Eliminating the Blind Spot: Adapting 3D Object Detection and Monocular Depth Estimation to 360° Panoramic Imagery

Grégoire Payen de La Garanderie, Amir Atapour Abarghouei, and Toby P. Breckon

Department of Computer Science
Durham University
gregoire.p.payen-de-la-garander@durham.ac.uk
{amir.atapour-abarghouei,toby.breckon}@durham.ac.uk

Abstract. Recent automotive vision work has focused almost exclusively on processing forward-facing cameras. However, future autonomous vehicles will not be viable without a more comprehensive surround sensing, akin to a human driver, as can be provided by 360° panoramic cameras. We present an approach to adapt contemporary deep network architectures developed on conventional rectilinear imagery to work on equirectangular 360° panoramic imagery. To address the lack of annotated panoramic automotive datasets availability, we adapt a contemporary automotive dataset, via style and projection transformations, to facilitate the cross-domain retraining of contemporary algorithms for panoramic imagery. Following this approach we retrain and adapt existing architectures to recover scene depth and 3D pose of vehicles from monocular panoramic imagery without any panoramic training labels or calibration parameters. Our approach is evaluated qualitatively on crowd-sourced panoramic images and quantitatively using an automotive environment simulator to provide the first benchmark for such techniques within panoramic imagery.

Keywords: object detection, panoramic imagery, monocular 3D object detection, style transfer, monocular depth, panoramic depth, 360 depth

1 Introduction

Recent automotive computer vision work (object detection [51,50], segmentation [3], stereo vision [38,49], monocular depth estimation [41,26,1]) has focused almost exclusively on the processing of forward-facing rectified rectilinear vehicle mounted cameras. Indeed by sharp contrast to the abundance of common evaluation criteria and datasets for the forward-facing camera case [19,18,4,39,48,16,2], there are no annotated evaluation datasets or frameworks for any of these tasks using 360° view panoramic cameras.

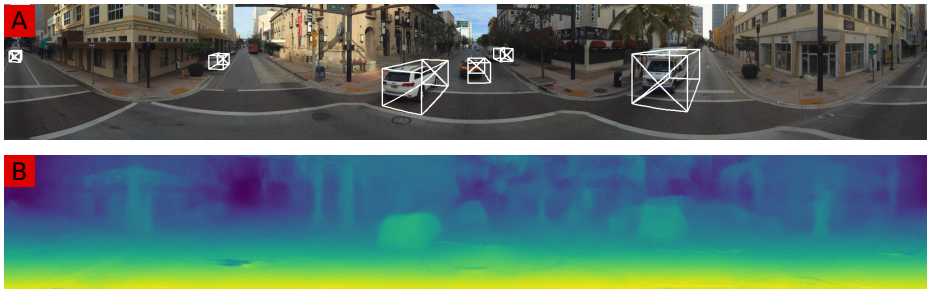


Fig. 1: Our monocular panoramic image approach. A: 3D object detection. B: depth recovery.

However, varying levels of future vehicle autonomy will require full 360° situational awareness, akin to that of the human driver of today, in order to be able to function across complex and challenging driving environments. One popularly conceived idea of capturing this awareness is to use active sensing in the form of 360° LIDAR, however this is currently an expensive, low-resolution method which does not encompass the richness of visual information required for high fidelity semantic scene understanding. An alternative is to fuse the information from multiple cameras surrounding the vehicle [29] and such methods have been used to fuse between a forward-facing camera and LIDAR [10,27]. However, here opportunities are lost to share visual information in early stages of the pipeline with further computational redundancy due to overlapping fields of view. Alternatively the imagery from a multiview setup can be stitched into a 360° panorama [5]. A roof mounted on-vehicle panoramic camera offers superior angular resolution compared to any LIDAR, is 1–2 orders of magnitude lower cost and provides rich scene colour and texture information that enables full semantic scene understanding [35].

Panoramic images are typically represented using an equirectangular projection (Fig. 1A); in contrast, a conventional camera uses a rectilinear projection. In this projection, the image-space coordinates are proportional to latitude and longitude of observed points rather than the usual projection onto a focal plane as shown in Fig. 1A.

Recent work on panoramic images has largely focused on indoor scene understanding [63,61], panoramic to rectilinear video conversion [57,34,42] and dual camera 360° stereo depth recovery [30,46]. However, no work to date has explicitly tackled contemporary automotive sensing problems.

By contrast, we present an approach to adapt existing deep architectures, such as convolutional neural networks (CNN) [6,26], developed on rectilinear imagery to operate on equirectangular panoramic imagery. Due to the lack of explicit annotated panoramic automotive training datasets, we show how to reuse existing non-panoramic datasets such as KITTI [19,18] using style and projection transformations, to facilitate the cross-domain retraining of contemporary algorithms for panoramic imagery. We apply this technique to estimate dense

monocular depth (see example in Fig. 1B) and to recover the full 3D pose of vehicles (Fig. 1B) from panoramic imagery. Additionally, our work provides the first performance benchmark for the use of these techniques on 360° panoramic imagery acting as a key driver for future research on this topic. Our technique is evaluated qualitatively on crowd-sourced 360° panoramic images from Mapillary [45] and quantitatively using ground truth from the CARLA [13] high fidelity automotive environment simulator[1].

2 Related Work

Related work is considered within panoramic imagery (Section 2.1), monocular 3D object detection (Section 2.2), monocular depth recovery (Section 2.3) and domain adaptation (Section 2.4).

2.1 Object Detection within Panoramic Imagery

Even though significant strides have been made in rectilinear image object proposal [33] and object detection methods utilizing deep networks [51,55,25,24,31,6], comparatively limited literature exists within panoramic imagery.

Deng *et al.* [11] adapted, trained and evaluated Faster R-CNN [51] on a new dataset of 2,000 indoor panoramic images for 2D object detection. However their approach does not handle the special case of object wrap-around at the equirectangular image boundaries.

Recently, object detection and segmentation has been applied directly to equirectangular panoramic images to provide object detection and saliency in the context of virtual cinematography [34,42] using pre-trained detectors such as Faster R-CNN [51]. Su and Grauman [56] introduce a Flat2Sphere technique to train a spherical CNN to imitate the results of an existing CNN facilitating large object detection at any angle.

In contemporary automotive sensing problems, the required vertical field of view is small as neither the view above the horizon nor the view directly underneath the camera have any useful information for those problems. Therefore, the additional complexity of the spherical CNN introduced by [56] is not needed in the specific automotive context. Instead we show how to reuse existing deep architectures without requiring any significant architectural changes.

2.2 Monocular 3D Object Detection

Prior work on 3D pose regression in panorama is mostly focused on indoor scene reconstruction such as PanoContext by Zhang *et al.* [63] and Pano2CAD by Xu *et al.* [61]. The latter retrieves the object poses by regression using a bank of known CAD (Computer-Aided Design) models. In contrast, our method does not require any *a priori* knowledge of the object geometry.

[1] for future comparison our code, models and evaluation data is publicly available at: https://gdlg.github.io/panoramic

Contemporary end-to-end CNN driven detection approaches are based on the R-CNN architecture introduced by Girshick [23]. Successive improvements from Fast-RCNN [22] and Faster-RCNN [51] increased the performance by respectively sharing feature maps across proposals and generating the proposals using a Region Proposal Network (RPN) instead of traditional techniques based on sliding windows. This allowed a unified end-to-end training of the network to solve the combined detection and classification tasks. More recently, Yang *et al.* [62] and Cai *et al.* [6] introduced a multi-scale approach by pooling the region proposals from multiple layers in order to reduce the number of proposals needed as well as to improve performance on smaller objects such as distant objects.

While most of the work has focused on 2D detection, the work of Chen *et al.* [9,10] leverages 3D pointcloud information gained either from stereo or LIDAR modalities to generate 3D proposals which are pruned using Fast R-CNN. Whereas these works use complex arrangements using stereo vision, handcrafted features or 3D model regression, recent advances [8,47,7] show that it is actually possible to recover the 3D pose from monocular imagery. Chen *et al.* [8] use post-processing of the proposals within an energy minimization framework assuming that the ground plane is known. Chabot *et al.* [7] use 3D CAD models as templates to regress the 3D pose of an object given part detections; while Mousavian *et al.* [47] show the 3D pose can be recovered without any template assumptions using carefully-expressed geometrical constraints. In this work, we propose a new approach, similar to [47], however without explicitly-expressed geometrical constraints, which performs on both rectilinear and equirectangular panoramic imagery without any knowledge of the ground plane position with respect to the camera.

2.3 Monocular Depth Estimation

Traditionally dense scene depth is recovered using multi-view approaches such as structure-from-motion and stereo vision [54], relying on an explicit handling of geometrical constraints between multiple calibrated views. However recently with the advance of deep learning, it has been shown that dense scene depth can also be recovered from monocular imagery.

After the initial success of classical learning-based techniques such as [52,53], depth recovery was first approached as a supervised learning problem by the depth classifier of Ladický *et al.* [41] and deep learning-based approaches such as [15,43]. However, these techniques are based on the availability of high-quality ground truth depth maps, which are difficult to obtain. In order to combat the ground truth data issue, the method in [1] relies on readily-available high-resolution synthetic depth maps captured from a virtual environment and domain transfer to resolve the problem of domain bias.

On the other hand, other monocular depth estimation methods have recently emerged that are capable of performing depth recovery without the need for large quantities of ground truth depth data. Zhou *et al.* [64] estimate monocular depth and ego-motion using depth and pose prediction networks that are trained via

view synthesis. The approach proposed in [40] utilizes a deep network semi-supervised by sparse ground truth depth and then reinforced within a stereo framework to recover dense depth information.

Godard *et al.* [26] train their model based on left-right consistency inside a stereo image pair during training. At inference time, however, the model solely relies on a single monocular image to estimate a dense depth map. Even though said approach is primarily designed to deal with rectilinear images, in this work we further adapt this model to perform depth estimation on equirectangular panoramic images.

2.4 Domain Adaptation and Style Transfer

Machine learning architectures trained on one dataset do not necessarily transfer well to a new dataset – a problem known as dataset bias [58] or covariate shift [28]. A simple solution to dataset bias would be fine-tuning the trained model using the new data but that often requires large quantities of ground truth, which are not always readily-available.

While many strategies have been proposed to reduce the feature distributions between the two data domains [44,21,12,59], a novel solution was recently proposed in [1] that uses image style transfer as a means to circumvent the data domain bias.

Image style transfer was first proposed by Gatys *et al.* [17] but since then remarkable advances have been made in the field [36,60,14,20]. In this work, we attempt to transform existing rectilinear training images (such as KITTI [19,18]) to share the same style as our panoramic destination domain (Mapillary [45]). However, these two datasets have been captured in different places and share no registration relationship. As demonstrated in [1,32], unpaired image style transfer solved by CycleGAN [65], can be used to transfer the style between two data domains that possess approximately similar content.

2.5 Proposed Contributions

Overall the main contributions, against the state of the art [6,26,19,18,47,26,13], presented in this work are:

- a novel approach to convert deep network architectures [6,26] operating on rectilinear images for equirectangular panoramic images based on style and projection transformations;
- a novel approach to reuse and adapt existing datasets [19,18] in order to train models for panoramic imagery;
- the subsequent application of these approaches for monocular 3D object detection using a simpler formulation than earlier work [47], additionally operable on conventional imagery without modification;
- further application of these techniques to monocular depth recovery using an adaptation of the rectilinear imagery approach of Godard *et al.* [26];
- provision of the first performance benchmark based on a new synthetic evaluation dataset (based on CARLA [13]) for this new challenging task of automotive panoramic imagery depth recovery and object detection evaluation.

3 Approach

We first describe the mathematical projections underlining rectilinear and equirectangular projections and the relationship between the two required to enable our approach within panoramic imagery (Sec. 3.1). Subsequently we describe the dataset adaptation (Sec. 3.2), its application to monocular 3D pose recovery (Sec. 3.3) and depth estimation (Sec. 3.4) and finally the architectural modifications required for inference within panoramic imagery (Sec. 3.5).

3.1 Rectilinear and Equirectangular Projections

Projection using a classical rectified rectilinear camera is typically defined in terms of its camera matrix P. Given the Cartesian coordinates (x, y, z) of a 3D scene point in camera space, its projection (u_{lin}, v_{lin}) is defined as:

$$\begin{bmatrix} u_{lin} \\ v_{lin} \end{bmatrix} = \left\lfloor P \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \right\rfloor \tag{1}$$

where $\lfloor \cdot \rfloor$ denotes the homogeneous normalization of the vector by its last component. The camera matrix P is conventionally defined as:

$$P = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

where f and (c_x, c_y) are respectively the focal length and the principal point of the camera.

The rectilinear projection as defined in Eqn. 1 is advantageous because the camera matrix P can be combined with further image and object space transformations into a single linear transformation followed by an homogeneous normalization. However, this transformation can also be written as:

$$\begin{bmatrix} u_{lin} \\ v_{lin} \end{bmatrix} = P \cdot \begin{bmatrix} x/z \\ y/z \\ 1 \end{bmatrix} \tag{3}$$

This formulation (Eqn. 3) is convenient because the image-space coordinates are expressed in terms of the ratio x/z and y/z which are the same regardless of the distance from the 3D scene point to the camera.

In contrast, the equirectangular projection is defined in terms of the longitude and latitude of the point. The longitude and latitude, respectively (λ, ϕ), are defined as:

$$\lambda = \arctan\, x/z \tag{4}$$

$$\phi = \arcsin\, y/r \quad \text{where } r = (x^2 + y^2 + z^2)^{\frac{1}{2}} \tag{5}$$

The latitude definition in Eqn. 5 can be conveniently rewritten in terms of the ratios x/z and y/z as in Eqn. 3 for rectilinear projections:

$$\phi = \arcsin \frac{y/z}{r} \quad \text{where } r = (x/z^2 + y/z^2 + 1^2)^{\frac{1}{2}} \tag{6}$$

For the sake of simplicity, this computation of the latitude and longitude from the Cartesian coordinates can be represented as a function Γ:

$$\begin{bmatrix} \lambda \\ \phi \end{bmatrix} = \Gamma \left(\begin{bmatrix} x \\ y \\ z \end{bmatrix} \right) = \Gamma \left(\begin{bmatrix} x/z \\ y/z \\ 1 \end{bmatrix} \right) \tag{7}$$

Finally, we define an image transformation matrix T_{equi} which transforms the longitude and latitude to image space coordinates (u_{equi}, v_{equi}):

$$\begin{bmatrix} u_{equi} \\ v_{equi} \\ 1 \end{bmatrix} = T_{equi} \cdot \begin{bmatrix} \lambda \\ \phi \\ 1 \end{bmatrix} = T_{equi} \cdot \Gamma \left(\begin{bmatrix} x/z \\ y/z \\ 1 \end{bmatrix} \right) \tag{8}$$

The matrix T_{equi} can be defined as:

$$T_{equi} = \begin{bmatrix} \alpha & 0 & c_\lambda \\ 0 & \alpha & c_\phi \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

where α is an angular resolution parameter akin to the focal length. Like the focal length, it can be defined in terms of the field of view:

$$\alpha = \text{fov}_\lambda / w = \text{fov}_\phi / h \tag{10}$$

where $\text{fov}_\lambda, \text{fov}_\phi, w, h$ are respectively the image horizontal field of view, vertical field of view; width and height. In contrast to rectilinear imagery, where the focal length is difficult to determine without any kind of camera calibration, the equirectangular imagery, commonly generated by panoramic cameras from the raw dual-fisheye pair, can be readily used without any prior calibration because the angular resolution $\alpha = 2\pi/w$ depends only on the image width. Therefore, approaches that would require some knowledge of the camera intrinsics of rectilinear images (*e.g.* monocular depth estimation) can be readily used on any 360° panoramic image without any prior calibration.

By coupling the definitions of both the rectilinear and equirectangular projections in terms of the ratios x/z and y/z (Eqn. 3 & 8), we establish the relationship between the coordinates in the rectilinear projection and equirectangular projection for the given matrices P and T_{equi}:

$$\begin{bmatrix} u_{equi} \\ v_{equi} \\ 1 \end{bmatrix} = T_{equi} \cdot \Gamma \left(P^{-1} \cdot \begin{bmatrix} u_{lin} \\ v_{lin} \\ 1 \end{bmatrix} \right) \tag{11}$$

This enables us to reproject an image from one projection to another, such as from the rectilinear image (Fig. 2A) to an equirectangular image (Fig. 2C) and vice versa — a key enabler for the application of our approach within panoramic imagery.

3.2 Dataset Adaptation

In our approach, the source domain is the KITTI [19,18] dataset of rectilinear images captured using a front-facing camera rig (1242×375 image resolution; 82.5° horizontal FoV and 29.7° vertical FoV); while our target domain consist of 30,000 images from the Mapillary [45] crowd-sourced street-level imagery (2048×300 image resolution; 360° × 52.7° FoV). These latter images are cropped vertically from 180° down to 52.7° which is more suitable for automotive problems. This reduced panorama has an angular coverage 7.7 times larger than our source KITTI imagery. Due to the lack of annotated labels for our target domain, we adapt the source domain dataset to train deep architectures for panoramic imagery via a methodology based on projection and style transformations.

Due to dataset bias [58], training on the original source domain is unlikely to perform well on the target domain. Furthermore our target is relatively low resolution and has numerous compression artefacts not present in the source domain – present due to the practicality of 360° image transmission and storage. To improve generalization to the target domain, we transform the source domain to look similar to imagery from our target domain via a two-step process.

The first step transfers the style of our target domain (reprojected as rectilinear images) onto each image from the source domain (Fig. 2A); resulting images are shown in Fig. 2B. We use the work of Zhu *et al*. on CycleGAN [65] to learn a transformation back and forth between the two unpaired domains. Subsequently, this transformation model is used to transfer the style of our target domain onto all the images from our source domain. In essence, the style transfer introduces a tone mapping and imitates compression artifacts present in most panoramic images while preserving the actual geometry. Without the use of style transfer, the weights are biased toward high-quality imagery and perform poorly on low-quality images.

The second step reprojects the style-transferred images (Fig. 2B) and annotations from the source domain rectilinear projection to an equirectangular projection (Fig. 2D). The transformed images represent small subregions (FoV: 82.5° ×29.7°) of a larger panorama. While this set of transformed images does

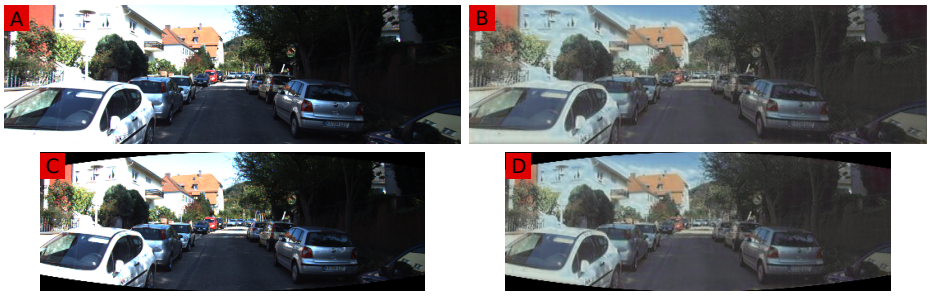


Fig. 2: Output of each step of the adaptation of an image from the KITTI dataset: A: No tranformation, B: Style transfer, C: Projection transfer, D: Style and projection

not cover the full panorama, we find that they are sufficient to train deep architectures that perform well on full size panoramic imagery.

3.3 3D Object Detection

For 3D detection, we use a network by Cai *et al.* [6] based on Faster R-CNN [51]. This network generates a sequence of detection proposals using a Region Proposal Network (RPN) and then pools a subregion around each proposal to further regress the proposal 2D location. We extend this network to support 3D object pose regression. Uniquely, our extended network can be used on either rectilinear or equirectangular imagery without any changes to the network itself, instead only requiring a change to the interpretation of the output for subsequent rectilinear or equirectangular imagery use.

While Mousavian *et al.* [47] shows that 3D pose can be estimated without any assumptions of known 3D templates, their algorithm relies on geometrical properties. In contrast, we regress the 3D pose directly, simplifying the computation and making it easier to adapt to equirectangular images.

Here, we directly regress the 3D dimensions (width, length and height) in meters of each object using a fully-connected layer as well as the orientation as per [47]. Moreover, instead of relying on geometrical assumptions, we also regress the object disparity $d_{lin} = \frac{r}{f}$ which is the inverse of the distance r multiplied by the focal length f. For equirectangular imagery, we use a similar definition $d_{equi} = \frac{r}{\alpha}$ substituting the angular resolution for the focal length. Using a fully-connected layer connected to the last common layer defined in [6], we learn coefficients a, b such that the disparity d can be expressed as:

$$d = ah_{roi} + b \tag{12}$$

where h_{roi} is the height of the region proposal generated by the RPN. To simplify the computation, we also learn the 2D projection of the centre of the object onto the image (u, v) using another fully-connected layer. As a result, we can recover the actual 3D position (x, y, z) using:

$$[x, y, z]^T = r \cdot u[P^{-1} \cdot [u_{lin}, v_{lin}, 1]^T] \qquad \textit{rectilinear case} \tag{13}$$

$$[x, y, z]^T = r \cdot u[\Gamma^{-1}(T_{equi}^{-1} \cdot [u_{equi}, v_{equi}, 1]^T)] \qquad \textit{equirectangular case} \tag{14}$$

where $u[v] = \frac{v}{\|v\|}$ is the unit vector in the direction of v.

For network training of our model, we additionally use data augmentation including image cropping and resizing as defined by [6]. Any of those operations on the image must be accompanied by the corresponding transformation of the corresponding camera matrix P or T_{equi} in order to facilitate effective training.

As noted by Mousavian *et al.* [47], distant objects (far) pose a significant challenge for reliable detection of the absolute orientation (*i.e.* relative front to back directional pose). Confronted with such an ambiguity (absolute directional orientation), a naive regression using the mean-square error would choose the

average of the two extrema rather than the most likely extremum. To circumvent this problem, given the object yaw θ (orientation on the ground plane), we instead learn $c = \cos^2\theta$ an $s = \sin^2\theta$ which are both independent of the directionality. Noting that $\cos^2\theta + \sin^2\theta = 1$, c and s can be very conveniently learned with a fully-connected layer followed by a *Softmax()* layer. For each pair (s, c), there are four possible angles each in a different quadrant depending on the sign of the sine and cosine:

$$\hat{\theta} = \text{atan2}(\pm\sqrt{s}, \pm\sqrt{c}) \tag{15}$$

We further discriminate between the four quadrants:

$$\{(-1,-1),(-1,1),(1,-1),(1,1)\} \tag{16}$$

using a separate classifier consisting of a fully-connected layer followed by a *Softmax()* classification layer.

Our entire network, comprising the architecture of [6] and our 3D pose regression extension, is fine-tuned end-to-end using a multi-task loss over 6 sets of heterogeneous network outputs: *class* and *quadrant classification* are learned via cross entropy loss while *bounding-box position*, *object centre*, *distance*, *orientation* are dependent on a mean-square loss. As a result, it would be time-consuming to manually tune the multi-task loss weights, therefore we use the methodology of [37] to dynamically adjust the multi-task weights during training based on homoscedastic uncertainty without any use of manual hyperparameters.

3.4 Monocular Depth Recovery

We rely on the approach of Godard *et al.* [26] which was originally trained and tested on the rectilinear stereo imagery of the KITTI dataset [19]. We reuse the same architecture and retrain it on our domain-adapted KITTI dataset constructed using the methodology of Sec. 3.2.

Following the original work [26], the loss function is based on a left-right consistency check between a pair of stereo images. In our new dataset, both stereo images have been warped to an equirectangular projection as well as depth smoothness constraints. While Godard *et al.* uses the stereo disparity $d_{stereo} = \frac{fB}{zw}$ where f is the focal length, B the stereo baseline and w the width of the image, we replace the focal length with the angular resolution: $d_{equi} = \frac{\alpha B}{rw}$.

Given a point $p_l = (u_l, v_l)^T$, the corresponding point $p_r = (u_r, v_r)^T$ for a given disparity d can be calculated as:

$$p_r = T_{equi} \cdot \Gamma \left[u \left[\Gamma^{-1}(T_{equi}^{-1} \cdot p_l) \right] + \left[\frac{d_{equi} w}{\alpha}, 0, 0 \right]^T \right] \tag{17}$$

with definitions as per Sec. 3.1. The corresponding point p_r in Eqn. 17 is differentiable w.r.t. d_{equi} and is used for the left/right consistency check instead of the original formulation presented in [26]. This alternative formulation (Eqn. 3.1) explicitly takes into account that the epipolar lines in a conventional rectilinear stereo setup are transformed to epipolar curves within panoramic imagery, hence enabling the adaptation of monocular depth prediction [26] to this case.

3.5 360° Network Adaptation

While the trained network can be used as is [34,11] without any further modification, objects overlapping the left and right extremities of the equirectangular image would be split into two objects; one on the left, and one on the right (as depicted in Fig. 3(a), bottom left). Moreover, information would not flow from one side of the image to the other side of the image — at least in the early feature detection layers. As a result, the deep architecture would "*see*" those objects as if heavily occluded. Therefore, it is more difficult to detect objects overlapping the image boundary leading to decreased overall detection accuracy and recall.

A cropped equirectangular panorama can be folded into a 360° ring shown in Fig. 3(a) by stitching the left and right edges together. A 2D convolution on this ring is equivalent to padding the left and right side of the equirectangular image with respective pixels from the right and left side as if the image was tiled (as illustrated on Fig. 3(b) for 3×3 convolutions). This horizontal ring-padding is hence used on all convolutional layers instead of the conventional zero-padding to eliminate these otherwise undesirable boundary effects.

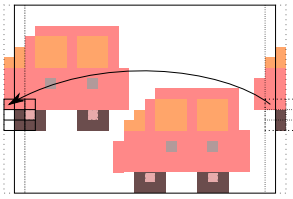
For 3D detection, our proposed approach based on Faster R-CNN [51] generates a sequence of detection proposals and subsequently pools a subregion around each proposal to further regress the final proposal location, class and 3D pose. To adapt this operation, instead of clamping subregion coordinates by the equirectangular image extremities, we instead wrap horizontally the coordinates of each pixel within the box:

$$u_{wrap} \equiv u \pmod{w} \tag{18}$$

where u is the horizontal coordinate of the pixel, u_{wrap} the wrapped horizontal coordinate within the image and w the image width.

As a result of this approach, we are hence able to hide the image boundary, as a result, enabling a true 360° processing of the equirectangular imagery.

(a) A 360° equirectangular image can be folded over itself until the ends meet.



(b) A 3×3 convolution kernel, a column of padding copied from the other side is added at each extremity

Fig. 3: Convolutions are computed seamlessly across horizontal image boundaries using our proposed padding approach.

4 Evaluation

We evaluate our approach both qualitatively on panoramic images from the crowd-sourced street-level imagery of Mapillary [45] as well as quantitatively using synthetic data generated using the CARLA [13] automotive environment simulator[2].

4.1 Qualitative Evaluation

As discussed in Sec. 2.4, we qualitatively evaluate our method using 30,000 panoramic images (Miami, USA) from the crowd-sourced street-level imagery of Mapillary [45]. Fig. 4 shows our depth recovery and 3D object detection results on a selection of images of representative scenes from the data. Ring-padding naturally enforces continuity across the right/left boundary; for instance, zero-padding can prevent detection of vehicles crossing the image boundary (Fig 5A) whereas ring-padding seamlessly detects such vehicle (Fig. 5C). Similarly zero-padding introduces depth discontinuities on the boundary (Fig 5B) whereas ring-padding enforces depth continuity (Fig. 5D).

The algorithm is able to successfully estimate the 3D pose of vehicles and recover scene depth. However the approach fails on vehicles which are too close to the camera, almost underneath the camera. Indeed, those view angles from above are not available in the narrow vertical field of view of the KITTI benchmark. Following the conventions of the KITTI dataset, any vehicles less than 25 pixels in image height were ignored during training. Due to the lower resolutions of the panoramic images, an average-size vehicle (about $2m$ height) with an apparent height of 25 pixels in KITTI is approximately at a distance of $56.6m$, whereas the same vehicle in a panoramic image will stand at $26m$. As a result, the range of the algorithm is reduced even though this is not a fundamental limitation of the approach itself. Rather, we expect this maximum distance to be increased as the resolution of the panoramic imagery is increased.

Further results are available in the supplementary video[2].

4.2 Quantitative Evaluation Methodology

Due to the lack of available annotated automotive panoramic imagery dataset, we evaluate our algorithm on synthetic data generated using the CARLA automotive environment simulator [13] adapted for panoramic imagery rendering using the same format as our qualitative dataset. Due to lack of variety, our dataset based on CARLA is not suitable for training purposes, while it is suitable for cross-dataset validation. Following KITTI conventions, we filtered out vehicles less than 25 pixels in height from our detection results.

Table 1 shows the *mean average precison* (mAP) using an *intersection over union* (IoU) of 0.5 across variations of our algorithm on 8,000 images. Overall, the projection transformation during training impairs the results by about

[2] for future comparison our code, models and evaluation data is publicly available at: https://gdlg.github.io/panoramic

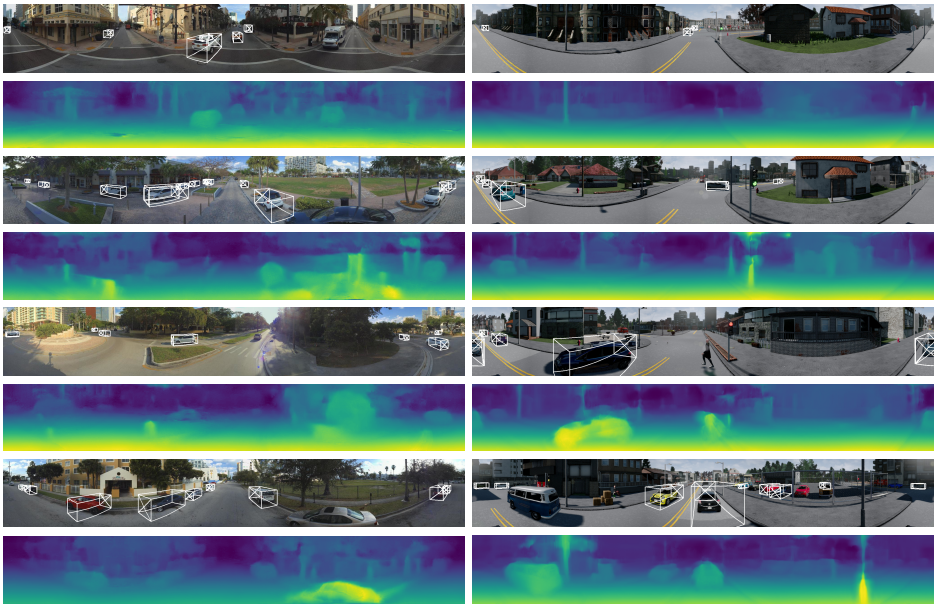


Fig. 4: Monocular depth recovery and 3D object detection with our approach. Left: Real-world images. Right: Synthetic images.

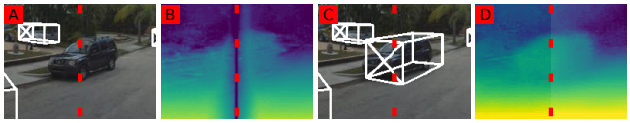


Fig. 5: Right/left boundary effect. A,B: Zero-padding; C,D: Ring-padding.

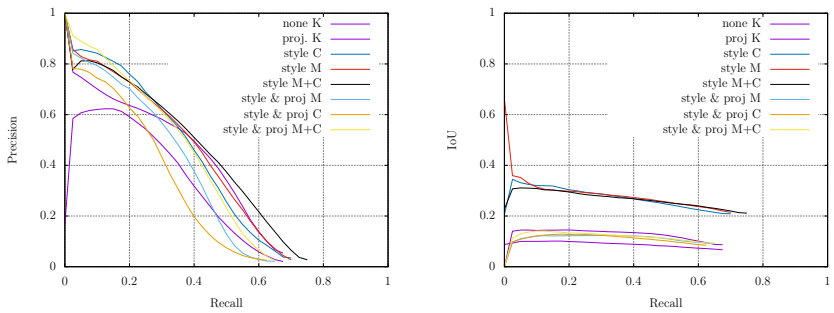
10% points. Our best results come from the combined style-transferred training dataset consisting of both Mapillary and CARLA (4% points increased compared to original) whilst training on the CARLA-adapted dataset alone increases the performance by 2% points. This is due to the simplistic rendering and lack of variety of the synthetic dataset which impairs the style transfer. As a result, the CARLA-adapted dataset significantly boosts the accuracy for very low recall; however, it also reduces the recall ability of the network (Fig. 6(a)). The model trained on the CARLA-adapted dataset achieves a mAP of 0.82 on our evaluation set of the adapted images but only 0.35 on the actual CARLA dataset which shows that the style transfer is somewhat limited. Qualitatively, style transfer toward the Mapillary dataset, which is of similar scene complexity to KITTI, is significantly better than CARLA. By contrast, the combined dataset is able to outperform on both metrics (Fig. 6(a)).

The monocular depth estimation results are shown in Table 1 for 200 images (for distances $< 50m$). Similar to our detection result, using CARLA-adapted imagery impairs the performance. Using projection transformation, we see an increase of about 2.5% points in accuracy. Overall, those differences are smaller than those on object detection across the different transformations (Table 1).

Transformation	Dataset	Detection[a] mAP	Depth Error Metrics[b] Abs. rel.	Sq. rel.	RMSE	RMSE log	Depth Acc.[a] $\delta < 1.25$
none	K	0.336	0.247	7.652	3.484	0.465	0.697
proj.	K	0.244	0.251	7.381	3.451	0.445	0.732
style	C	0.355	0.262	7.668	3.601	0.480	0.686
style	M	0.359	0.257	7.937	3.634	0.474	0.682
style	M+C	0.378	0.230	6.338	3.619	0.474	0.679
style & proj.	C	0.259	0.292	9.649	3.660	0.469	0.723
style & proj.	M	0.308	0.300	10.467	3.798	0.473	0.719
style & proj.	M+C	0.344	0.231	6.377	3.598	0.463	0.716

a Higher, better b Lower, better

Table 1: 3D Object detection (mAP) results; and depth recovery results using metrics defined by [15]. Training dataset: C: CARLA, M: Mapillary, K: KITTI



(a) Object detection results (b) 3D Intersection over Union

Fig. 6: Object detection results

From our results, we can clearly see that we have identified a new and challenging problem within the automotive visual sensing space (Table 1) when compared to the rectilinear performance of contemporary benchmarks [19,18].

5 Conclusion

We have adapted existing deep architectures and training datasets, proven on forward-facing rectilinear camera imagery, to perform on panoramic images. The approach is based on domain adaptation using geometrical and style transforms and novel updates to training loss to accommodate panoramic imagery. Our approach is able to recover the monocular depth and the full 3D pose of vehicles.

We have identified panoramic imagery has a new set of challenging problems in automotive visual sensing and provide the first performance benchmark for the use of these techniques on 360° panoramic imagery, with a supporting dataset, hence acting as a key driver for future research on this topic.

References

1. Atapour-Abarghouei, A., Breckon, T.P.: Real-Time Monocular Depth Estimation using Synthetic Data with Domain Adaptation. In: Proc. Computer Vision and Pattern Recognition. IEEE (2018) 1, 4, 5
2. Austrian Institute of Technology: WildDash Benchmark. http://www.wilddash.cc/ (2018) 1
3. Badrinarayanan, V., Kendall, A., Cipolla, R.: SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(12), 2481–2495 (Nov 2015). https://doi.org/10.1109/TPAMI.2016.2644615 1
4. Brostow, G.J., Fauqueur, J., Cipolla, R.: Semantic object classes in video: A high-definition ground truth database. Pattern Recognition Letters **30**, 88–97 (2009). https://doi.org/10.1016/j.patrec.2008.04.005 1
5. Brown, M., Szeliski, R., Winder, S.: Multi-image matching using multi-scale oriented patches. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol. 1, pp. 510–517 vol. 1 (Jun 2005). https://doi.org/10.1109/CVPR.2005.235 2
6. Cai, Z., Fan, Q., Feris, R.S., Vasconcelos, N.: A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection. In: Computer Vision – ECCV 2016. pp. 354–370. Springer International Publishing (2016) 2, 3, 4, 5, 9, 10
7. Chabot, F., Chaouch, M., Rabarisoa, J., Teulière, C., Chateau, T.: Deep MANTA: A Coarse-to-fine Many-Task Network for joint 2D and 3D vehicle analysis from monocular image. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR. pp. 1827–1836 (2017). https://doi.org/10.1109/CVPR.2017.198 4
8. Chen, X., Kundu, K., Zhang, Z., Ma, H., Fidler, S., Urtasun, R.: Monocular 3D Object Detection for Autonomous Driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2147–2156 (2016) 4
9. Chen, X., Kundu, K., Zhu, Y., Berneshawi, A.G., Ma, H., Fidler, S., Urtasun, R.: 3D Object Proposals for Accurate Object Class Detection. In: Advances in Neural Information Processing Systems. pp. 424–432 (2015) 4
10. Chen, X., Ma, H., Wan, J., Li, B., Xia, T.: Multi-View 3D Object Detection Network for Autonomous Driving. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 6526–6534 (2017). https://doi.org/10.1109/CVPR.2017.691 2, 4
11. Deng, F., Zhu, X., Ren, J.: Object detection on panoramic images based on deep learning. In: 2017 3rd International Conference on Control, Automation and Robotics (ICCAR). pp. 375–380 (Apr 2017). https://doi.org/10.1109/ICCAR.2017.7942721 3, 11
12. Donahue, J., Krähenbühl, P., Darrell, T.: Adversarial Feature Learning. CoRR **abs/1605.09782** (2016) 5
13. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An Open Urban Driving Simulator. CoRL (2017) 3, 5, 12
14. Dumoulin, V., Shlens, J., Kudlur, M.: A Learned Representation For Artistic Style. arXiv:1610.07629 [cs] (Oct 2016) 5
15. Eigen, D., Puhrsch, C., Fergus, R.: Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2. pp. 2366–2374. NIPS'14, MIT Press (2014) 4, 14

16. Fisher, Y.: Berkeley Data Drive. http://data-bdd.berkeley.edu/ (2018) 1
17. Gatys, L.A., Ecker, A.S., Bethge, M.: Image Style Transfer Using Convolutional Neural Networks. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition. pp. 2414–2423 (Jun 2016). https://doi.org/10.1109/CVPR.2016.265 5
18. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The KITTI dataset. The International Journal of Robotics Research **32**(11), 1231–1237 (Sep 2013). https://doi.org/10.1177/0278364913491297 1, 2, 5, 8, 14
19. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The KITTI vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 3354–3361 (Jun 2012). https://doi.org/10.1109/CVPR.2012.6248074 1, 2, 5, 8, 10, 14
20. Ghiasi, G., Lee, H., Kudlur, M., Dumoulin, V., Shlens, J.: Exploring the structure of a real-time, arbitrary neural artistic stylization network. BMVC (2017) 5
21. Ghifary, M., Kleijn, W.B., Zhang, M., Balduzzi, D., Li, W.: Deep Reconstruction-Classification Networks for Unsupervised Domain Adaptation. In: Computer Vision – ECCV 2016. pp. 597–613. Lecture Notes in Computer Science, Springer, Cham (Oct 2016) 5
22. Girshick, R.: Fast R-CNN. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 1440–1448 (Dec 2015). https://doi.org/10.1109/ICCV.2015.169 4
23. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 580–587 (Jun 2014). https://doi.org/10.1109/CVPR.2014.81 4
24. Girshick, R.: Fast R-CNN. In: 2015 IEEE International Conference on Computer Vision (ICCV) (2015) 3
25. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. pp. 580–587 (2014) 3
26. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised Monocular Depth Estimation with Left-Right Consistency. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6602–6611 (2017). https://doi.org/10.1109/CVPR.2017.699 1, 2, 5, 10
27. González, A., Vázquez, D., López, A.M., Amores, J.: On-Board Object Detection: Multicue, Multimodal, and Multiview Random Forest of Local Experts. IEEE Transactions on Cybernetics **47**(11), 3980–3990 (Nov 2017). https://doi.org/10.1109/TCYB.2016.2593940 2
28. Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., Schölkopf, B.: Covariate Shift by Kernel Mean Matching (2008). https://doi.org/10.1.1.165.8929 5
29. Hamilton, O.K., Breckon, T.P.: Generalized dynamic object removal for dense stereo vision based scene mapping using synthesised optical flow. In: 2016 IEEE International Conference on Image Processing (ICIP). pp. 3439–3443 (Sep 2016). https://doi.org/10.1109/ICIP.2016.7532998 2
30. Häne, C., Heng, L., Lee, G.H., Sizov, A., Pollefeys, M.: Real-Time Direct Dense Matching on Fisheye Images Using Plane-Sweeping Stereo. In: 2014 2nd International Conference on 3D Vision. vol. 1, pp. 57–64 (Dec 2014). https://doi.org/10.1109/3DV.2014.77 2
31. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 2980–2988. IEEE (2017) 3

32. Hoffman, J., Tzeng, E., Park, T., Zhu, J.Y., Isola, P., Saenko, K., Efros, A.A., Darrell, T.: CyCADA: Cycle-Consistent Adversarial Domain Adaptation. arXiv:1711.03213 [cs] (Nov 2017) 5
33. Hosang, J., Benenson, R., Dollár, P., Schiele, B.: What makes for effective detection proposals? IEEE Transactions on Pattern Analysis and Machine Intelligence **38**(4), 814–830 (2016) 3
34. Hu, H.N., Lin, Y.C., Liu, M.Y., Cheng, H.T., Chang, Y.J., Sun, M.: Deep 360 Pilot: Learning a Deep Agent for Piloting through 360° Sports Video. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1396–1405 (May 2017). https://doi.org/10.1109/CVPR.2017.153 2, 3, 11
35. Janai, J., Güney, F., Behl, A., Geiger, A.: Computer Vision for Autonomous Vehicles: Problems, Datasets and State-of-the-Art. arXiv:1704.05519 [cs] (Apr 2017) 2
36. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual Losses for Real-Time Style Transfer and Super-Resolution. arXiv:1603.08155 [cs] (Mar 2016) 5
37. Kendall, A., Gal, Y., Cipolla, R.: Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. arXiv:1705.07115 [cs] (May 2017) 10
38. Kendall, A., Martirosyan, H., Dasgupta, S., Henry, P., Kennedy, R., Bachrach, A., Bry, A.: End-to-End Learning of Geometry and Context for Deep Stereo Regression. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 66–75 (2017) 1
39. Kondermann, D., Nair, R., Honauer, K., Krispin, K., Andrulis, J., Brock, A., Güssefeld, B., Rahimimoghaddam, M., Hofmann, S., Brenner, C., Jähne, B.: The HCI Benchmark Suite: Stereo and Flow Ground Truth with Uncertainties for Urban Autonomous Driving. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 19–28 (Jun 2016). https://doi.org/10.1109/CVPRW.2016.10 1
40. Kuznietsov, Y., Stückler, J., Leibe, B.: Semi-supervised deep learning for monocular depth map prediction. In: Proc. Conf. Computer Vision and Pattern Recognition. pp. 6647–6655 (2017) 5
41. Ladický, L., Shi, J., Pollefeys, M.: Pulling Things out of Perspective. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 89–96 (Jun 2014). https://doi.org/10.1109/CVPR.2014.19 1, 4
42. Lai, W.S., Huang, Y., Joshi, N., Buehler, C., Yang, M.H., Kang, S.B.: Semantic-driven Generation of Hyperlapse from 360° Video. IEEE Transactions on Visualization and Computer Graphics (2018). https://doi.org/10.1109/TVCG.2017.2750671 2, 3
43. Liu, F., Shen, C., Lin, G., Reid, I.: Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields. IEEE Transactions on Pattern Analysis and Machine Intelligence **38**(10), 2024–2039 (Oct 2016). https://doi.org/10.1109/TPAMI.2015.2505283 4
44. Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning Transferable Features with Deep Adaptation Networks. arXiv:1502.02791 [cs] (Feb 2015) 5
45. Mapillary: Mapillary Research. https://research.mapillary.com/ 3, 5, 8, 12
46. Matzen, K., Cohen, M.F., Evans, B., Kopf, J., Szeliski, R.: Low-cost 360 Stereo Photography and Video Capture. ACM Trans. Graph. **36**(4), 148:1–148:12 (Jul 2017). https://doi.org/10.1145/3072959.3073645 2
47. Mousavian, A., Anguelov, D., Flynn, J., Kosecka, J.: 3D Bounding Box Estimation Using Deep Learning and Geometry. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5632–5640 (Dec 2016). https://doi.org/10.1109/CVPR.2017.597 4, 5, 9

48. Neuhold, G., Ollmann, T., Bulò, S.R., Kontschieder, P.: The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 5000–5009 (Oct 2017). https://doi.org/10.1109/ICCV.2017.534 1
49. Pang, J., Sun, W., Ren, J.S., Yang, C., Yan, Q.: Cascade Residual Learning: A Two-stage Convolutional Neural Network for Stereo Matching. In: 2017 IEEE International Conference on Computer Vision Workshops (ICCVW). pp. 878–886 (2017). https://doi.org/10.1109/ICCVW.2017.108 1
50. Ren, J., Chen, X., Liu, J., Sun, W., Pang, J., Yan, Q., Tai, Y.W., Xu, L.: Accurate Single Stage Detector Using Recurrent Rolling Convolution. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 752–760 (2017). https://doi.org/10.1109/CVPR.2017.87 1
51. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems 28, pp. 91–99. Curran Associates, Inc. (2015) 1, 3, 4, 9, 11
52. Saxena, A., Chung, S.H., Ng, A.Y.: Learning depth from single monocular images. In: NIPS. pp. 1161–1168 (2006) 4
53. Saxena, A., Sun, M., Ng, A.Y.: Make3d: Learning 3d scene structure from a single still image. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(5), 824–840 (2009) 4
54. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision **47**(1), 7–42 (2002). https://doi.org/10.1109/SMBV.2001.988771 4
55. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229 (2013) 3
56. Su, Y.C., Grauman, K.: Flat2Sphere: Learning Spherical Convolution for Fast Features from 360° Imagery. arXiv:1708.00919 [cs] (Aug 2017) 3
57. Su, Y.C., Jayaraman, D., Grauman, K.: Pano2Vid: Automatic Cinematography for Watching 360° Videos. In: ACCV (2016) 2
58. Torralba, A., Efros, A.A.: Unbiased look at dataset bias. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1521–1528 (Jun 2011). https://doi.org/10.1109/CVPR.2011.5995347 5, 8
59. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial Discriminative Domain Adaptation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2962–2971 (Jul 2017). https://doi.org/10.1109/CVPR.2017.316 5
60. Ulyanov, D., Lebedev, V., Vedaldi, A., Lempitsky, V.: Texture Networks: Feed-forward Synthesis of Textures and Stylized Images. arXiv:1603.03417 [cs] (Mar 2016) 5
61. Xu, J., Stenger, B., Kerola, T., Tung, T.: Pano2CAD: Room Layout from a Single Panorama Image. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 354–362 (Mar 2017). https://doi.org/10.1109/WACV.2017.46 2, 3
62. Yang, F., Choi, W., Lin, Y.: Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifier. In: IEEE International Conference on Computer Vision and Pattern Recognition (2016) 4
63. Zhang, Y., Song, S., Tan, P., Xiao, J.: PanoContext: A Whole-Room 3D Context Model for Panoramic Scene Understanding. In: Computer Vision – ECCV 2014. pp. 668–686. Lecture Notes in Computer Science, Springer, Cham (Sep 2014) 2, 3

64. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: Proc. Conf. Computer Vision and Pattern Recognition. pp. 6612–6619 (2017) 4
65. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 2242–2251 (Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial). https://doi.org/10.1109/ICCV.2017.244 5, 8