A Lightweight Neural Network for Monocular View Generation with Occlusion Handling

Simon Evain
Supervision: Christine Guillemot

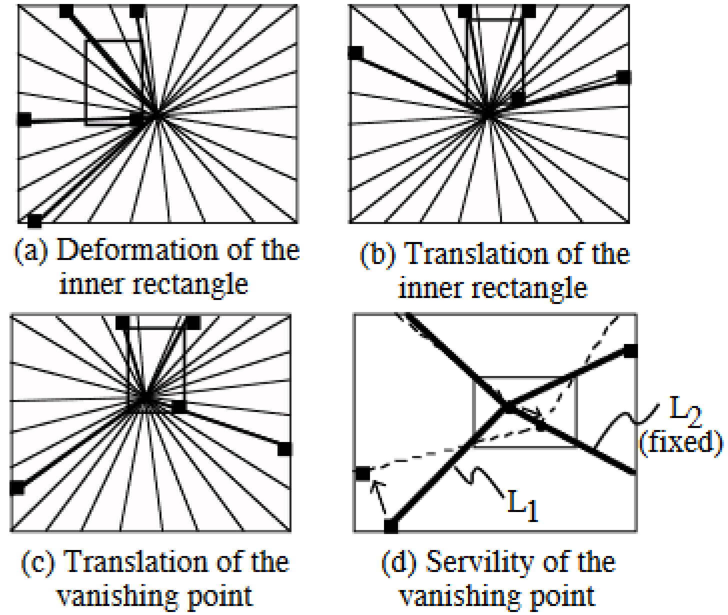
SIROCCO team
Inria Rennes – Bretagne Atlantique

Monocular View Synthesis

- *Monocular view synthesis*: Generating new viewpoints from one single image.
- In our case, we want our image to be totally unannotated, and the disparity ranges to be significant
- Tricky and mathematically ill-posed problem.

Monocular View Synthesis

- Requires definition of mathematical priors
- Handcrafted priors (*Tour into the Picture,* Horry et al., 1997 ; *Automatic photo pop-up*, Hoeim et al., 2005)
- Emergence of learning-based methods leads to more efficient, data-driven priors.



(a) Deformation of the inner rectangle

(b) Translation of the inner rectangle

(c) Translation of the vanishing point

(d) Servility of the vanishing point

Deep Learning for View Synthesis

- A straightforward and naive DL-based approach can not really return good results in our setting.

- Performing a direct pixelwise minimization risks leading to blurry results.

- There is no perfect correlation between the PSNR and the visual quality, we will tend to favor prediction #2, while a pixelwise-trained network will tend to favor prediction #1.

Target image

Prediction #1

Prediction #2

Our work

- Architecture able to perform view synthesis from one single image. During training, we use **stereo pairs of images**.

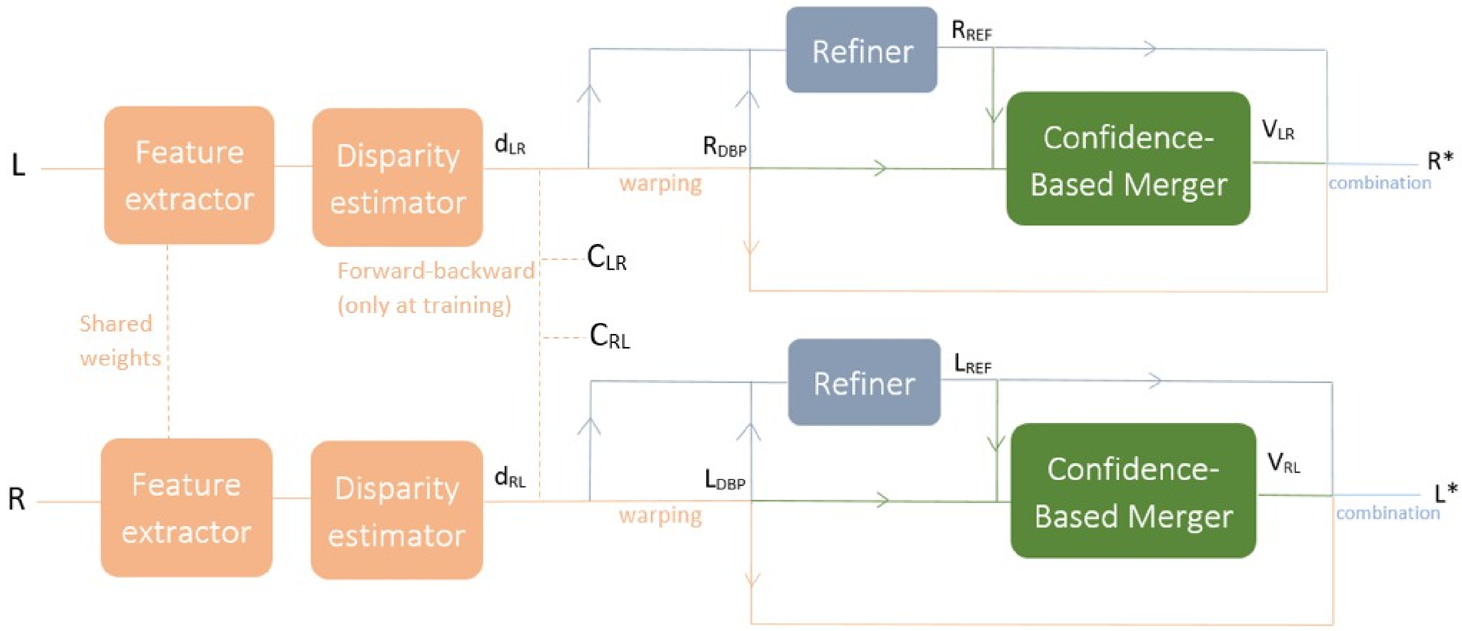
- Predicting a **disparity map** for the scene

- Estimating a **pixelwise confidence map** in the final prediction.

- A lightweight approach, with a relatively small number of parameters (~6,5M) when compared with reference methods.
- Scalable, can be applied to images on various resolutions with convincing performance.
- Short training time (~ a few hours)

Overall structure

- 3 components:
  - DBP (Disparity-Based Predictor): **Estimates the depth of the scene.**
  - CBM (Confidence-Based Merger): Identifies the **occluded and non-Lambertian regions** through **forward-backward consistency** in disparity maps.
  - REF (Refiner): Refines the prediction in ‘tricky’ regions using a new neural network.



Disparity-Based Predictor (DBP)

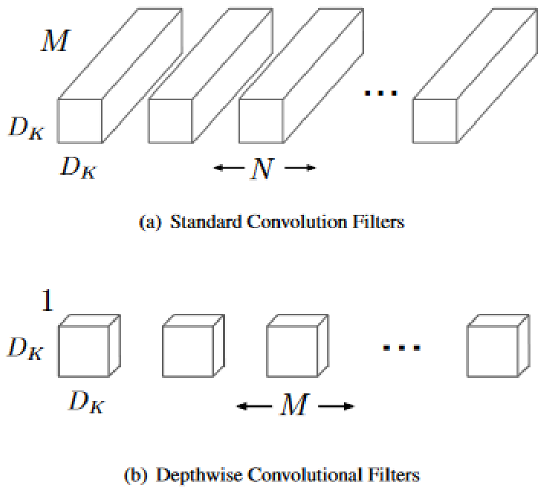
- Idea: predicting the disparity map from one single image, through unsupervised learning.

- Our disparity predictor is made up of a CNN followed by a **non-learnable Spatial Transformer Layer** (which computes a **specific geometrical transformation**, here warping). The learning metrics is based on the image, and set at the output of the spatial transformer layer. Disparity is thus learnt in an **unsupervised** fashion.

DBP architecture

- The feature extractor is based upon the **MobileNet** architecture, with pre-trained weights on ImageNet.
- MobileNet is lighter than most feature extractors, due to its handling of convolutions.
- Since the output of the DBP must be the same resolution as the input, we add a second portion, built as **its symmetrical counterpart**.

MobileNet: Efficient convolutional neural networks for mobile vision applications, Howard et al., 2017

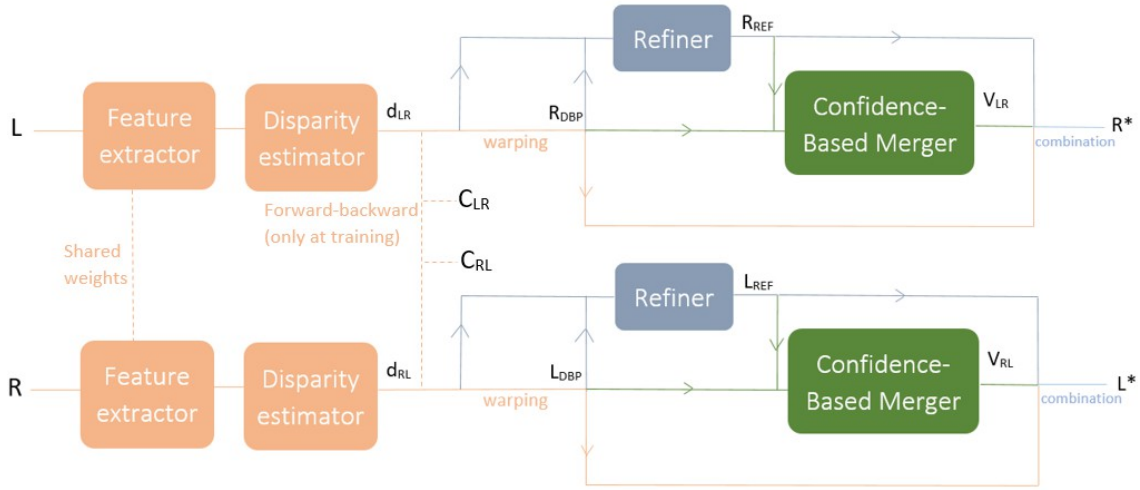


(c) 1×1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

Limitations

- **Occlusions** cannot be processed by this warping method
- It is also difficult for this method to take care of **non-Lambertian regions**
- All regions that are not matched properly contain structural artifacts.

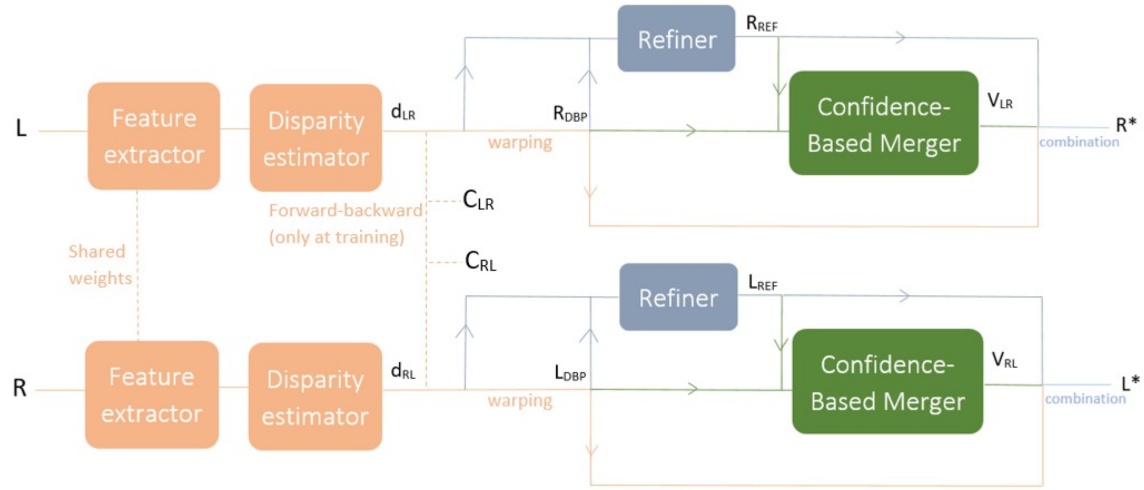
Confidence Measures



- We want to identify the regions where our Disparity-Based Predictor will be insufficient.
- To do so, we exploit the two disparity maps that were produced.
- We build then a pixelwise confidence measure based on the **forward-backward consistency** of these two disparity maps. If their confidence is high, we assume we can trust the disparity-based prediction for this pixel.

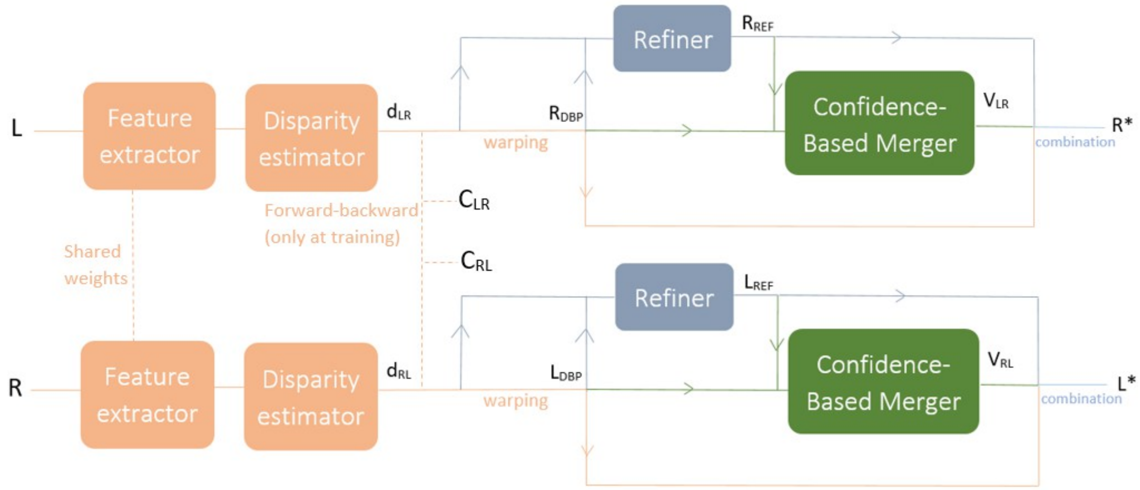
$$C_{RL}(x,y) = \exp(-\gamma|d_{RL}(x,y) - d_{LR}(x - d_{RL}(x,y),y)|)$$
$$C_{LR}(x,y) = \exp(-\gamma|d_{LR}(x,y) - d_{RL}(x + d_{LR}(x,y),y)|)$$

Confidence Measures



- Computing these confidence maps is possible at training time, for we have the two branches in a joint training. At test time, we only have one branch.
- The **Confidence-Based Merger** (CBM) is a CNN architecture of a few successive convolutional layers, which aims at estimating, during training, the value of these confidence maps. This way, at test time, we can use the estimations to identify the low-confidence regions.

Refiner



- The Refiner aims at **improving the visual quality of the occluded and non-Lambertian regions**.

- It is a CNN made up of a few successive convolutional layers.

Blending

- Our final prediction can then be written as:

$$L^* = V_{RL} L_{REF} + (1 - V_{RL}) L_{DBP}$$
$$R^* = V_{LR} R_{REF} + (1 - V_{LR}) R_{DBP}$$

Learning Schedule

- Learning schedule in 3 steps (interrelated components):

- **DBP-only:** First, only DBP is trained. We add to the pixelwise metrics a gradient-based loss.

$$\lambda_0(||L_{DBP} - L||_1 + ||R_{DBP} - R||_1) + \lambda_1(||\nabla L_{DBP} - \nabla L||_1 + ||\nabla R_{DBP} - \nabla R||_1)$$

- **Geometrical restructuring:** Then, we change the·metrics to apply a stronger structural constraint to our prediction. This allows us to correct a significant amount of the various structures within

$$\lambda_2(||\frac{2}{\max(d_{RL})}\nabla d_{RL} - \nabla L||_1 + ||\frac{2}{\max(d_{LR})}\nabla d_{LR} - \nabla R||_1) + \lambda_3(||L_{DBP} - L||_1 + ||R_{DBP} - R||_1)$$

- **Final prediction:** We finally estimate the requested confidence measures, and minimize our metrics to estimate our final prediction.

$$\lambda_4(||L_{REF} - L||_1 + ||R_{REF} - R||_1) + \lambda_5(||\nabla L_{REF} - \nabla L||_1 + ||\nabla R_{REF} - \nabla R||_1) + \lambda_6(||L^* - L||_1 + ||R^* - R||_1) + \lambda_7(||\nabla L^* - \nabla L||_1 + ||\nabla R^* - \nabla R||_1) + \lambda_8(||V_{LR} - (1 - C_{LR})||_1 + ||V_{RL} - (1 - C_{RL})||_1)$$

Implementation

- Trained on the **stereo training KITTI** dataset (automatic driving dataset). Evaluated (at first) on the **KITTI test set**.

- **256*256 patches** are randomly chosen from the **400** pair of stereo images to be used as input to the network.

- On average, the 3 learning steps allow to converge after **only a few hours**.

- The whole network is made up of about **6.5 M** parameters.

Visual results

Input image

Prediction

Ground Truth image

Visual results

Input image

Prediction

Estimated
Disparity Map

Visual results

Input image

Prediction

Estimated Confidence Map

Visual results

Input image

Prediction

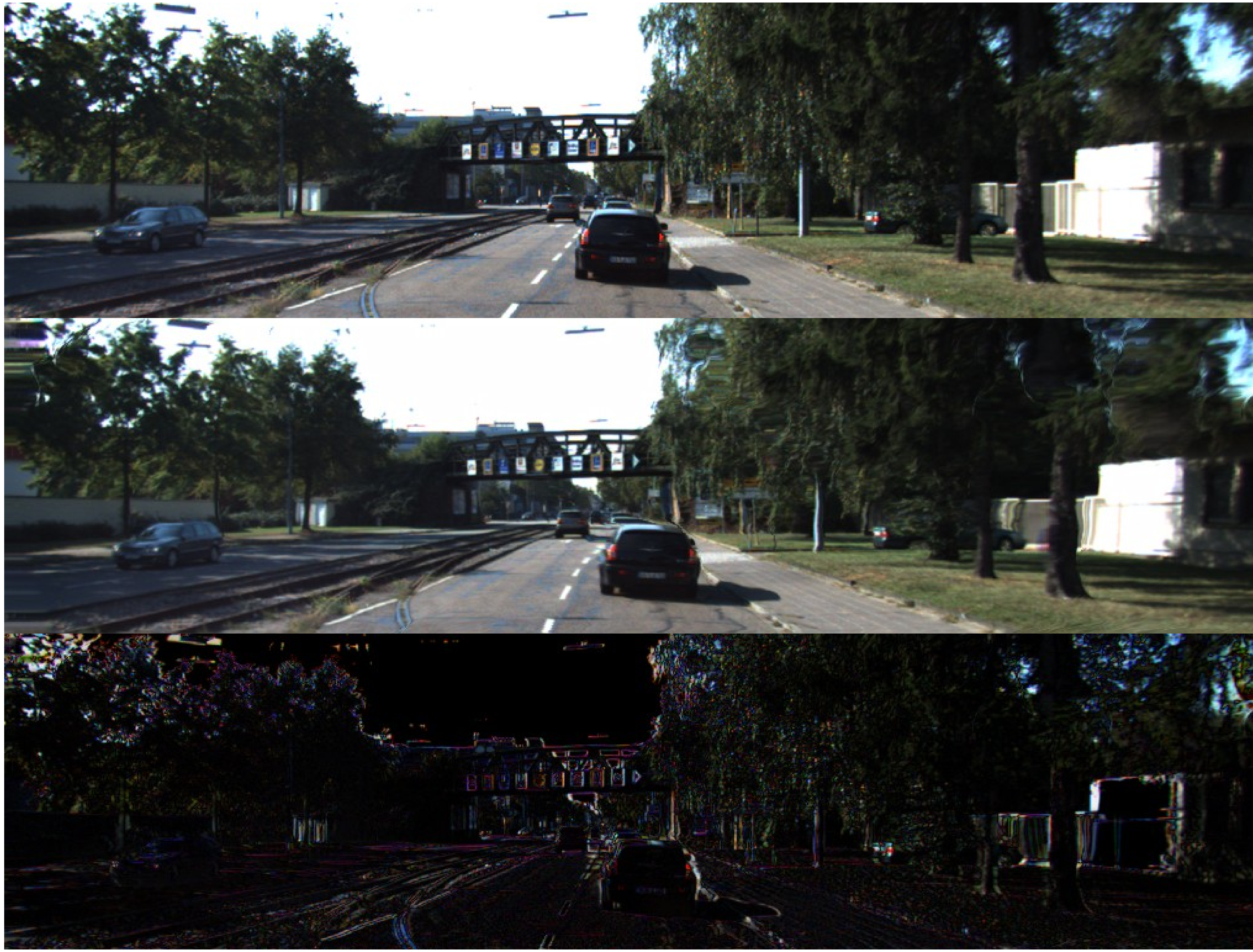
Ground Truth image

Visual results

Input image

Prediction

L1 error

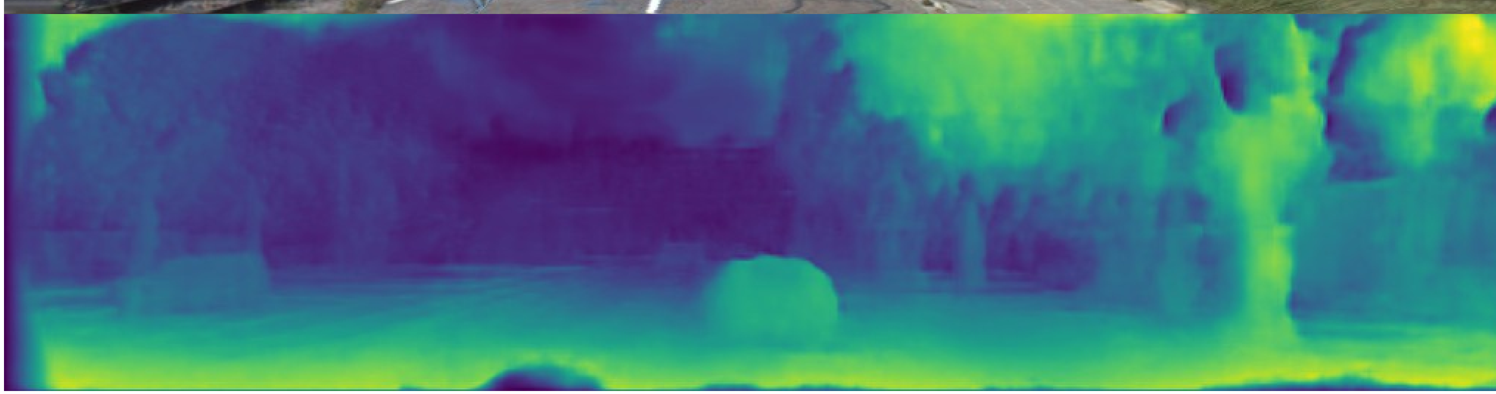


Visual results

Input image

Prediction

Estimated disparity map



Warping the estimated disparity map for multi-view generation

- To evaluate both the visual quality of our prediction and the accuracy of our depth map estimation, we can interpolate the estimated disparity maps.

Result comparison

- Comparison with reference methods (Deep3D/Godard)

- **Godard** → monocular disparity estimation method (**30M** parameters). True benefits in setting the problem as a view synthesis one (*Unsupervised Monocular Depth Estimation with Left-Right Consistency*, Godard et al., 2017)

- **Deep3D** → reference method (**61M** parameters) in 3d monocular view generation (*Deep3D: Fully Automatic 2D-to-3D Video Conversion with Deep Convolutional Neural Networks*, Xie et al., 2016)

Result comparison

Input image

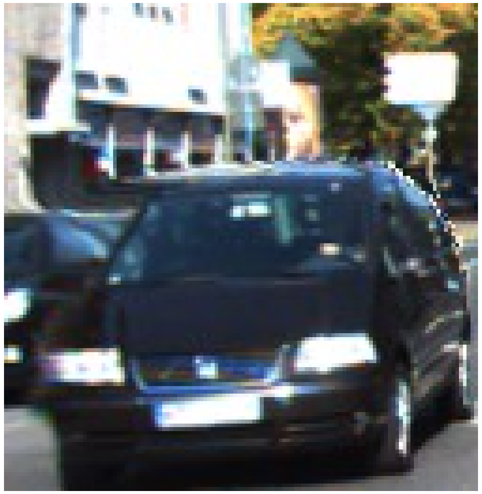
Ours

Deep3D

Godard

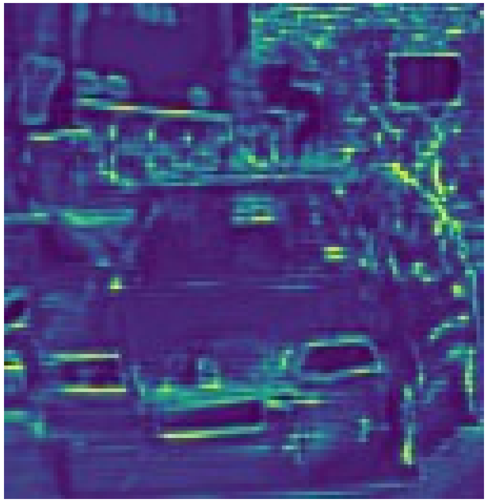
Ablation study

- Contribution of the occlusion processing component.



Disparity-Based Prediction

Final prediction



Confidence map

Ablation study

- Specific learning schedule vs end-to-end
- Confidence map vs no constraint on the blending masks

Ours

End-to-end

No constraint on the confidence map

Results on other data elements

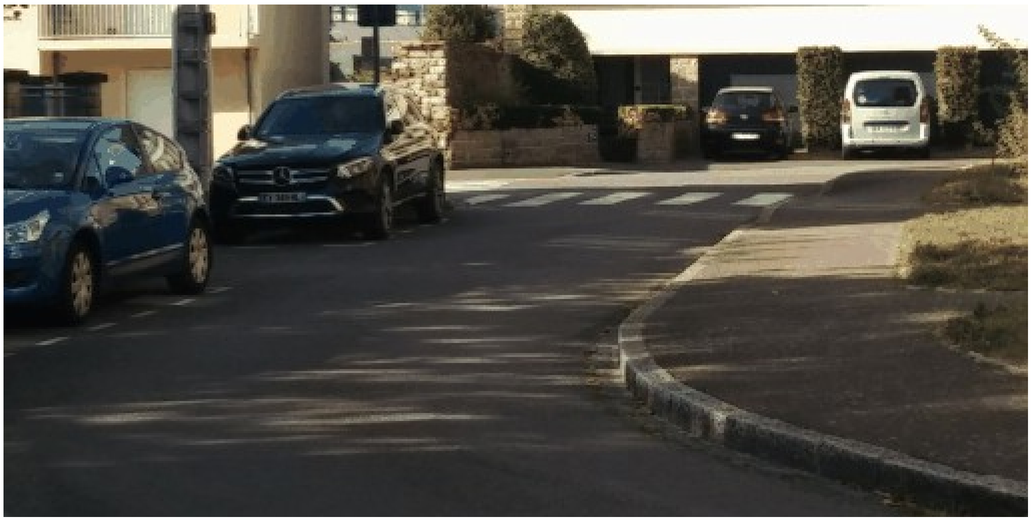
- Network trained on KITTI and applied on Cityscapes:

Input image

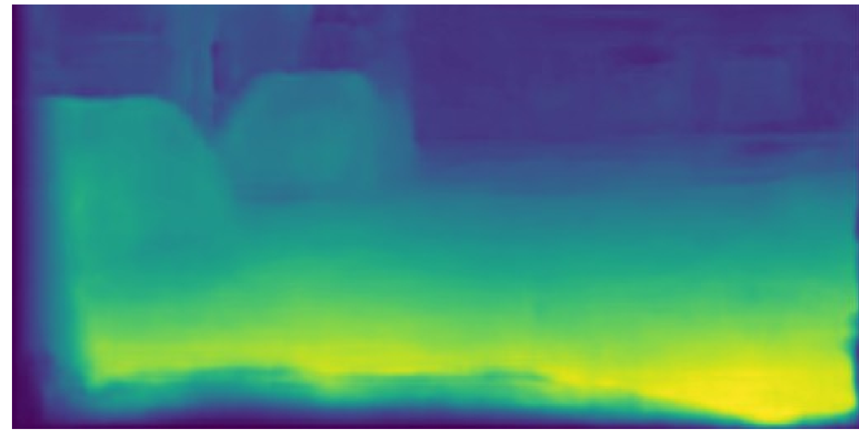
Our prediction

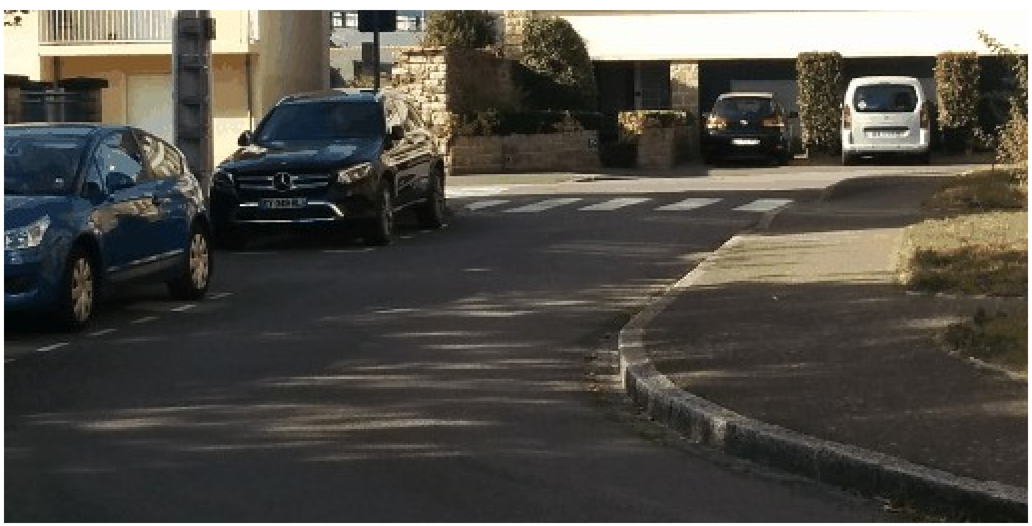
Results on other data elements

- Network trained on KITTI and applied on a picture taken in Rennes:

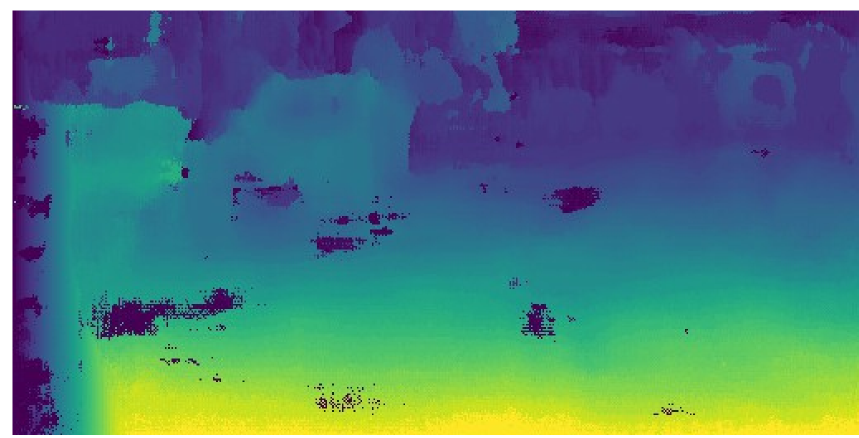


Ours





Deep3D

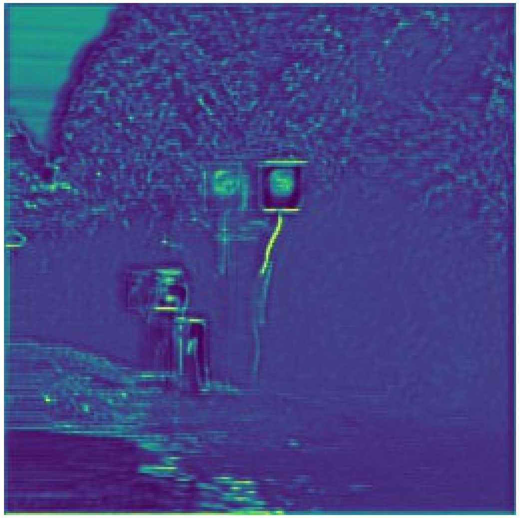


Failure cases

- Uncommon structures with strong color gradients.

Input image

Our prediction



Confidence map

Conclusion

- A lightweight architecture to perform monocular view synthesis with automatic occlusion identification and handling.

- Outperforms reference methods both statistically and visually.

References

- *Tour into the Picture,* Horry et al., Conference on Computer graphics and interactive techniques, 1997
- *Automatic photo pop-up*, Hoeim et al., ACM Transactions on Graphics, 2005
- *MobileNet: Efficient convolutional neural networks for mobile vision applications*, Howard et al., arXiv, 2017
- *Unsupervised Monocular Depth Estimation with Left-Right Consistency*, Godard et al., CVPR, 2017
- *Deep3D: Fully Automatic 2D-to-3D Video Conversion with Deep Convolutional Neural Networks*, Xie et al., ECCV, 2016

- Our article is available at: http://clim.inria.fr/research/MonocularSynthesis/pdf/article.pdf

Thanks for your attention !

Statistical comparisons

KITTI Test set	PSNR	SSIM	LPIPS	params
Ours	**19.24**	**0.74**	**0.139**	**6.5M**
Deep3D ([7])	19.08	**0.74**	0.220	61M
Godard et al. ([14])	18.44	0.71	0.148	30M

KITTI	PSNR	SSIM	LPIPS	PSNR disocc.
Phase I	18.76	0.72	0.144	14.84
Phases I-II	18.87	0.72	0.144	14.85
Phases I-II-III	19.24	0.74	**0.139**	15.32
Phases I-III	19.11	0.73	0.206	15.04
Phase III	19.23	0.74	0.345	15.35
No confidence	19.40	0.75	0.190	15.48