variety of 2D and 3D systems for object detection and disambiguation. Finally, we evaluate several performance metrics for object disambiguation relative to the baseline success rate of a human.

1 Introduction

The advent of affordable, highly miniaturized wearable camera technology in combination with the latest improvement of head-up display has intensified interest in augmented reality applications. The availability of such devices in the foreseeable future as well as the large scope of use cases in the consumer market (e.g. games) as well as industrial applications (e.g. maintenance) begs the question if current computer vision techniques can shoulder the expectations.

We investigate this question on a task of assisting a maintenance worker in a factory setting. The system has to provide an overlay to the worker so that machines parts are correctly identified. Depending on the application this simply supports successful completion of a task, averts dangers from the worker or prevents damage to the machine. In our study we will focus on sensing by a monocular camera as this is still the most commonly deployed modality in those devices to date.

Object recognition and detection has significantly matured over the last decade. We have seen great progress in instance [11] as well as category recognition [7, 12]. However in many of the aforementioned tasks we are faced with compositionality of objects from potentially repetitive parts. Robust matching of parts with their relational structure is required to detect the object as a whole and give semantics to the individual parts. We denote the task of predicting the identities of parts as object disambiguation.

Although such an object disambiguation task is really at the core of many augmented reality systems and systems for assistance in work environments in particular, there has been little progress in quantifying performance in these settings. In general, computer vision research has a strong tradition in building benchmarks that allow for measuring and comparing performance of object recognition and detection approaches. Most prominently the PASCAL challenge has greatly supported progress in object detection and the ImageNet challenge has played a similar role for object recognition. Therefore we advocate the need of a benchmark for augmented reality settings. We realize that it is very challenging to build such a benchmark in a completely task agnostic manner. In our study, we are focusing on a maintenance work task.

In order to establish a well defined benchmark, a performance metric is needed that allows for automatic evaluation. While there are widely adopted metrics for object recognition and detection, those are not directly applicable to our settings. First, object disambiguation has to deal with potentially repetitive objects whose identities are only resolved in context and therefore it is not captured by previous object detection metrics. Second, we are interest in the actual success of the user of the augmented reality system. Hence we seek a metric that measures the user's success in disambiguating the objects given the observation of the system's output.

We propose the first benchmark for augmented reality systems in maintenance work. Different metrics are evaluated to judge the systems performance in the context of the application. We propose a metric that closely follows the actual performance achieved by the human observer of the system's output. We propose the first system for object disambiguation that leverage 3d context from a SLAM system as well as flexible constraints on the matching procedure in order to robustly interpret the output of state-of-the-art object detectors for the task of object disambiguation.

2 Related Work

2D detection Our approach uses object detectors in order to evidence of machine parts from the image. We evaluate a range of commonly used object detectors[4, 7, 11, 16, 19]. All of them meet real-time constraints. While some of the were already built with efficiency in mind [4, 11, 19], other have seen recent extension by algorithms speed up as well as GPU computation [5, 14, 17]. As we are facing the challenge of reoccurring parts, object detection on it's own is insufficient to resolve ambiguities.

3D context Previous approach have explored improving object detection based on normal, size, height information extracted from dense 3D data [9, 10]. The most related approach to our work is using 3D layouts of object detectors in order for indoor scene understanding [3]. Most notably, our approach differs as it uses the layout information for the purpose of

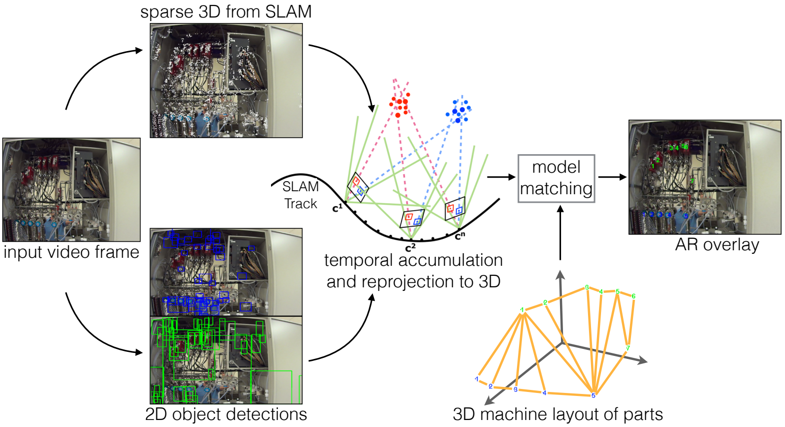


Figure 1: Overview of our system for object disambiguation.

disambiguation, we add an expectation over the matched viewpoints as well as address a different task (augmented reality).

Augmented Reality Application and maintenance work Augmented reality application have been studied for over two decades [1]. A recent overview of approaches, techniques and datasets can be found in [18] and is beyond the scope of exposition. The predominant body of work deals with registration and matching based on markers, low-level features or single objects. We argue for object centric evidence for ease of deployment. In particular, our use case of maintenance work calls for compositional models of multiple objects that allow for object disambiguation within the 3D context. We are not aware of previous efforts of establishing a public dataset for this purpose.

3 Object Disambiguation

As outlined before we seek a monocular system that operates markerless and exploits state-of-the-art object detectors in order to disambiguates objects as parts of a machine. For disambiguating multiple visual identical parts we fuse the object detector output with a SLAM system that allows us resolve ambiguities by reasoning over the spatial context. Figure 1 shows an overview of our system.

2D object detection At the core of our model are objects of which a machine is composed off. In order to localize them at test time we investigate a set of recent detectors: LINE-MOD2D[11], cascade with haar features [19] as well as HOG features [4], color-DPM[16]. As such models are all learning-based we can easily adopt our model to new machines and scenarios by training new detectors from examples and plugging them into our model. While the instance based detectors as well as the cascades are fast by design also the more complex detectors have seen recent extension so that they can be computed at interactive rates [5, 14, 17].

Sparse 3D from SLAM A sparse 3D point cloud is extracted from video using a monocular simultaneous localization and mapping (SLAM) system [13]. An extrinsic camera matrix is estimated based on the set of map points visible in the current frame, and the map is expanded as the camera is moved. The use of monocular image-based SLAM avoids the need for specialized sensors but also introduces challenges. In particular, tracking can fail if insufficient map points are visible (e.g. due to severe motion blur, absence of image features) to reliably triangulate the camera position as well as 3D estimate can be noisy due to complex scene geometry, occlusions and reflective surfaces. Tracking can be reinitialized at the cost of resetting the SLAM coordinate system. Direct use of the sparse 3D information has shown to yield unreliable matches wherefore we opt for integrating 2D and 3D information in the following step.

Temporal accumulation and reprojection to 3D We use the sparse 3D information generated by the SLAM system in order to reproject the 2D object detections to 3D. The depth for a particular detection is computed as the average over the covered SLAM features. As all preceding frames are connected by the SLAM track, we accumulate the reprojected 2D object detections over time. The benefits are threefold. First, object evidence is accumulated over time and can therefore compensate for missing or weak 2D detections in individual frames. Second, potential lag of the detection system can be compensated for as detections from previous frames are already available. Third, partial and ambiguous views of the machine that occur due to zooming in or shifting the viewpoint can be compensated due to previous viewpoints.

3D machine layout of parts We require a 3D machine layout that specifies the relative locations of each object. Such description are often provided by the machine specifications. Please note that the model does not have to be metric – nor do we require a complete 3D model or 3D scan of the machine. This is desirable for easy deployment and adaptation to new scenarios as a complete model can be specified by providing object detectors and a 3D layout.

Model Matching In order to match the 3D layout with N objects g_n to the observed detections d, we define an energy function that is taking into account the object appearance ($E_{appearance}$), deformation of the layout ($E_{deformation}$), scale (E_{scale}), viewpoint ($E_{viewpoint}$) as well as amount of matched objects (optional part in the deformation energy). The energy on scale and viewpoint capture an expectation of typical viewpoints the machine is viewed in. We seek the best match by finding an assignment of detections $d_1, \dots, d_N$ as well as a projection matrix M so that the following objective:

$$\underset{d_1, d_2, \dots, d_N, M}{\arg\min} \; E_{deformation} + E_{appearance} + E_{scale} + E_{viewpoint} \tag{1}$$

where

$$\begin{aligned}
E_{deformation} &= \frac{\sum_{n=1}^{N} \delta_n}{N} \sum_{n=1}^{N} \delta_n \cdot log(\left\|\bar{M}(P_{g_n}) - P_{d_n}\right\|) \\
E_{appearance} &= -\sum_{n=1}^{N} \delta_n \cdot A_{d_n} \\
E_{scale} &= \begin{cases} 0, & \bar{s} \in [\mu_s - 2 \cdot \sigma_s, \mu_s + 2 \cdot \sigma_s] \\ \infty, & otherwise \end{cases} \\
E_{viewpoint} &= \begin{cases} 0, & \bar{x} \in [\mu_x - 2 \cdot \sigma_x, \mu_x + 2 \cdot \sigma_x], \forall x = \{\alpha, \beta, \gamma\} \\ \infty, & otherwise \end{cases}
\end{aligned} \tag{2}$$

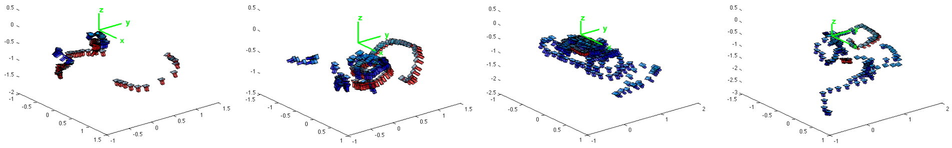


Figure 2: Visualization of the distribution for viewpoints in each machine. The red cameras are from the testing videos while the blue ones are from the training sets. The coordinate system is based on the 3D machine layout.

P_{g_n} and P_{d_n} denotes the 3D coordinate of g_n and d_n, while A_{d_n} is the detection score of the match d_n. The indicator variable δ_n is for handling the non-matched machine parts, where $\delta_n = 1$ if $\left\|\bar{M}(P_{g_n}) - P_{d_n}\right\|$ smaller than a threshold ε, and $\delta_n = 0$ otherwise. The 3D transformation $M(\cdot)$ includes the scale factor s, rotation matrix composed of three rotation angles $\{\alpha, \beta, \gamma\}$ and also a translation vector t. From the training videos of each machine, we compute the distribution of the scale factors and the rotation angles to get their mean μ and standard deviations σ. In the energy terms for both scale E_{scale} and the viewpoint $E_{viewpoint}$, we hard-constraint the scale factor $\bar{s}$ and rotation angles $\bar{x}, \forall x = \{\alpha, \beta, \gamma\}$ extracted from estimate 3D transformation $\bar{M}$ to be within 2 times of standard deviation from the mean. Figure 2 shows the viewpoints of training (blue) and testing (red) in the coordinate system of the machine layout.

In order to minimize the objective, we follow a RANSAC [8] pipeline by randomly selecting candidate alignments between the detections and the machine layout which results in an initial geometric transformation. According to this initial fitting, we iteratively refine the estimate [2] and re-associate the transformed groundtruth points to the closest detection points.

4 Experiments

We propose the first benchmark for an object disambiguation task in maintenance work that is composed of an annotated dataset as well as a metric that approximates human judgement. Furthermore, we evaluate our proposed model as well as its components.

4.1 Object Disambigutation DataSet (ObDiDaS)

We present the first annotated dataset that allows to quantify performance on a object disambiguation task as it frequently occurs in augmented reality settings and assistance for maintenance work. The dataset captures 4 machines composed of 13 components. Each machine is built of a subset of these potentially repeating components that occur in different spatial arrangements. We provide 14 videos with different viewing scenarios. For each videos we provide human annotation on every 60 frames (at 30fps), with in total 249 frames annotated and 6244 object annotations that specify the type as well as a unique identity. We take one video per machine as testing set and the rest is used for training. Examples are shown in the Figure 3. There are various types of difficulties in this dataset, including the wide changes in viewing angles of different object classes, occlusions and motion blur in the videos, reflective surfaces. The dataset allows studies of machine part detection and disambiguation, combination of 2D and and 3D cues based on monocular input, generalization

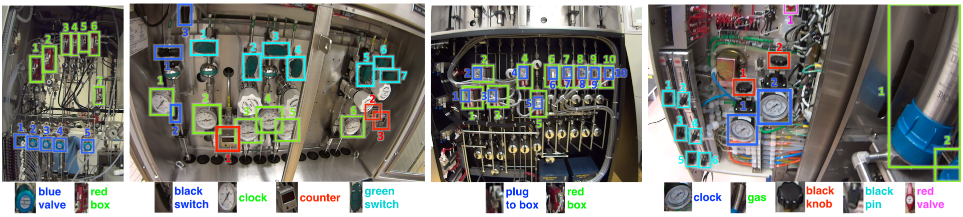


Figure 3: Example images for the dataset. In each image we use different color codes for different classes of machine parts. And each instance of the machine parts are labelled with unique identities of the machine.

between machines and adaptation to new scenarios. The ObDiDaS dataset is available at http://datasets.d2.mpi-inf.mpg.de/object-disambiguation/.

4.2 Object Disambiguation Metrics

While object detection metrics assess the performance of object localization in isolation, we are interested in a metric that captures the object disambiguation performance of a human if provided with the produced overlay. Therefore we propose a set of candidate metrics and then evaluate which one is closest to actual human judgement on the task.

Given a video frame with the SLAM extrinsic matrix H and the ground-truth annotation of N visible machine parts by bounding boxes B_{gt}. By using H to project the matches in RANSAC to this frame as bounding boxes, we denote the M visible ones with B_{est}. For each bounding box in ground-truth annotation or RANSAC estimation, they have the labels of their object classes and instance ids. (Note that we define $C(\cdot)$ and $I(\cdot)$ as functions to get the object class label and instance id of the bounding box)

Pascal Object Detection Criterion [Pascal] Inspired by the Pascal Challenge [6], for each $b_{gt}^{n}, n = 1 \cdots N$, we find the corresponding bounding box b_{est}^{m} with the same class label $C(b_{gt}^{n})$ and instance id $I(b_{gt}^{n})$ as b_{gt}^{n} from B_{est}, and measure the intersect-over-union metric between b_{gt}^{n} and b_{est}^{m}:

$$O(b_{gt}^{n}, b_{est}^{m}) = \frac{b_{gt}^{n} \cap b_{est}^{m}}{b_{gt}^{n} \cup b_{est}^{m}} \tag{3}$$

Then we define the Pascal metric as:

$$Score_{pascal} = \frac{1}{N} \sum_{n=1}^{N} \rho_n \text{ , where } \rho_n = \begin{cases} 1, & O(b_{gt}^{n}, b_{est}^{m}) > th \\ 0, & otherwise \end{cases} \tag{4}$$

The variable th is the overlapping threshold, which we set it to be 0.001 in our experiments.

Nearest Neighbor (within/across) We define the pairwise distance $dist(b_{gt}^{n}, b_{est}^{m})$ between b_{gt}^{n} and b_{est}^{m} as the euclidean distance between their box centers in the image coordinate. For each $b_{gt}^{n}, n = 1 \cdots N$, we find its nearest neighbor $b_{est}^{NN_{within}}$ from B_{est} with the same object class label: $B_{est}^{C} = \left\{ b_{est}^{m} | C(b_{est}^{m}) = C(b_{gt}^{n}) \right\}$. Then we define the NN_{within} metric as:

$$Score_{NN_{within}} = \frac{1}{N} \sum_{n=1}^{N} \rho_n \text{ , where } \rho_n = \begin{cases} 1, & I(b_{gt}^{n}) = I(b_{est}^{NN_{within}}) \\ 0, & otherwise \end{cases} \tag{5}$$

	machine 1	machine 2	machine 3	machine 4	average
Human Judge.	74.12%	100.00%	99.68%	70.57%	86.09%
Pascal	60.92%	98.68%	95.60%	25.10%	70.08%
NN (within)	57.05%	94.76%	88.06%	72.88%	78.19 %
NN (across)	56.07%	91.97%	65.20%	56.84%	67.52 %
1-to-1 (within)	77.55%	99.18%	99.68%	79.25%	88.92%
1-to-1 (across)	74.63%	96.92%	93.10%	72.45%	84.28 %

Table 1: Evaluation of different metrics.

Instead of finding the nearest neighbor with the same object class label, in metric NN_{across} we extend to search from all the bounding boxes in B_{est}, we denote the found nearest neighbor as $b_{est}^{NN_{across}}$. Then the metric NN_{across} is represented as:

$$Score_{NN_{across}} = \frac{1}{N}\sum_{n=1}^{N}\rho_n \text{ , where } \rho_n = \begin{cases} 1, & C(b_{gt}^{n}) = C(b_{est}^{NN_{across}}) \text{ and } I(b_{gt}^{n}) = I(b_{est}^{NN_{across}}) \\ 0, & otherwise \end{cases} \tag{6}$$

One-to-One (within/across) In comparison to computing the nearest neighbor, we further restrict to have one-to-one matching between b_{gt}^{n} and b_{est}^{m} and turn it to be a weighted bipartite matching scenario, where the weights are the $dist(b_{gt}^{n}, b_{est}^{m})$. We use Hungarian method [15] to solve this problem. Assume there are in total L object classes shown in this video frame, for each class l we build up the distance matrix by $B_{gt}^{l} = \left\{b_{gt}^{n_l} | C(b_{gt}^{n_l}) = l\right\}$ and $B_{est}^{l} = \left\{b_{est}^{m_l} | C(b_{est}^{m_l}) = l\right\}$. Then for each $b_{gt}^{n_l}$ we have the match $b_{est}^{One_{within}^{l}}$ after applying Hungarian method. We define the One_{within} metric as:

$$Score_{One_{within}} = \frac{1}{N}\sum_{l=1}^{L}\sum_{n_l=1}^{N_l}\rho_{n_l} \text{ , where } \rho_{n_l} = \begin{cases} 1, & I(b_{gt}^{n_l}) = I(b_{est}^{One_{within}^{l}}) \\ 0, & otherwise \end{cases} \tag{7}$$

Similar in nearest-neighbor metrics, we can also extend to do the one-to-one matching across classes. Hence we build up the distance matrix between B_{gt} and B_{est}. For each b_{gt}^{n} we have the match $b_{est}^{One_{across}}$. and the metric One_{across} is written as:

$$Score_{One_{across}} = \frac{1}{N}\sum_{n=1}^{N}\rho_n \text{ , where } \rho_n = \begin{cases} 1, & C(b_{gt}^{n}) = C(b_{est}^{One_{across}}) \text{ and } I(b_{gt}^{n}) = I(b_{est}^{One_{across}}) \\ 0, & otherwise \end{cases} \tag{8}$$

Evaluation of metrics In Table 1 we compare the proposed metrics to actual human judgement. We use the output of our full model. For the human judgement, we present the produced overlay to a human observer and assess in how many cases the correct object was identified. We observe that pascal metric significantly underestimates the system performance. We attribute this to an implicit matching that the human observer performs between the overlay and the observed machine parts. The nearest neighbor metric narrows the gap – at least for the case of matching within the object types (NN_{within}). The closest match to the true performance is obtained by the one-to-one metric. It takes further into account that the human observer also makes use of the context in order to align the overlay with the observed

	LINE-MOD	Haar cascade	HoG cascade	LBP cascade	color-DPM
avg. precision	10.81%	8.37%	13.38 %	8.90 %	36.73 %

Table 2: Evaluation of 2D object detectors.

	machine 1	machine 2	machine 3	machine 4	average
full model	74.63%	96.92%	93.10%	72.45%	84.28 %
no appearance	67.29%	93.32%	64.05%	51.06%	68.93%
no deformation	83.89%	95.05%	61.44%	40.30%	70.17%
no scale constraint	67.29%	98.53%	53.94%	43.57%	65.84%
no viewpoint constraint	38.01%	88.89%	43.04%	10.21%	45.04%
no scale and viewpoint	38.01%	88.89%	43.04%	10.21%	45.04%
no non-matched objects	74.61%	74.16%	64.10%	55.65%	67.13%

Table 3: Evaluation of different model components.

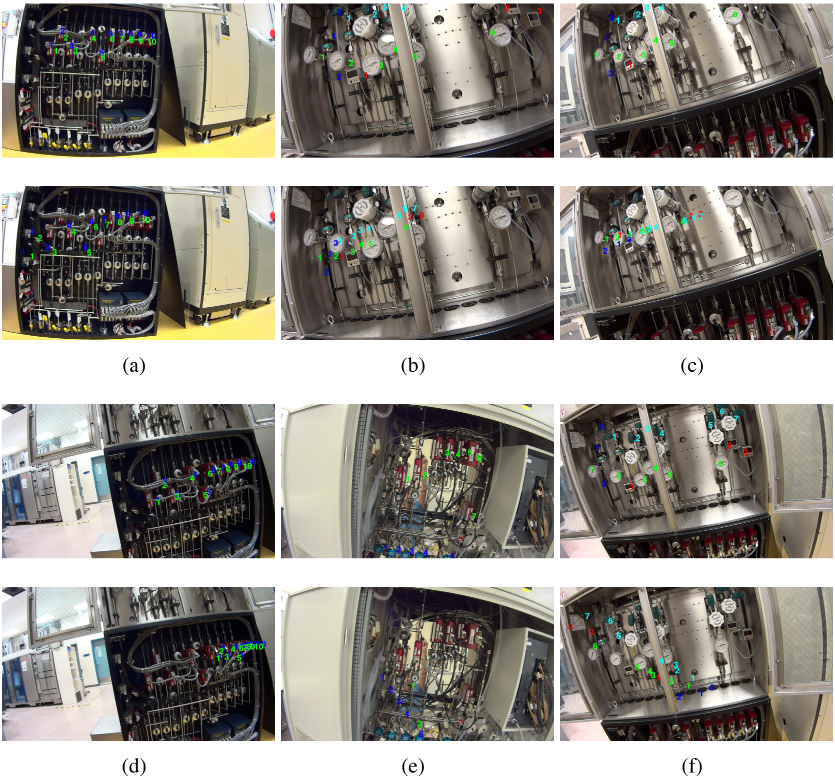
objects. As the "within" variant overestimates the performance we suggest and use the one-to-one (across) metric in the following experiments. A more detailed analysis of correlation scores on the individual object level has yielded the same ranking of metrics.

4.3 Evaluation

2D detectors in isolation We compare a range of 2D object recognition/detection algorithms on our new dataset: LINE-MOD2D[11], cascades with haar features [19] or histogram of gradient features [4] and color-DPM[16]. Table 2 shows average precision scores for the individual methods averaged across all objects and machines. This evaluation uses the pascal criterion as it evaluates object detection in isolation. We conclude that the color-DPM model outperforms the competitors by a large margin on this task. Therefore we will use it as a object detector throughout our experiments.

Full and partial models on object disambiguation task We evaluate our full model as well as switching energy terms off one at a time in order to provide further insights. Table 3 shows the individual performance numbers of the object disambiguation task (under the one-to-one-across metric), Figure 5 shows example results of our system in comparison to the groundtruth annotations and Figure 4 illustrates the effect on the output if parts of the matching energy are not used. We observe the most dramatic drop in performance if the viewpoint and scale constraints are not used, which results in a performance drop of almost 40%. The corresponding visualizations show that disabling this part of our model leads to estimates that exhibit a strong camera roll or suggest a fit beyond working distance. Appearance and the model deformation seem roughly equally important and both boost the performance by over 10%. Also our explicit treatment of non-matched objects is similarly important. Effects can again be observed in Figure 4 where a mismatch caused by a partial visible machine is remedied by the full model.

While our full model shows strong performance on machine 2 and 3, there is still a need for improvement on the other two. We attribute the missing performance to reflective surfaces (mirror in the back) that cause problems to the SLAM and detection system, complex 3D structure of machine layout, weak evidence from detector for certain objects and background clutter.



(a) (b) (c)

(d) (e) (f)

Figure 4: Top figure shows output of full model; result in bottom figure has a particular energy switched off (a)with/without appearance term (b)with/without deformation term (c)with/without non-matched objects handling (d)with/without scale term (a)with/without viewpoint term (a)with/without scale and viewpoint term

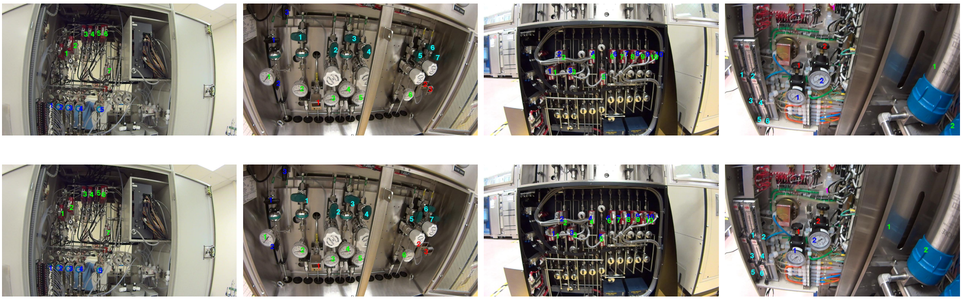


Figure 5: Example results. First row are examples for the groundtruh of each machine. Second row are the corresponding results from our proposed method.

5 Conclusion

We have investigated a object disambiguation task in a markerless augmented reality scenario, where object identities are inferred from monocular input by exploiting contextual information. To the best of our knowledge, we present the first dataset that allows to quantify the performance of such a system. We propose different metrics and compare them to human judgement. Our proposed metric gives a more realistic estimate of the system performance than a traditional object detection metric that consistently underestimates the system performance. Finally, we present an automatic system for object disambiguation that shows strong performance due to a matching formulation that is based on a composite energy function. We analyze the contribution of each component which underlines in particular the importance of modeling expectations over viewpoints and scales in the matching process.

Acknowledgement We acknowledge support from the Intel Visual Computing Institute (Intel VCI).

References

[1] Ronald T. Azuma. A survey of augmented reality. In *Presence: Teleoperators and Virtual Environments*, 1997.

[2] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics, 1992.

[3] W. Choi, Y. W. Chao, C. Pantofaru, and S. Savarese. Understanding indoor scenes using 3d geometric phrases. In *CVPR*, 2013.

[4] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[5] Thomas Dean, Mark A. Ruzon, Mark Segal, Jonathon Shlens, Sudheendra Vijayanarasimhan, and Jay Yagnik. Fast, accurate detection of 100,000 object classes on a single machine. In *CVPR*, 2013.

[6] Mark Everingham, Luc van Gool, Chris Williams, John Winn, and Andrew Zisserman. Pascal visual object class challenge. http://pascallin.ecs.soton.ac.uk/challenges/VOC/.

[7] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[8] Martin A Fischler and Robert C 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.

[9] Mario Fritz, Kate Saenko, and Trevor Darrell. Size matters: Metric visual search constraints from monocular metadata. In *Advances in Neural Information Processing Systems (NIPS)*. 2010.

[10] Stephen Gould, Paul Baumstarck, Morgan Quigley, Andrew Y. Ng, and Daphne Koller. Integrating visual and range data for robotic object detection. In *Workshop on Multi-camera and Multi-modal Sensor Fusion*, 2008.

[11] Stefan Hinterstoisser, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *ICCV*. IEEE, 2011.

[12] Yangqing Jia. Caffe: An open source convolutional architecture for fast feature embedding. http://caffe.berkeleyvision.org/, 2013.

[13] Georg Klein and David Murray. Parallel tracking and mapping for small AR workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007.

[14] Iasonas Kokkinos. Shufflets: Shared mid-level parts for fast object detection. In *ICCV*, 2013.

[15] Harold W Kuhn. The hungarian method for the assignment problem. In *50 Years of Integer Programming 1958-2008*, pages 29–47. Springer, 2010.

[16] Fahad Shahbaz Khan, Rao Muhammad Anwer, Joost van de Weijer, Andrew D Bagdanov, Maria Vanrell, and Antonio M Lopez. Color attributes for object detection. In *CVPR*, 2012.

[17] Hyun Oh Song, Stefan Zickler, Tim Althoff, Ross Girshick, Mario Fritz, Christopher Geyer, Pedro Felzenszwalb, and Trevor Darrell. Sparselet models for efficient multi-class object detection. In *ECCV*, 2012.

[18] Hideaki Uchiyama and Eric Marchand. Object detection and pose tracking for augmented reality: Recent approaches. In *Korea-Japan Joint Workshop on Frontiers of Computer Vision*, 2012.

[19] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.