Learning to See through Turbulent Water

Zhengqin Li* Zak Murez* David Kriegman Ravi Ramamoorthi Manmohan Chandraker
University of California, San Diego
{zhl378, zmurez, kriegman, ravir, mkchandraker}@cs.ucsd.edu

Abstract

Imaging through dynamic refractive media, such as looking into turbulent water, or through hot air, is challenging since light rays are bent by unknown amounts leading to complex geometric distortions. Inverting these distortions and recovering high quality images is an inherently ill-posed problem, leading previous works to require extra information such as high frame-rate video or a template image, which limits their applicability in practice. This paper proposes training a deep convolution neural network to undistort dynamic refractive effects using only a single image. The neural network is able to solve this ill-posed problem by learning image priors as well as distortion priors. Our network consists of two parts, a warping net to remove geometric distortion and a color predictor net to further refine the restoration. Adversarial loss is used to achieve better visual quality and help the network hallucinate missing and blurred information. To train our network, we collect a large training set of images distorted by a turbulent water surface. Unlike prior works on water undistortion, our method is trained end-to-end, only requires a single image and does not use a ground truth template at test time. Experiments show that by exploiting the structure of the problem, our network outperforms state-of-the-art deep image to image translation.

1. Introduction

Consider the imaging scenario in which the camera views a scene through a refractive medium, in which the interface is constantly changing. Two common examples of this occur when looking from air into water with a turbulent surface and imaging through a medium with temperature variations that gives rise to atmospheric refraction or mirages. In all such cases, the scene appears distorted due to the bending of light as it passes through the refractive interface.

Removing such distortions from a single image is challenging since the shape of the interface is not known a priori and must be estimated simultaneously with the latent image.

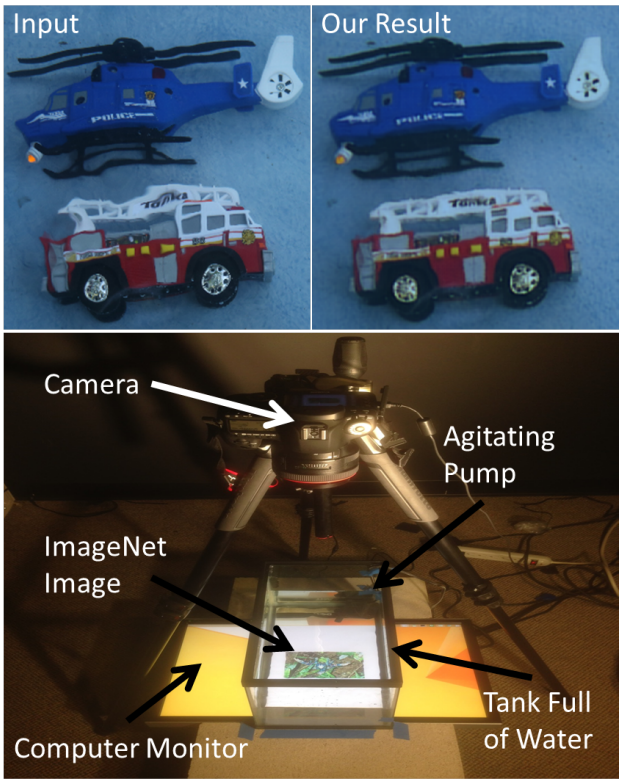

Figure 1. Top: Input and our result on a scene captured in the wild. Note the distortions to the ladder on the top of the fire truck and the landing skids of the helicopter. Bottom: Our laboratory setup for generating large amounts of training data.

The problem is similar to blind deconvolution, but the kernel is spatially varying and can be much larger than what is typically considered in image deblurring. As such, most previous works [6, 7, 9, 27] assume an input video instead of a single frame.

In contrast, we attempt to solve the single image undistortion problem by building upon the recent success of deep convolutional neural networks at solving image-to-image translations [13]. Our hypothesis is that the space of natural images as well as the space of natural refractive distortions is structured enough that a neural network can learn a reasonable mapping between distorted input images and undistorted

*These two authors contributed equally

output images. We demonstrate that it is in fact the case by training a network end-to-end for our task.

Although, in principle, a purely convolutional and deconvolutional network could learn the complex mapping between distorted images and undistorted images directly, we find training such a network to be difficult. Instead we propose a two-step framework to address the nature of images observed through dynamic refraction. The first step outputs a warping field and applies it to the input image to undistort it. Note that we can apply the warping in a differentiable manner by using bilinear sampling. While such a warping network is able to remove many of the geometric distortions, there is often information lost during image formation due to blurring and holes induced by the complex shape of the interface. To correct for this, we train another color network that takes the output of the warp net and hallucinates plausible details. Both the networks are trained together in an end-to-end manner. As has been observed in prior work [19], when the network is trained solely with the L1 or L2 loss, the output images are blurry. To combat this, our network is also trained with adversarial [10] and perceptual losses [19].

To train the network we need a large number of input distorted and ground truth image pairs. Unlike previous works that use computer graphics simulations to generate voluminous data, we find that our application demands a narrower domain gap between training and testing. Since no such dataset with real images currently exists, we construct a new large scale dataset by displaying ImageNet images on a monitor placed under a glass tank full of water and capturing images from above. We demonstrate that by training our network on this dataset we are able to generalize to images of real objects, even in completely different environments (see Fig 1). Our method consistently produces high quality undistorted images from a single distorted input, in contrast to the recent end-to-end learning framework of [14]. Our dataset and code will be publicly released to stimulate further research towards this challenging problem.

In summary, we make the following contributions: 1) propose using deep learning to solve the as yet unattempted problem of single image distortion removal, 2) design a new special network architecture that takes advantage of the physical image distortion model, 3) construct a large scale image dataset that can be used to train our network, 4) show high quality results on real objects imaged through diverse distortions in various settings.

2. Related Work

Imaging Through Refractive Distortions: Water distortion removal is an extremely challenging problem due to its inherent ill-posed nature. To the best of our knowledge, all previous methods assume additional information beyond a single input image. Maybe some works about HW1 with the burst?

One common approach is to use a video sequence of a still scene under varying distortions. Murase et al. [23] proposes the common assumption that the water surface slant is Gaussian with mean zero over time. This means that the temporal average of frames will give a reasonable undistorted image. This suggests the method known as lucky imaging in which the image with the least distortion is chosen as the restoration. Going beyond this, Efros et al. [9] divide the images into patches and choose the best patch for each location across the video sequence and stitch the results into the final result. Donate et al. [6, 7] improve this method by further removing the motion blur and by using k-means clustering to reduce the number of patches being considered for the patch selection process. Wen et al. [31] combines lucky imaging with Fourier domain spectral analysis for better reconstruction. Tian et al. [27] propose a compact spatial distortion model based on the wave equation and use it to design an image restoration technique specifically for water distortion. Periodicity and smoothness constraints for water surfaces are used as regularization to help avoid poor local minima. Oreifej et al. [24] propose an iterative two stage restoration in which the first stage robustly aligns the frames to the temporal mean image and the second stage removes sparse noise using a low rank assumption.

all these might serve as the "radius" papers

Another branch of works focuses on recovering the shape of the water surface from a distorted and non-distorted image pair. Note that this is a slightly different problem than ours as the desired non-distorted image is assumed known. In this case the problem can be posed as an image alignment problem seeking the warping field that warps the distorted image to the undistorted one. Tian et al. [28] develop a data-driven gradient descent algorithm that iteratively recovers the warping field. They first generate a large set of training samples with known distortions. Then in each iteration, they find the nearest neighbor of the current distorted image in the training set and use its distortion parameters to warp the distorted image back to the template. Tian et al. [29] further develop a hierarchical structure which needs much less training samples and can consider global and local distortion simultaneously. Zhang et al. [35] uses defocus and distortion cues from a video along with a non-distorted template to solve for both the water surface and object depth.

Alterman et al. [2] considers the problem of multi-view stereo through a dynamic refractive interface. They use multiple cameras along a wide baseline to observe a scene under uncorrelated distortions and recover sparse point clouds. Xue et al. [32] estimate flow velocity in a dynamic refractive medium using optical flow.

CNNs for Estimating Transformations: Siamese networks have been used for estimating rigid or non-rigid transformations between two images for tasks such as motion estimation or matching [1, 17]. In contrast, we use a single

image for undistortion, since the ground truth target image is not known at test time. The spatial transformer has been proposed as a trainable module in classification networks by Jaderberg et al. [15] to estimate parametric transformations, with a convolutional variant used for correspondence learning in [3]. Non-parametric transformations in the form of a shape basis representation are estimated in [33] to handle articulations. In contrast to those works, we address the problem of distortions induced by waves on the surface of water, which is not a parametric transformation and often too complex to be representable by a small number of bases.

Image-to-Image Deep Learning: Although deep learning first saw great success on the problem of image classification [18], it has also proven very successful on image to image problems such as semantic segmentation [21]. Recently many works have trained convolutional/ deconvolutional networks to perform a variety of image to image problems, such as image super-resolution [8, 19], image colorization [5, 4, 36, 12], image inpainting [25], image style transfer [20], image manipulation guided by user constraints [37] and image de-raining [34].

Many of these works rely on generative adversarial networks (GANs), which have recently shown promise at the task of natural image generation [10]. A GAN consists of two networks: a generator, whose task is to generate realistic looking images, and a discriminator, whose job is to label images from the generator as fake and real images as real. These two networks are trained together forcing the generator to learn to produce realistic images. Despite recent work on improving the training of GANs [26], the resulting images are not yet of high quality. However, when the generator is conditioned on an input image and can can be trained with a traditional loss, such as L1 or L2, in addition to the adversarial loss, the results are much more impressive [19, 14]. The adversarial loss drives the results away from the mean/median image that is learned from solely the L2/L1 loss, which allows the network to learn to predict more detailed, less blurry, realistic looking images.

Isola et al. [14] build upon these to propose a general framework for image-to-image translation problems that involves training a convolutional/ deconvolutional network on input and output image pairs using a combination of L1 pixel loss and an adversarial loss. Although this can be used to solve our problem in principle, our experiments indicate that their general purpose net has difficulty in learning to correct geometric distortions in practice.

3. Model

We train a deep neural network to take in images distorted by a dynamic refractive interface and output the undistorted image that would have been observed without an interface. Although, in theory, a purely convolutional/ deconvolutional architecture such as [14] could learn this complex mapping, we find it does not perform well in practice (see Figure 4). Unlike most previous image-to-image networks [14, 19, 34], we draw inspiration from the physical image formation model to help simplify the problem for the network.

Let $\mathbf{I}(\mathbf{x})$ be the image that would have been observed without any refractive distortion and $\tilde{\mathbf{W}}(\mathbf{x})$ be a 2D warping field that corresponds to the distortion induced by the refractive interface. When the height of the variations of the water are small compared to the depth of the scene and the height of the camera, $\tilde{\mathbf{W}}$ is linearly related to the gradient of the surface height $\nabla Z(\mathbf{x})$. Then the observed, distorted image $\mathbf{J}(\mathbf{x})$ is given by

$$\mathbf{J}(\mathbf{x}) = \mathbf{I}(\mathbf{x} + \tilde{\mathbf{W}}(\mathbf{x})) \tag{1}$$

Unfortunately, inverting 1 is difficult not only since both $\mathbf{I}(\mathbf{x})$ and $\tilde{\mathbf{W}}$ are unknown, but also because the mapping need not be one-to-one.

Inspired by this, we train our network to predict the inverse warping field $\mathbf{W}(\mathbf{x})$ such that

$$\mathbf{I}(\mathbf{x}) = \mathbf{J}(\mathbf{x} + \mathbf{W}(\mathbf{x})) \tag{2}$$

Thus, given a predicted warping field $\mathbf{W}(\mathbf{x})$ from our network, we can easily compute the desired undistorted image by interpolation of the input image. We use bilinear interpolation since it is differentiable, which allows end-to-end training. Here we have taken advantage of the fact that we know the mapping between input and output images to be a warp. By performing the warping explicitly through interpolation, we do not require the network to learn to do it through convolutions.

However as stated above, the forward warping need not be one-to-one and thus information may be lost in the distorted image $\mathbf{J}(\mathbf{x})$. This is often observed as blurring, double images and singularities. To handle this, we train a second image-to-image network, which we call the color network, that takes the unwarped image $\mathbf{J}(\mathbf{x}+\mathbf{W}(\mathbf{x}))$ and outputs our final image. The goal of this second network is to add back details lost during the warping and correct other artifacts that the warping network could not handle (partly due to its limited modeling).

Let our warping network be denoted as $\mathbf{W}_\theta$ and our color network as $\mathbf{C}_\phi$, where θ and ϕ are the learnable parameters of each network, respectively. Then our full generator network is given by

$$\mathbf{G}_{\theta\phi}(\mathbf{J}(\mathbf{x}), \mathbf{x}) = \mathbf{C}_\phi(\mathbf{J}(\mathbf{x} + \mathbf{W}_\theta(\mathbf{J}(\mathbf{x})), \mathbf{x}), \tag{3}$$

which we train end-to-end.

3.1. Network Architecture

The architectures of our warping network $\mathbf{W}_\theta$ and color network $\mathbf{C}_\phi$ are inspired by Ledig et al. [19] and Isola et

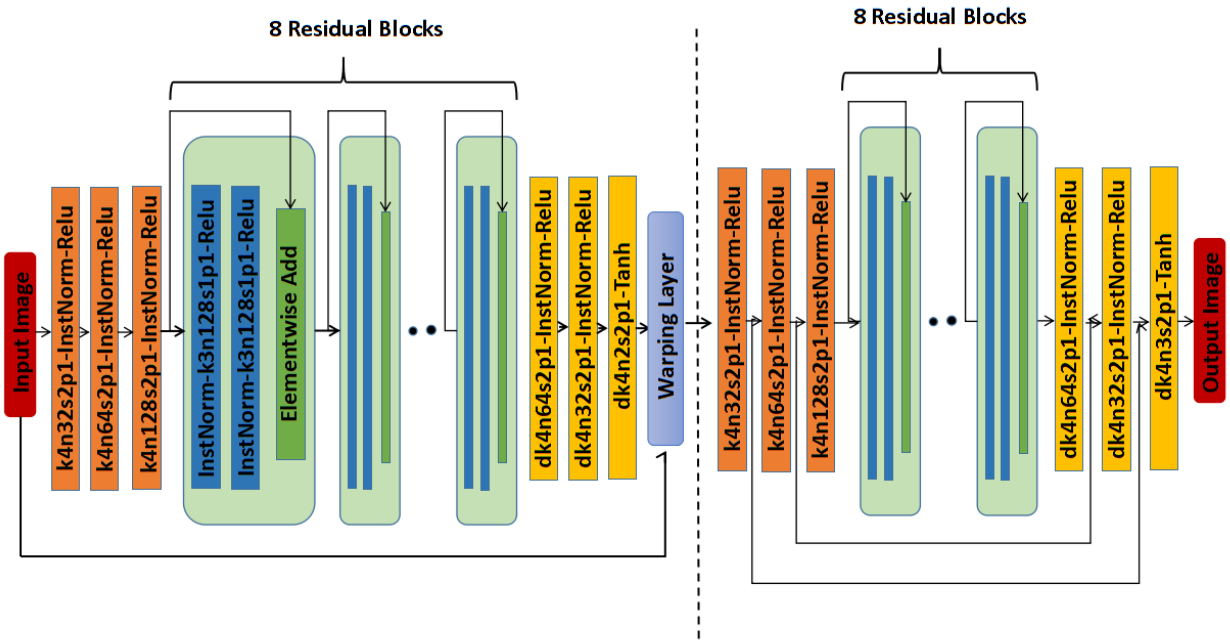

Figure 2. The network structure of our generator. For each convolutional layer, k represents the kernel size, n represents the number of feature maps, s represents the stride and p represents the padding size. Here, d represents the transpose convolutional layer.

al. [14], but we make a few important changes to better suit our problem. Both our networks have the same general structure with only a few differences, which we now discuss in detail.

Both nets consist of three stride 2, size 4 convolution layers, followed by eight residual blocks, followed by three stride 2, size 4 deconvolution layers (see Figure 2). The output feature dimensions are 32, 64, 128, 64, 32, x, where x is a two channel warping field for the warp net and a 3 channel RGB image for the color net. Each residual block consists of two stride 1, size 3, dim 128 convolution layers followed by an additive skip connection, following the design of [11]. We also add concatenation skip connections between corresponding convolution and deconvolution layers of the color net to help maintain fine details in the output image. This is not necessary for the warp net. Note that a similar, but much shallower, two stage network structure was proposed in [16] for the problem of lightfield interpolation where the warps are small.

We find that normalization plays an important role in generalizing from our training set to real objects that have somewhat different color statistics (see Section 4). With standard batch normalization, we achieve the best results (in terms of L1 loss) on the training set, but observe bright blob artifacts when testing on real objects. This is due to the network over fitting to the color statistics of our training images. The problem is not alleviated solely by using instance normalization as suggested by [30] because unlike them, we expect the network to preserve the brightness and contrast of the input image. To address this, we use instance normalization throughout our network, but save the mean and variance extracted from the input layer and use it to scale and shift the output.

3.2. Training Objective

We train our network by minimizing the L1 loss in pixel space

$$L_{con} = \sum_{\mathbf{x}} |\mathbf{I}(\mathbf{x}) - \mathbf{G}_{\theta\phi}(\mathbf{J}(\mathbf{x}), \mathbf{x})| \quad (4)$$

which we call the content loss. However, the L1 loss alone trains the network to predict the median image, which is often blurry and lacking in high frequency details. As in [14, 19] we also train our network with an adversarial loss to help encourage the predicted images to reside on the natural image manifold. This forces the network to produce sharp images with more fine details, and even hallucinate missing information from large distortions.

use plug-and-play priors? (Miki's talk) combined with L1 and ADV?

We train an additional discriminator network D_γ to distinguish between undistorted images from the generator and the natural non-distorted images, while the generator is trained to fool the discriminator. During the training process, the discriminator and generator are trained in an alternating manner to solve the min-max problem

$$\min_{\theta,\phi} \max_{\gamma} \; \mathbf{E}[\log D_\gamma(\mathbf{I})] + \mathbf{E}[\log(1 - D_\gamma(G_{\theta\phi}(\mathbf{J}))\,)] \quad (5)$$

For more stable training we use the Least Squares GAN objective [22]

$$L_{adv} = -(D_\gamma(G_{\theta\phi}(\mathbf{J})) - 1)^2 \quad (6)$$

The discriminator architecture follows the guidelines proposed in [26]. We use 7 convolutional layers with kernel size 4 and stride 2 and increasing feature dimension (32,64,128,256,512,512,1). Each convolution except the last is followed by batch normalization and LeakyReLU activations. The last output is followed by a sigmoid activation. We also try the *PatchGAN* [13] by decreasing the receptive field of discriminator to 70×70 and apply it through the image convolutionally. However, in our case, using *PatchGAN* does not improve the image quality.

Although the adversarial loss encourages more details, it also introduces some artifacts due to the unstable nature of GAN training. To combat this we follow [19] and add a perceptual loss defined by

$$L_{per} = \sum_{\mathbf{x}} |\psi(\mathbf{I}(\mathbf{x})) - \psi(\mathbf{G}_{\theta\phi}(\mathbf{J}(\mathbf{x}), \mathbf{x}))|, \quad (7)$$

where ψ is the output of an intermediate feature layer of a pretrained convolutional neural net. In our implementation we use the output of the conv4_3 layer of VGG.

Our final loss function is a weighted combination of the 3 losses

$$L = L_{con} + \lambda_{adv} L_{adv} + \lambda_{per} L_{per} \quad (8)$$

Training Detail: We largely follow the training scheme in [19] and [26]. We first train the network with L1 loss alone from scratch and then fine tune the network adding adversarial loss and perceptual loss. The weight for adversarial loss and perceptual loss are 0.0005 and 0.3 respectively. Compared with [19], our weight for adversarial loss and perceptual loss is much lower and we do not remove L1 loss when fine tuning the network. This is because we observe that L1 loss is important for our problem and if we remove L1 loss the network will not generate reasonable results. When training with L1 loss, we set the learning rate to be 0.001 and divide it by 10 after 15000 iterations. We train the network for 30,000 iterations with a batch size of 32. Then we fine tune the network adding perceptual loss and adversarial loss for 2000 iterations with learning rate 0.0002 and batch size 16.

4. Training Data

To train our deep network, we need a large training set. However, collecting a large number of images distorted by a water surface along with the corresponding non-distorted ground truth is challenging. There are no such existing large scale datasets. Tian et al. [27] provide a small dataset but that is not nearly enough to train a deep network.

Synthetic data is a natural option, but we found generating diverse enough water surfaces to be challenging. We tried using Gaussian Processes and the wave equation as in [27], as well as perturbing the surface with random Gaussian shaped drops. In each case, the network quickly over fit to the particular distribution of water surfaces generated and failed to generalize to real images. Creating diverse synthetic water surfaces is an interesting direction for future work.

might use data augmentation for that case?

Instead, we choose to construct a large dataset of distorted and non-distorted image pairs by capturing images of ImageNet images displayed under a water surface (see Figure 1). We place a computer monitor under a glass tank, which is filled with approximately 13cm water. The water is kept in motion using a small agitating pump. A Cannon 5D Mark IV is placed approximately 1.5m above the tank. Images are resampled using bilinear interpolation to fill the available screen space in the tank, after which the captured image is tightly cropped to its original shape and downsampled to its original size. The camera is set to f/1.2, ISO100, with exposure time of 1/320s. The camera is manually focused just beyond the monitor as this slight defocus removes the Moiré pattern observed in properly focused images.

The process of displaying and recapturing the images changes the color space slightly due to nonlinear gamma curves and pixel sensitivity. To handle this, we pretrain a small color correction net that consists of 6 convolutional layers with receptive fields of size 1 to minimize the L1 distance between the captured image and the original ImageNet image. This mostly solves the problem, however we find that proper normalization in the net (as described in Section 3.1) is important for generalization to real objects. We collect 324,452 images from all 1000 ImageNet categories. We withhold 5 images from each category to form a validation set of 5000 images.

We note that creating a large real image dataset for 3D underwater scenes in the wild is extremely difficult. The intent of our training data collection is to easily generate sufficient volume in conditional similar but not identical to the application scenario. The choice of using flat images is a deliberate one, sacrificing realism for quantity. This is in line with several studies that use simulations for generating training data. Our laboratory setup similarly allows collecting large-scale data, but with reduced domain gap. Our experiments demonstrate generalization from the laboratory tank setup with flat images to real images of non-flat objects and in wild settings.

5. Results

We show results on our validation set captured by displaying ImageNet images on a monitor under a water surface, as well as images of real objects underwater. To demon-

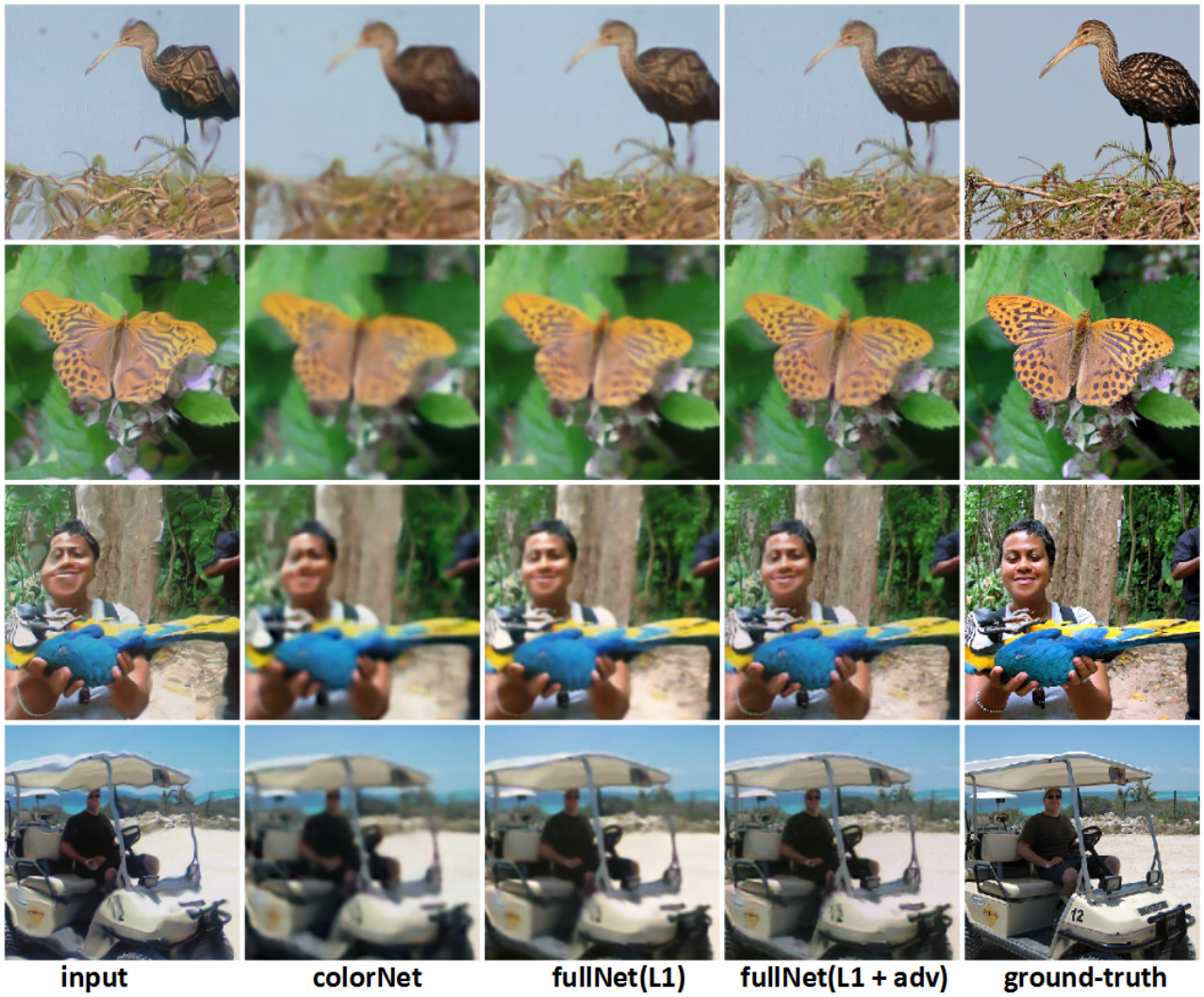

Figure 3. Qualitative results for ablative study on ImageNet validation test set. From left to right: input image, color net with L1 loss, warp+color net with L1 loss, warp+color net with L1+Adv+Per losses, ground truth. We observe that estimating the undistortion with the warp net significantly improves the geometry, while the adversarial loss allows better perceptual alignment to ground truth.

Method	L1	MSE	PSNR	SSIM
colorNet	20.318	998.631	18.841	0.470
warpNet	20.140	961.978	19.035	0.490
fullNet(L1)	19.091	902.032	19.306	0.502
fullNet(adv+L1+per)	19.109	894.178	19.348	0.499

Table 1. Quantitative results for the ImageNet validation set. The network is trained with L1 loss, whereby we observe that L1 error reduces for the full network compared to warp or color net alone. As expected, the adversarial loss increases the L1 error, but allows for better appearance. For completion, we also show other metrics not directly related to the training, such as MSE, PSNR and SSIM.

strate generalization ability, the real objects are imaged in a different larger tank, as well as outdoors in a fountain pool.

In Figure 3, we show our results and an ablation study on the ImageNet validation set. In addition to the input image (column 1), our result (column 4) and ground truth (column 5), we also show two ablation results. The first is our color net alone trained with only the L1 loss (column 2). The second is our full generator architecture but trained with only the L1 loss (column 3). The five rows show the outputs for different input images.

We observe that the color net alone struggles to remove the large geometric distortions. Adding the warp net that accounts for the structure of the problem results in significantly better geometric undistortion, while also producing good colors. Next, adding the adversarial and perceptual loss has the effect of recovering sharp detailed images that are perceptually closer to the ground truth.

Figure 4 shows our results on real objects as well as a

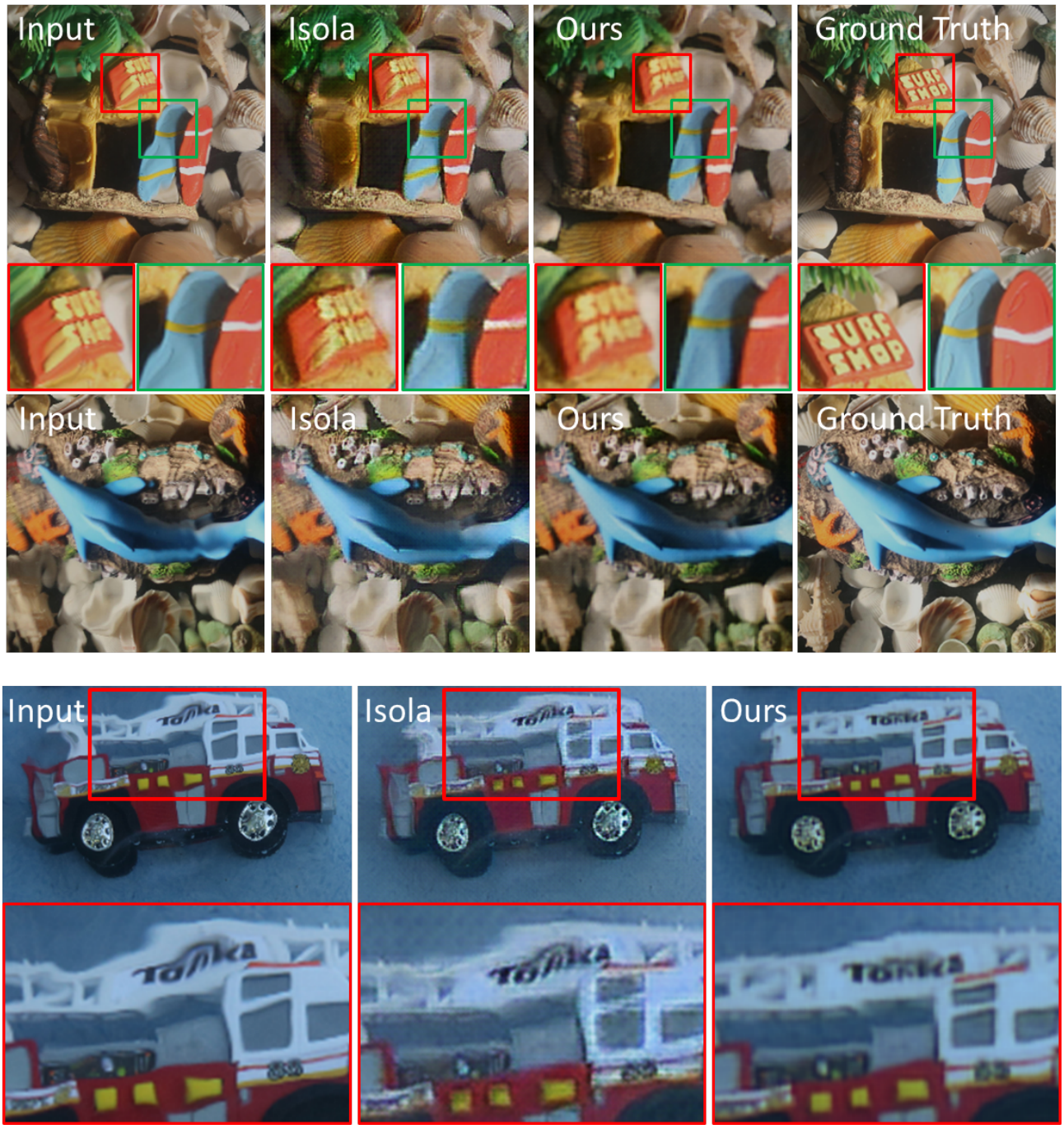

Figure 4. Results on real objects demonstrating generalization. (Rows 1 and 2) From left to right: input image in a larger tank, result of Isola et al. [14], our result and ground truth. We observe that our framework that uses problem structure and careful normalizations produces better geometric undistortion and color outputs. (Bottom row) We show another example of further generalization by acquiring an image in a fountain pool. We see more significant contrasts relative to [14], with clearly better undistortion performance for our method.

comparison to a state-of-the-art method for image to image translation [14]. This method is similar to our color net alone but does not generalize well to real data due to the normalization issues discussed above. It also does not take advantage of domain knowledge that the transformation is a warp. Due to these factors, we observe that our method produces results that are closer to ground truth as compared to [14]. This is emphasized by the insets, showing better geometric warp estimation in regions with long edges and also better color estimates than [14] which produces subtle

checkerboard artifacts.

Although no previous work in the water undistortion literature attempts the problem of single image blind undistortion, we note that methods such as [28] estimate a warping field by assuming the ground truth nondistorted image is known. In comparison, we do not require the assumption of a template, which might not exist in wild settings. Even in lab settings, acquiring a template requires careful alignment of images before and after the water surface is agitated. Other works such as [27] additionally assume high frame rate video inputs, whereas we require only a single image.

Finally, in Figures 1 and 4, we show example outputs on an underwater sequence captured in a wild setting. We use the same network trained on ImageNet images observed through distortions in a tank, but the test images in this experiment are acquired outdoors at a water fountain. While there is no available ground truth, it is observed that the network generalizes quite well to this unseen condition, as reflected by the undistortion output that preserves edge shapes and displays plausible colors.

6. Conclusion

We have proposed a novel approach that uses deep learning to solve the previously unattempted problem of using a single image to remove distortions due to a refractive interface such as water surface. Since a turbulent water surface induces distortions that are too complex to be modeled as parametric or basis transformations, we use domain knowledge to model the distortion as a warp. This is different from general purpose image to image translation networks, which does not utilize problem structure. We demonstrate in experiments that our formulation as an end-to-end trainable two-stage network that estimates geometry and color, along with careful consideration of normalizations, leads to better results and generalization ability. To train our network, we collected a large scale dataset in lab settings with displayed images and show that it generalizes to images of real scenes imaged in different settings including unconstrained ones. Our work also opens the doors to several directions of future research. In particular, we will consider extensions to recover the shape of dynamic refractive interfaces using our estimated warps, as well as imaging in participating media that introduce other interesting distortions such as scattering.

Acknowledgments

This work was supported by the US Office of Naval Research grant N000141512013 and the UC San Diego Center for Visual Computing. We gratefully acknowledge the support of NVIDIA Corporation with the donation of a Titan X Pascal GPU used for this research.

References

[1] P. Agrawal, J. Carreira, and J. Malik. Learning to See by Moving. In *ICCV*, 2015.

[2] M. Alterman, Y. Schechner, and Y. Swirski. Triangulation in random refractive distortions. *IEEE transactions on pattern analysis and machine intelligence*, 2016.

[3] C. B. Choy, J. Gwak, S. Savarese, and M. Chandraker. WarpNet: Weakly supervised matching for single-view reconstruction. In *NIPS*, 2016.

[4] A. Deshpande, J. Lu, M.-C. Yeh, and D. Forsyth. Learning diverse image colorization. *arXiv preprint arXiv:1612.01958*, 2016.

[5] A. Deshpande, J. Rock, and D. Forsyth. Learning large-scale automatic image colorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 567–575, 2015.

[6] A. Donate, G. Dahme, and E. Ribeiro. Classification of textures distorted by waterwaves. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 2, pages 421–424. IEEE, 2006.

[7] A. Donate and E. Ribeiro. Improved reconstruction of images distorted by water waves. In *Advances in Computer Graphics and Computer Vision*, pages 264–277. Springer, 2007.

[8] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2016.

[9] A. A. Efros, V. Isler, J. Shi, and M. Visontai. Seeing through water. In *NIPS*, volume 17, pages 393–400, 2004.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[12] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Transactions on Graphics (TOG)*, 35(4):110, 2016.

[13] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016.

[14] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016.

[15] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

[16] N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM Transactions on Graphics (TOG)*, 35(6):193, 2016.

[17] A. Kanazawa, D. W. Jacobs, and M. Chandraker. WarpNet: Weakly supervised matching for single-view reconstruction. In *CVPR*, 2016.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[19] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint arXiv:1609.04802*, 2016.

[20] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European Conference on Computer Vision*, pages 702–716. Springer, 2016.

[21] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[22] X. Mao, Q. Li, H. Xie, R. Y. Lau, and Z. Wang. Multi-class generative adversarial networks with the l2 loss function. *arXiv preprint arXiv:1611.04076*, 2016.

[23] H. Murase. Surface shape reconstruction of a nonrigid transparent object using refraction and motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10):1045–1052, 1992.

[24] O. Oreifej, G. Shu, T. Pace, and M. Shah. A two-stage reconstruction approach for seeing through water. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1153–1160. IEEE, 2011.

[25] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016.

[26] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[27] Y. Tian and S. G. Narasimhan. Seeing through water: Image restoration using model-based tracking. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2303–2310. IEEE, 2009.

[28] Y. Tian and S. G. Narasimhan. Globally optimal estimation of nonrigid image distortion. *International journal of computer vision*, 98(3):279–302, 2012.

[29] Y. Tian and S. G. Narasimhan. Theory and practice of hierarchical data-driven descent for optimal deformation estimation. *International Journal of Computer Vision*, 115(1):44–67, 2015.

[30] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[31] Z. Wen, A. Lambert, D. Fraser, and H. Li. Bispectral analysis and recovery of images distorted by a moving water surface. *Applied optics*, 49(33):6376–6384, 2010.

[32] T. Xue, M. Rubinstein, N. Wadhwa, A. Levin, F. Durand, and W. T. Freeman. Refraction wiggles for measuring fluid depth and velocity from video. In *European Conference on Computer Vision*, pages 767–782. Springer, 2014.

[33] X. Yu, F. Zhou, and M. Chandraker. Deep deformation networks for object landmark localization. In *ECCV*, 2016.

[34] H. Zhang, V. Sindagi, and V. M. Patel. Image de-raining using a conditional generative adversarial network. *arXiv preprint arXiv:1701.05957*, 2017.

[35] M. Zhang, X. Lin, M. Gupta, J. Suo, and Q. Dai. Recovering scene geometry under wavy fluid via distortion and defocus analysis. In *European Conference on Computer Vision*, pages 234–250. Springer, 2014.

[36] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *European Conference on Computer Vision*, pages 649–666. Springer, 2016.

[37] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision*, pages 597–613. Springer, 2016.