Blind Geometric Distortion Correction on Images Through Deep Learning
Supplementary Material

This supplementary material includes CNN training details (Section 1), discussion of model fitting method (Section 2), GeoNetS and GeoNetM (Section 3), additional results on real images (Section 4) and co-occurring distortion correction (Section 5), respectively.

1. CNN Training Details

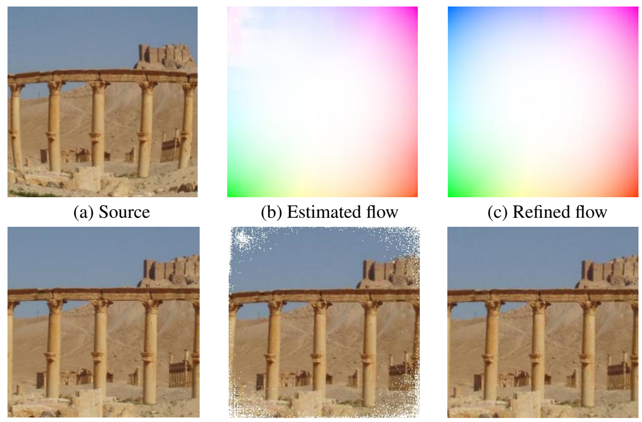
We use images from the Places365-Standard dataset [3] as original images to generate distorted images for 6 types of distortions. When synthesizing the distorted images, we randomly set the distortion parameters ρ^{β} for each type with uniform distribution in a certain range so that the synthesized dataset can cover cases with different distortion levels. There are 50,000 images for training and 5,000 images for testing for each type of distortion. Therefore, we have 300,000 images for training and 30,000 images for testing in GeoNetM. The networks are trained with the Adam optimizer [1] with a batch size of 32. The default learning rate of the Adam optimizer is 10^{-4}, and the learning curve converges when training for 6 epochs. All the distorted images in the dataset have been cropped to the size 256×256.

2. Model fitting

In this section, we provide a qualitative result to show the benefit of our model fitting method based Hough transform. Figure 1 shows that the flow becomes globally smoother, resulting in a better final image after model fitting. To test the robustness of model fitting, we tested our model fitting step on 500 samples. 74.6% of them get lower endpoint error (EPE) after fitting, 15.6% of them increase the error by less than 0.2, and only 9.8% of them increase by more than 0.2. The overall error is reduced by 0.94. Also, the fitting performance is insensitive to the bin range of the accumulator. Dividing the space into 50 to 200 bins are all feasible and get similar results in our case. These show our model fitting step is robust.

3. GeoNetS and GeoNetM

In Figure 2, we provide qualitative results of GeoNetS and GeoNetM with model fitting on the test dataset. These examples show that GeoNetS and GeoNetM with model



(a) Source (b) Estimated flow (c) Refined flow
(d) Ground truth (e) Corrected from (b) (f) Corrected from (c)

Figure 1. Comparison of the flow estimated by the network with the refined flow after model fitting. Model fitting provides a more globally smooth and accurate flow.

fitting can obtain similar accurate results. And GeoNetS achieves lower flow error than GeoNetM with model fitting since additional information needs to be learned for the multi-type task.

In the second part of GeoNetS, it downsamples the feature maps and estimates the distortion parameters by a fully connected layer, so it has no decoder with residual blocks to upsample the maps to full resolution as in GeoNetM. Also, we noticed that by incorporating the distortion model in the network, GeoNetS can achieve lower error than GeoNetM even with simpler building blocks and fewer layers. But GeoNetS are specialized for different distortions, while GeoNetM can handle all distortions within one network.

For GeoNetM, We have tried four different values (0.05, 0.1, 0.2, 0.5) for λ to train the network and found that the difference between them is small (less than 5% in EPE), which shows that GeoNetM is not too sensitive to λ, unless we set it to zero or a very high value, in which case the one branch will be dominant.

4. Additional results on real images

We also test on real images using GeoNetM with the model fitting method in Figure 3. The first three rows are

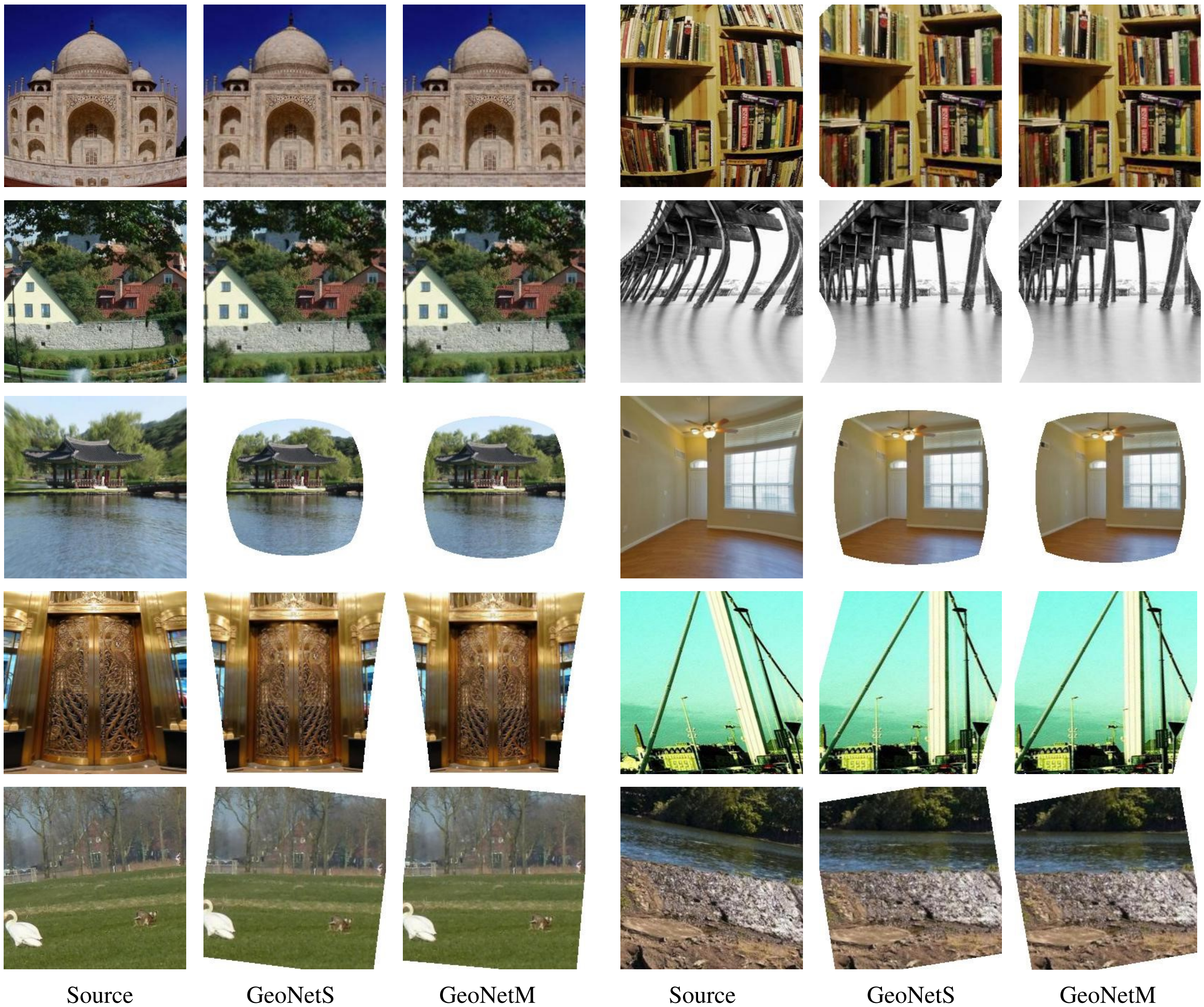


Figure 2. More results of GeoNetS and GeoNetM with model fitting.

images from a wide angle lens distortion database [2] created using a Nikon D90 Camera with a Tokina 10-17mm Lens. Different settings of the lens focal distance are used obtaining a variety of image lens distortions. The last two rows are images from the Internet.

5. Co-occurring distortion correction

Sometimes an image could have more than one type of distortion, as shown in the first column of Figure 4. These two images also come from the database [2] and both of them have lens distortion and perspective distortion simultaneously. We can correct the distorted image simply by running our correction algorithm twice iteratively. The second and third column of Figure 4 shows the corrected results. For each iteration, it detects and corrects the most severe type of distortion that it encounters.

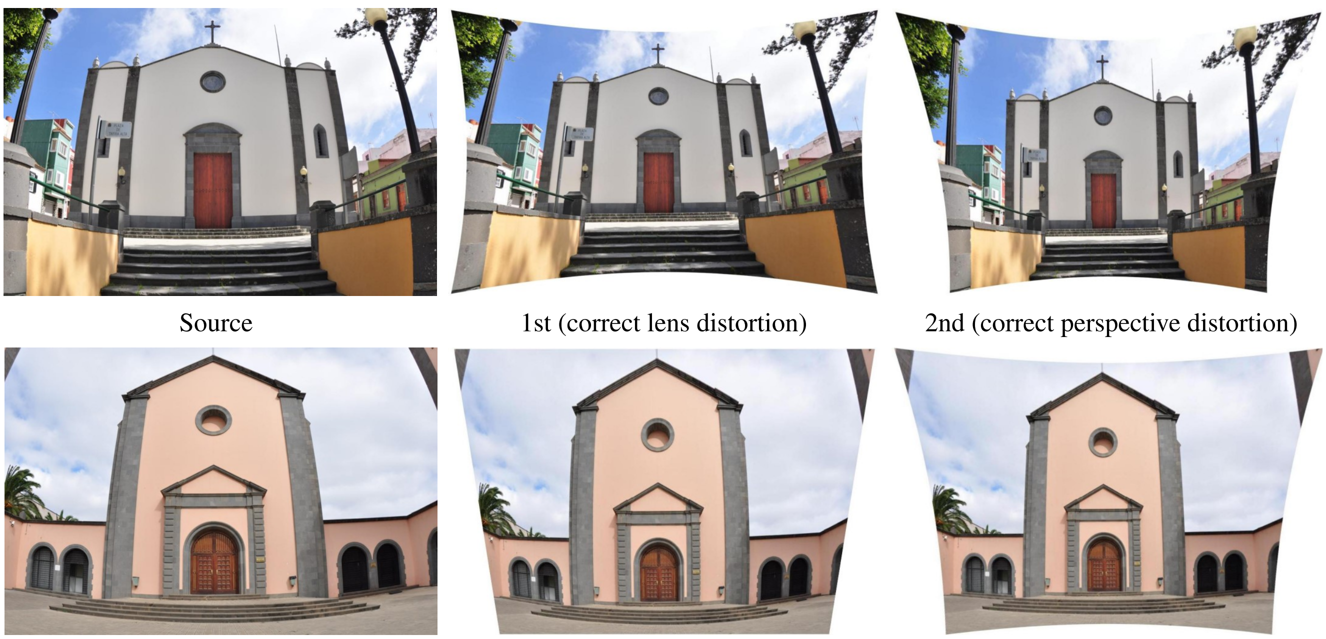
References

[1] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[2] D. Santana-Cedrés, L. Gomez, M. Alemán-Flores, A. Salgado, J. Esclarín, L. Mazorra, and L. Alvarez. Invertibility and estimation of two-parameter polynomial and division lens distortion models. *SIAM Journal on Imaging Sciences*, 8(3):1574–1606, 2015. 2

[3] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 1

Figure 3. More results on real images by GeoNetM with model fitting.



Source | 1st (correct lens distortion) | 2nd (correct perspective distortion)

Source | 1st (correct perspective distortion) | 2nd (correct lens distortion)

Figure 4. We run our correction algorithm for many times iteratively to correct an image with two or more co-occurring distortions.