A Search Space Strategy for Pedestrian Detection and Localization in World Coordinates

Mikael Nilsson[1], Martin Ahrnbom[1], Håkan Ardö[1] and Aliaksei Laureshyn[2]
[1] *Centre of Mathematical Sciences, Lund University, Lund, Sweden*
[2] *Traffic and Roads, Department of Technology and Society, Lund University, Lund, Sweden*

Keywords: Pedestrian, Detection, World Coordinates, Machine Learning, Camera Calibration.

Abstract: The focus of this work is detecting pedestrians, captured in a surveillance setting, and locating them in world coordinates. Commonly adopted search strategies operate in the image plane to address the object detection problem with machine learning, for example using scale-space pyramid with the sliding windows methodology or object proposals. In contrast, here a new search space is presented, which exploits camera calibration information and geometric priors. The proposed search strategy will facilitate detectors to directly estimate pedestrian presence in world coordinates of interest. Results are demonstrated on real world outdoor collected data along a path in dim light conditions, with the goal to locate pedestrians in world coordinates. The proposed search strategy indicate a mean error under 20 cm, while image plane search methods, with additional processing adopted for localization, yielded around or above 30 cm in mean localization error. This while only observing 3-4% of patches required by the image plane searches at the same task.

1 INTRODUCTION

Extracting relevant information from images is a key goal in many camera based applications. For example, pedestrian detection is one important and well-studied area of research (Dollár et al., 2012). Despite the extensive research on pedestrian detection, recent papers still show significant improvements, suggesting that a saturation point has not yet been reached (Dollár et al., 2014; Zhang et al., 2016b; Zhang et al., 2016a). These methods typically adopt a scale-space pyramid with sliding windows search or are combined with object proposal methods (Cheng et al., 2014; Zitnick and Dollár, 2014; van de Sande et al., 2011). However, this endeavor of detecting pedestrians is mainly focused on an image as the only input, and output as coordinates in the image plane.

Research has been conducted that focused on finding world information as a post-processing step following image plane detection (Andriluka et al., 2010; Xiang et al., 2014; Choy et al., 2015). While other works have exploited more explicit world, or three dimensional, reasoning, but only as a means of speeding up image plane search (Sudowe and Leibe, 2011; Benenson et al., 2012). Note that these methods all utilize an image plane search as a basis.

Other methods have exploited a more explicit use of 3D information for detection. For example, by prior camera calibration and geometric priors in sports tracking (Carr et al., 2012) and car detection (Nilsson and Ardö, 2014). However, those approaches make use of foreground/background segmentation rather than utilizing machine learning.

A key observation here is that there is a gap between exploiting directly available 3D information and machine learning, where state-of-the-art detectors work only in the image plane. In this paper, a core insight is that, with additional camera calibration information and geometric priors, one can produce a new search strategy, suitable for machine learning, to directly address the 3D localization problem. Thus what is proposed can be seen as a "glue" that ties 3D information together with patch based machine learning tools. Or, to put this in another light, what is proposed can be viewed as a specialized object proposal method resulting in rotated rectangles. Note though, that the "proposal part" here is directly formed using camera calibration, geometric priors and a world sampling grid. Furthermore, each object proposed has a corresponding world coordinate location.

The paper is organized as follows. The following section presents the real-world collected data and calibration used for evaluations. Section 3 presents how the framework proposed is formed from the image,

Nilsson, M., Ahrnbom, M., Ardö, H. and Laureshyn, A.
A Search Space Strategy for Pedestrian Detection and Localization in World Coordinates.
DOI: 10.5220/0006511800170024
In *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2018) - Volume 5: VISAPP*, pages 17-24
ISBN: 978-989-758-290-5

camera calibration and geometric prior to a search strategy resulting in patches that can be fed to a machine learning framework. Section 4 presents the machine learning used in the paper. Section 5 presents experiments comparing image plane search methodologies to the one proposed. Finally, conclusions are made in Section 6.

2 DATA COLLECTION AND CAMERA CALIBRATION

In an outdoor setup, using an Axis F41 camera with a F1015 lens mounted on top of a lamppost, pedestrians can be viewed on a piece of a path approximatively four meters wide.

A requirement for the proposed methodology is the existence of a calibrated camera. Note that the focus of this paper is not that of the camera calibration, it is on the design of search space utilizing a calibrated camera that can be feed to a patch based machine learning system. Due to the availability of a high precision GPS, a Leica GX1230 GG, the fixed camera could be calibrated with good results by marking twelve world reference positions, marked by spraying the ground at positions on the side of the path and measuring the world coordinates at each point. The points were then manually positioned in the camera image, see Fig. 1. Camera calibration was then performed using Tsai calibration (Tsai, 1987).

In a general description, let Θ denote all the Tsai calibration parameters, then a world point $\mathbf{p}_{world} = [x_w, y_w, z_w]^T$ can be projected as

$$\mathbf{p}_{image} = f(\mathbf{p}_{world}, \Theta) \qquad (1)$$

where f is a vector valued function involving all the world to image point operations in the Tsai method (Tsai, 1987) and $\mathbf{p}_{image} = [x, y]^T$ is the resulting image point. Furthermore, if N points are stacked into a matrix $\mathbf{P}$ of size $3 \times N$ then the operation $f(\mathbf{P}, \Theta)$ is one world to image mapping per column in $\mathbf{P}$ and the output a matrix of size $2 \times N$.

A dataset composed of ten images for each of twelve persons when passing the camera results in 120 images. Each pedestrian had their feet location annotated in world coordinates in each image. These will be explored for experimentation of pedestrian localization in world coordinates. Note that the viewpoint here is from a higher angle than usually appear in existing databases such as Caltech pedestrians (Dollár et al., 2009; Dollár et al., 2012) and INRIA pedestrians (Dalal and Triggs, 2005), where an eye level camera is typically applied, see examples in Fig. 2.

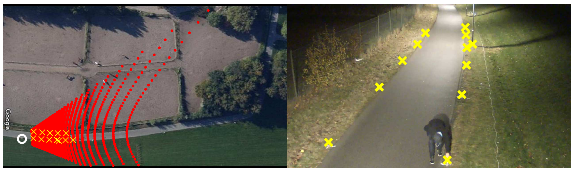


Figure 1: World position of camera (white), field of view (red) and calibration points (yellow) sprayed on the ground and measured with high precision GPS.

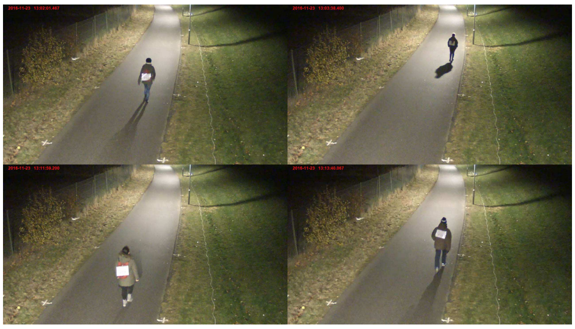


Figure 2: Examples of pedestrians from the outdoor scene.

3 IMAGE SEARCH STRATEGY AROUND 3D MODELS

The proposed search strategy works by transforming 3D models, here a 3D box, in the world to a sampling grid in the image plane. This sampling grid in the image is a rotated rectangle since camera rotation, roll in particular, as well as lens distortions produce tilted pedestrians, making an axis aligned rectangle less suitable. Furthermore, note that the method presented here can, in principle, be utilized with any 3D model in general. A general overview of the process for a given world point can be found in Fig. 3 and a specific example can be found in Fig. 4. The specifics for each step will follow.

With prior knowledge one can consider the presence of a pedestrian on several world coordinates of interest, as we will see later, a grid on the path for example, see Fig.6a. As will be seen later, such a grid, which utilize prior knowledge and a camera calibration, can produce far fewer patches to explore compared to a brute force image plane search. What follows is the proposed processing pipeline to get a classifier score from one such world point.

Given world coordinates for the feet of a pedestrian, a box is calculated around it. In general, a box of size *width* × *depth* × *height* is used to capture pedestrians in the world coordinate system. In this paper a standard box is considered to be $0.5 \times 0.5 \times 1.8$ meters. However, due to taller persons, and the desire to capture some context, an enlarged box of

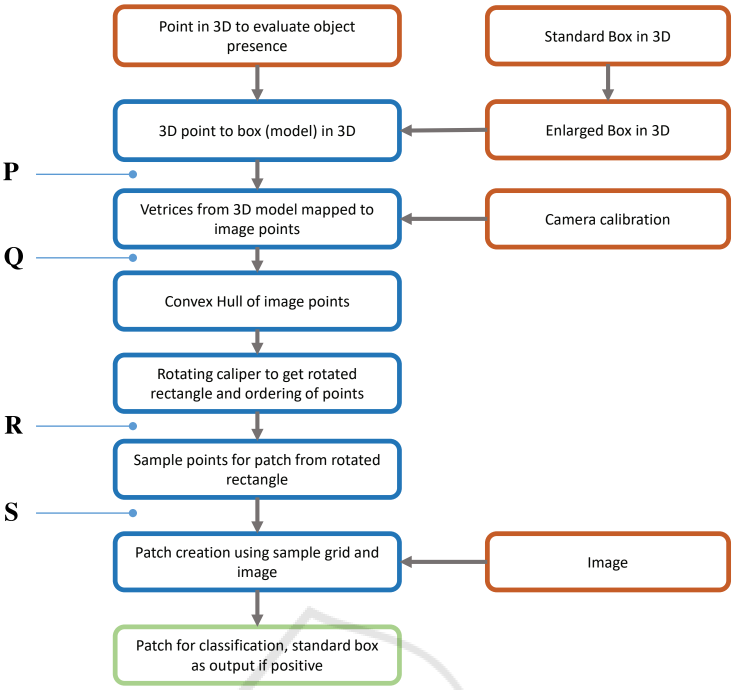
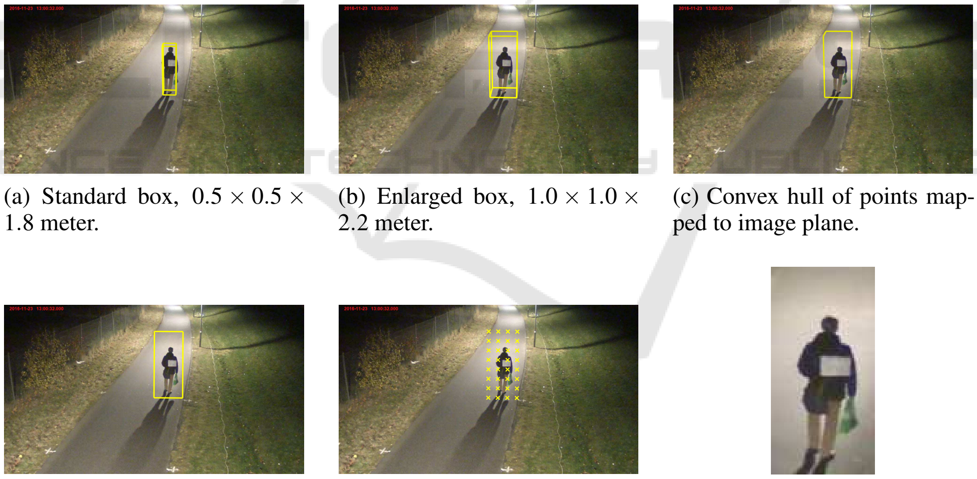

Point in 3D to evaluate object presence
Standard Box in 3D
3D point to box (model) in 3D
Enlarged Box in 3D
P
Vetrices from 3D model mapped to image points
Camera calibration
Q
Convex Hull of image points
Rotating caliper to get rotated rectangle and ordering of points
R
Sample points for patch from rotated rectangle
S
Patch creation using sample grid and image
Image
Patch for classification, standard box as output if positive

Figure 3: Principle for getting a rotated rectangle given a 3D point, an image, the camera calibration and a 3D shape.



(a) Standard box, 0.5 × 0.5 × 1.8 meter.

(b) Enlarged box, 1.0 × 1.0 × 2.2 meter.

(c) Convex hull of points mapped to image plane.

(d) Minimum rotate rectangle around the points.

(e) Sampling or rotated rectangle. Sampled with 64 × 128 points, here shown as 4 × 8 for clarity.

(f) Patch sampled on a 64 × 128 grid.

Figure 4: Steps a) to f) for creation of patches from a world coordinate box.

size 1.0 × 1.0 × 2.2 meters is employed. Thus, if a detection is found from the larger box, then the standard one is considered as the output detection box, see Fig. 4a and Fig. 4b for the different boxes. More formally, let W, D and H denote width, depth and height of the enlarged box, respectively. Then a matrix containing eight vertices, one at each column, of the box can be found as

$$\mathbf{P} = \begin{bmatrix} x_w - \frac{W}{2} & x_w - \frac{W}{2} & x_w + \frac{WH}{2} & x_w + \frac{W}{2} & x_w - \frac{W}{2} & x_w - \frac{W}{2} & x_w + \frac{W}{2} & x_w + \frac{W}{2} \\ y_w - \frac{D}{2} & y_w + \frac{D}{2} & y_w - \frac{D}{2} & y_w + \frac{D}{2} & y_w - \frac{D}{2} & y_w + \frac{D}{2} & y_w - \frac{D}{2} & y_w + \frac{D}{2} \\ z_w & z_w & z_w & z_w & z_w + H & z_w + H & z_w + H & z_w + H \end{bmatrix}. \quad (2)$$

The next step involves finding the corresponding vertices in the image plane. Let

$$\mathbf{Q} = f(\mathbf{P}, \Theta) \quad (3)$$

contain the eight points in the image plane. These points are now the main input towards finding a rotated rectangle in the image plane. The way this rotated rectangle is formed involves finding the minimum rotated rectangle enclosing all the points. This is achieved by first finding the convex hull and then finding the smallest-area enclosing rectangle of a polygon that has a side collinear with one of the edges of this convex hull. This methodology is known as the rotating calipers algorithm (Freeman and Shapira, 1975; Toussaint, 1983). See an example of finding the convex hull from **Q** in Fig. 4c and from the convex hull finding the minimum rotated rectangle in Fig. 4d. The output from the rotating calipers algorithm is four image points. To keep the order of these points in a consistent manner, they are ordered following the procedure outlined in Algorithm. 1. This process enforces a top-left, top-right, bottom-left and bottom-right ordering of the points. Note that the points of the rotated rectangle can be outside the image at times, examples of this can be seen in Fig. 5b. The result, in form of four ordered points, are stored in columns of a matrix

$$\mathbf{R} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_3 & \mathbf{p}_4 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \end{bmatrix}. \quad (4)$$

Algorithm 1: Order selection.

```
Input: A set of four image points. $W_{image}$ and $H_{image}$ being the width and height of the image, respectively.
Output: Four ordered image points
1: select point one as the one with minimum Euclidian distance to the point $[-W_{image}, -H_{image}]^T$ from the four points, remove this point from the set
2: select point two as the one with minimum Euclidian distance to the point $[2W_{image}, -H_{image}]^T$ from the three remaining points, remove this point from the set
3: select point three as the one with minimum Euclidian distance to the point $[-W_{image}, 2H_{image}]^T$ from the two remaining points, remove this point from the set
4: select the last point as the one left in the set
```

The next step involves producing a sampling grid matching a desired patch size. The four points in **R**, and a given patch size formed by S_{width} and S_{height}, are now used to produce a sampling grid in the image plane. This sampling grid is stored in matrix

$$\mathbf{S} = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \mathbf{s}_3 & \dots & \mathbf{s}_{S_{width} \cdot S_{height}} \end{bmatrix} \quad (5)$$

of size $2 \times S_{width} \cdot S_{height}$ and the construction of it can be found in Algorithm. 2.

Algorithm 2: Sampling rotated rectangle.

```
Input: $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$ and $\mathbf{p}_4$ are the ordered points, see Eq. (4). Chosen $S_{width}$ and $S_{height}$ being the desired width and height of the patch, respectively.
Output: The matrix $\mathbf{S}$ containing $S_{width} \cdot S_{height}$ sampling points in the image, see Eq. (5)
$k = 0$
$C = (S_{width} - 1)(S_{height} - 1)$
for $i = 1, 2, \dots, S_{height}$ do
    for $j = 1, 2, \dots, S_{width}$ do
        $k = k + 1$
        $w_1 = (S_{height} - i)(S_{width} - j)/C$
        $w_2 = (S_{height} - i)(j - 1)/C$
        $w_3 = (i - 1)(S_{width} - j)/C$
        $w_4 = (i - 1)(j - 1)/C$
        $\mathbf{s}_k = w_1\mathbf{p}_1 + w_2\mathbf{p}_2 + w_3\mathbf{p}_3 + w_4\mathbf{p}_4$
    end for
end for
```

Finally, the patch for classification is formed using the sampling points **S**. Example of a resulting patch can be found in Fig. 4f.

In the image view, boxes mapped from the world may become too small or heavily cropped at image borders to be useful. For this reason, three thresholds are enforced allowing a rotated rectangle to be used for processing only if it passes all three. The first and second threshold are on the width and height in pixels of the rotated rectangle, θ_{width} and θ_{height}, respectively. Another feature to threshold is the ratio of the sample points inside the image, denoted θ_{ratio}. Choices of thresholds used in the paper can be found in Table. 1. Examples of rejected boxes can be found in Fig. 5b.

4 CLASSIFICATION OF PATCH FROM ROTATED RECTANGLE

Given the possibility to produce patches from a given world position described in the previous section, it is now possible for produce training and test data for machine learning tools. In the dataset collected there are twelve unique individuals, each contributing with ten samples resulting in 120 tagged samples. Each sample contains only one single pedestrian in a given frame. Since only one pedestrian is visible in a given scene, both positive and negatives samples can be created with the following process: around the point, eight samples for the radii 5 cm and 10 cm, are collected, resulting in 17 positive patches for one tagging. For an initial negative sample set the radius of 40, 80 and 120 centimeters are used to collect eight samples for each, resulting in 24 negative patch samples. Hence, with this type of jittering one get 17 positive and 24 negative samples per annotated frame. The negative set will later be bootstrapped on images containing no pedestrians. All patches created here from a point in world coordinates are of size 64×128 from the grid in the rotated rectangle following proposed procedure, see Fig. 3 and Fig. 4. Note that there might be negative samples containing pedestrians to some degree using this approach, but those are not properly positioned to yield a good world localization, see Fig. 5a.

This collection of patches, as described above, can now be fed to basically any machine learning tool at one's disposal. For example, HOG-SVM (Dalal and Triggs, 2005), ACF (Dollár et al., 2014), HSC (Ren and Ramanan, 2013), Roerei (Benenson et al., 2013), VGG-16 (Simonyan and Zisserman, 2014) and various others (Zhang et al., 2016b). Note that the focus of this paper is not that of the the specific machine learning tools used, it is on the design of search space utilizing a calibrated camera that can be feed to a patch based machine learning system. Thus, while interesting, it is out of the scope of this paper to explore various methods for this task. Rather, the aim is to indicate the usefulness of the search strategy proposed. For this reason, a single method, inspired by HSC (Ren and Ramanan, 2013), employing a sparse coding and logistic regression framework is adopted. In particular, a sparse coding, with ability to incorporate supervised information (Nilsson, 2016), is used to build a discriminative dictionary from all non-overlapping 8×8 patches from all samples. This was done by forming a discriminate matrix with $K = 32$ atoms, where eight atoms was allocated for positive, 16 for negative and eight as a do-not-care region. For more in-depth details on this discriminative dictionary learning, the reader is referred to the work by Nilsson (Nilsson, 2016). Then, using this dictionary, codes are extracted from all training samples and feed to logistic regression with elastic net regularization (Nilsson, 2014).

In this setup a twelve-fold cross validation is used, implying that all samples from one person is left out in training and evaluated in a full search. During detection this full search is composed of a sampling grid produced with ten centimeter distances between the points on the path of interest, this resulting in 7047 patches to evaluate after rejecting rotated rectangles with $\theta_{width} = 32$, $\theta_{height} = 64$ and $\theta_{ratio} = 0.9$. Examples of rejected boxes with these thresholds can be found in Fig. 5b.

In image plane searches the Intersection over Union (IoU) is a commonly adopted measure in a Non-Maximum Suppression (NMS) method to prune detections. The results produced here can utilize the classifier scores on the grid in world coordinates, and could benefit from this knowledge. Hence, a World NMS (WNMS) is introduced instead. This WNMS is using Euclidian distance between points, in meters, and a threshold on this distance as a measure to decide overlap. In general, this WNMS have three parameters γ_{det}, γ_{radius} and γ_{count} where γ_{det} is the classifier threshold for detection, γ_{radius} the radius in meters to define overlap and γ_{count} is the number of detections that are pruned into the maximum one. In all examples in this paper $\gamma_{det} = 0.5$ (on the logistic regression output), $\gamma_{radius} = 0.5$ and $\gamma_{count} = 3$. An example of detection scores and the final WNMS output vs ground truth can be found in Fig. 6.

5 EXPERIMENTS

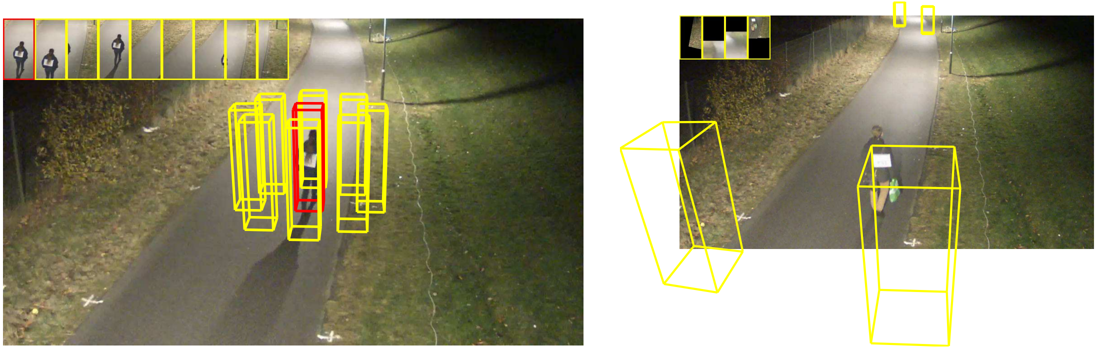
The experiments focus on investigating the world localization of pedestrians. First, baselines are formed by utilizing image plane searches. The goal is to see how far one can come by first running an image plane detector and then, as a second step, aim to find the world coordinates. Then the proposed sampling strategy is investigated. Finally, a comparison between the baselines and the proposed method is performed.

The core parameters introduced throughout the paper has been stated and described previously. A summary of them can be found in Table. 1, these values are used throughout the paper.

5.1 Image Plane Localization Baselines

5.1.1 Image Plane Detection

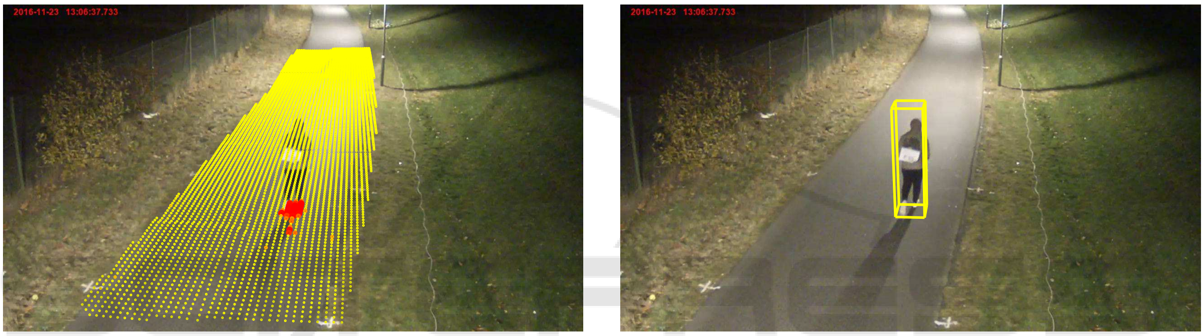
As a first step, an image plane scanning is performed. Three different methods are adopted as baselines. First, the same sparse coding and logistic regression framework, using twelve-fold cross validation, adopted for the proposed methods is used here. The



(a) One positive sample out of the 17 shown as red, and eight negative out of the 24 shown as yellow. Note that some negatives contain the pedestrian but are not aligned for proper localization.

(b) Examples of ignored boxes due to thresholds.

Figure 5: Positive and negative samples and examples of ignored boxes.



(a) Sampling grid and classifier scores. Yellow indicates low scores and red high scores.

(b) Final detection after WNMS.

Figure 6: From scores on the grid in the ground-plane to detction box after WNMS.

Table 1: Parameter choices.

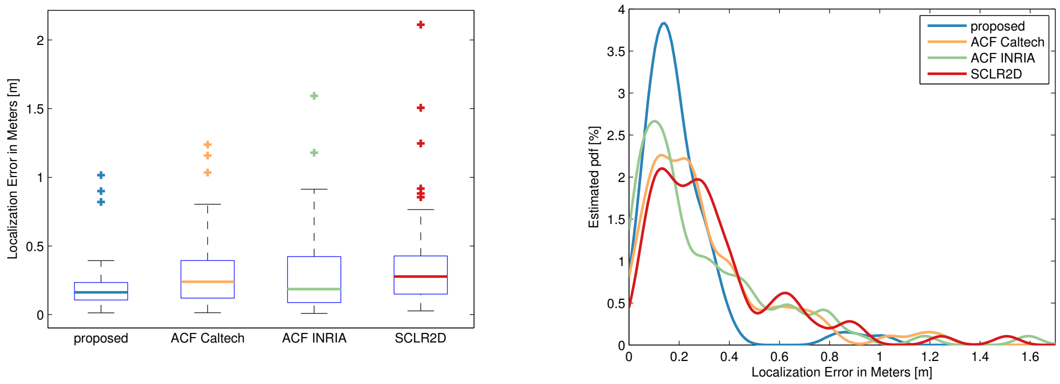
parameter	S_{width}	S_{height}	θ_{width}	θ_{height}	θ_{ratio}	γ_{det}	γ_{radius}	γ_{count}
value	64	128	32	64	0.9	0.5	0.5	3

difference is that axis aligned patches, formed from the same points that were used for rotated boxes in the proposed method, are used for training and a scale-space and sliding window search is adopted instead. This method used scaling 1.25 in the scale space and jumped three pixels, resulting in 187769 patches to evaluate. This method is denoted SCLR2D and resulted in 112 true positives and 30 false positives on the 120 images. A true positive was indicated if the Intersection over Union (IoU) with a ground truth bounding box was over 0.65.

In addition, the Aggregated Channel Features (ACF) method (Dollár et al., 2014) is employed on the 120 images containing one pedestrian each. Using an ACF detector trained on the INRIA database (Dalal and Triggs, 2005) resulted in 113 true positives and 64 false positives. The ACF, trained on the Caltech dataset (Dollár et al., 2009; Dollár et al., 2012), resulted in 97 true positives and ten false positives.

5.1.2 Image Plane Detection Box to World Coordinates

Note that the goal here is to do localization in world coordinates using the calibration. Therefore, conversion from the image plane detection box to world coordinates is required as a second step for image plane searches. A method positioning a fixed point in a normalized box in the image plane (width and height equal to one and top left position at $(0,0)$) was employed. For a given bounding box this fixed point, in normalized coordinates, is then mapped back to the detected bounding box, resulting in a point in the image. This point is then transformed to world coordinates at the ground plane using the camera cali-



(a) Box-Whisker plots of the localization errors.

(b) Parzen window density estimation of localization errors with bandwidth of 0.05.

Figure 7: Evaluation metrics on localization. Best viewed in color.

bration. In order to decide this fixed point in the normalized box, an optimization finding the best mean localization error in the world from all the detections was performed. Note that this is a optimistic localization done here, since the point in the normalized box is found on the same boxes as it is evaluated on. Nevertheless, this resulted in world localization errors for the ACF INRIA, ACF Caltech detector and localization baselines.

5.2 Direct Localization using Sampling Strategy

Here the world localization employing the proposed search strategy is employed. Hence, a resulting hypothesis for each of the points in a sampled grid, with ten centimeter distances, in the world coordinates are produced and detections follows the World NMS (WNMS), see Fig. 6. This detector resulted in 103 true positives and 11 false positives. A positive was considered to be a detection point located within a radius of two meters from the manual annotation. This choice of 2 m was to match some of the errors received using the baselines with an IoU choice of 0.65 in the image plane as detection choice, and could in practice been lower. More importantly, this localization was achieved by exploring only 7047 patches due to the exploitation of the camera calibration and prior knowledge. This to compare to 312977 patches explored by the ACF methods and 187769 patches by SCLR2D using only the image.

5.3 Comparisions

A set of 74 detections, those detections that all were detected by all three methods, were used for localization evaluation. The results gave mean error of 19.9 cm for the proposed method, 30.5 cm for ACF trained on Caltech, 29.4 cm for ACF trained on INRIA and 36.3 cm for SCLR2D. For a more detailed study of the statistics of the errors in meters, the Box-Whisker plot (Tukey, 1977) and density estimation using a Parzen window (Parzen, 1962) can be found in Fig. 7a and Fig. 7b, respectively. Note that the method proposed has far fewer outliers, in form of errors over 0.5 m. The main takeaways from these experiments and the proposed search space are:

- With a given camera calibration, it is possible to design a search space with no need for additional processing for world localization.
- The number of patches needed to be explored can be far fewer compared to image plane search.
- The localization accuracy can actually benefit in the process.

6 CONCLUSIONS

A search strategy producing a rotated rectangle from a camera calibration and prior 3D shape has been proposed and investigated. By exploiting camera calibration information it has been shown that the sampling method can be used to facilitate machine learning that can directly produce classification scores in world locations of interest. This approach lead to accurate localization of pedestrians in world coordinates with a mean error of 19.9 cm, while three image plane detectors, adopted for the localization task, resulted in mean errors of 29.4 cm, 30.5 cm and 36.3 cm. This while only observing less than 3-4% of patches needed in the image plane search. Future work involves exploring the methodology proposed on more views, more crowded scenarios, investigating various other machine learning methods for the task and applying the method on other objects.

REFERENCES

Andriluka, M., Roth, S., and Schiele, B. (2010). Monocular 3d pose estimation and tracking by detection. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 623–630.

Benenson, R., Mathias, M., Timofte, R., and Van Gool, L. (2012). Fast stixels estimation for fast pedestrian detection. In *ECCV, CVVT workshop*.

Benenson, R., Mathias, M., Tuytelaars, T., and Gool, L. V. (2013). Seeking the strongest rigid detector. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3666–3673.

Carr, P., Sheikh, Y., and Matthews, I. (2012). Monocular object detection using 3d geometric primitives. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part I*, ECCV'12, pages 864–878, Berlin, Heidelberg. Springer-Verlag.

Cheng, M. M., Zhang, Z., Lin, W. Y., and Torr, P. (2014). Bing: Binarized normed gradients for objectness estimation at 300fps. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3286–3293.

Choy, C. B., Stark, M., Corbett-Davies, S., and Savarese, S. (2015). Enriching object detection with 2d-3d registration and continuous viewpoint estimation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2512–2520.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1.

Dollár, P., Wojek, C., Schiele, B., and Perona, P. (2009). Pedestrian detection: A benchmark. In *CVPR*.

Dollár, P., Wojek, C., Schiele, B., and Perona, P. (2012). Pedestrian detection: An evaluation of the state of the art. *PAMI*, 34.

Dollár, P., Appel, R., Belongie, S., and Perona, P. (2014). Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1532–1545.

Freeman, H. and Shapira, R. (1975). Determining the minimum-area encasing rectangle for an arbitrary closed curve. *Commun. ACM*, 18(7):409–413.

Nilsson, M. (2014). Elastic net regularized logistic regression using cubic majorization. In *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*, pages 3446–3451.

Nilsson, M. (2016). Sparse coding with unity range codes and label consistent discriminative dictionary learning. In *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*.

Nilsson, M. and Ardö, H. (2014). In search of a car utilizing a 3d model with context for object detection. In *The International Conference on Computer Vision Theory and Applications (VISAPP)*, volume 2, pages 419–424.

Parzen, E. (1962). On estimation of a probability density function and mode. *Ann. Math. Statist.*, 33(3):1065–1076.

Ren, X. and Ramanan, D. (2013). Histograms of sparse codes for object detection. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3246–3253.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.

Sudowe, P. and Leibe, B. (2011). Efficient Use of Geometric Constraints for Sliding-Window Object Detection in Video. In *International Conference on Computer Vision Systems (ICVS'11)*.

Toussaint, G. (1983). Solving geometric problems with the rotating calipers. *In Proc. IEEE MELECON '83*, pages 10—02.

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

Tukey, J. (1977). *Exploratory Data Analysis*. Behavioral Science: Quantitative Methods. Addison-Wesley, Reading, Mass.

van de Sande, K. E. A., Uijlings, J., Gevers, T., and Smeulders, A. (2011). Segmentation as Selective Search for Object Recognition. In *ICCV*.

Xiang, Y., Mottaghi, R., and Savarese, S. (2014). Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*.

Zhang, L., Lin, L., Liang, X., and He, K. (2016a). *Is Faster R-CNN Doing Well for Pedestrian Detection?*, pages 443–457. Springer International Publishing, Cham.

Zhang, S., Benenson, R., Omran, M., Hosang, J., and Schiele, B. (2016b). How far are we from solving pedestrian detection? In *CVPR*.

Zitnick, C. L. and Dollár, P. (2014). *Edge Boxes: Locating Object Proposals from Edges*, pages 391–405. Springer International Publishing, Cham.