https://doi.org/10.2352/ISSN.2470-1173.2020.8.IMAWM-086

High-quality multispectral image generation using Conditional GANs

Ayush Soni Alexander Loui Scott Brown Carl Salvaggio
Rochester Institute of Technology, Rochester, NY.

Abstract

In this paper, we demonstrate the use of a Conditional Generative Adversarial Networks (cGAN) framework for producing high-fidelity, multispectral aerial imagery using low-fidelity imagery of the same kind as input. The motivation behind is that it is easier, faster, and often less costly to produce low-fidelity images than high-fidelity images using the various available techniques, such as physics-driven synthetic image generation models. Once the cGAN network is trained and tuned in a supervised manner on a data set of paired low- and high-quality aerial images, it can then be used to enhance new, lower-quality baseline images of similar type to produce more realistic, high-fidelity multispectral image data. This approach can potentially save significant time and effort compared to traditional approaches of producing multispectral images.

Introduction

In remote sensing applications, having access to high-fidelity, multispectral aerial imagery is necessary for various types of scene analysis. Traditionally, such imagery is collected using multiple sensors from satellites, aircrafts, or Unmanned Aerial Vehicle (UAV) platforms, or is generated using physics-driven softwares/models capable of producing realistic multispectral imagery, such as DIRSIG [10]. The latter approach, when appropriate, has some drawbacks. Specifically, the amount of time and effort involved in designing and rendering of high-fidelity imagery can still be very high. This time can be significantly reduced by producing less accurate or approximate simulations of aerial scenes, but doing so can significantly reduce image fidelity. It is therefore important to design an approach to reduce this time and effort, without significantly compromising the quality of the image data.

The objective of this research is to use algorithms which, given a data set pairing low- and high quality multispectral images, can learn an accurate mapping between the two. Once such a mapping is learned, it can then be used on new low quality images of similar content, to generate corresponding high quality, realistic images. The trained algorithm should be able to generate R, G, B as well as other spectral channels in the produced data set, using the same common framework. We look at this problem from image-to-image translation perspective, where given a low quality image, we try to translate it into a higher quality image.

Related work

Generative Adversarial Networks

Among various generative modelling methods, GAN as first proposed by Goodfellow et al. [1] and its various improved derivatives have become the most popular because of their ability to generate high quality data. GANs aim to model the data distribution of real samples by forcing the generated samples to be as similar to the real samples as possible. These networks leverage adversarial training, where the Generator is trained to produce "fake" samples that are indistinguishable from "real" samples, while the Discriminator is trained to accurately distinguish between generated "fake" samples and "real" samples. The Generator is penalized based on the feedback from Discriminator, hence it can be said that the Discriminator acts as a "learned" loss function rather than a task-specific, hand-crafted loss, which makes this framework general purpose and useful across various different generative tasks. Simply put, GANs can generate data from a random noise vector given as input.

In [3], A. Radford et al. first proposed the DCGAN architecture which was specifically aimed at modelling the distribution of image samples. They proposed various improvements over GAN, one of which was to use strided Convolutions (in Discriminator) and fractional-strided Convolutions (in Generator), also referred to as Deconvolutions or Transposed Convolutions. This introduction of Convolutional layers, owing to their success on various vision tasks, coupled with other improvements resulted in much higher quality generated images than vanilla GANs[1]. This work in addition to [1] drew a lot of attention and lead to many other papers introducing newer ideas such as improved objective functions [11], stabilized training on larger resolution images [12][13] and more, further improving image quality.

Conditional GANs and image-to-image translation

In [1], Goodfellow et al. discussed the idea of adding more information to the Generator input signal in addition to the random noise vector to get desired outputs. This idea was further explored in the first Conditional GAN paper [2] by Mirza et al, where they give an image as an input to the Generator network, and adapt the output based on this image. Thus, in a GAN, the Generator G learns the mapping from a random noise vector z to an output image y, $G : z \rightarrow y$, whereas in a cGAN, G learns the mapping from a random noise vector z as well as an input image x to an output image y, $G : \{x, z\} \rightarrow y$. The training of a cGAN network hence requires the input and output image pairs $\{x, y\}$, making it a supervised learning algorithm, in contrast to the unsupervised nature of GANs.

Image-to-image translation is defined as a task of translating one representation of an image to another, given sufficient training data [4]. Conditional GANs are suitable for this task since they can efficiently learn a mapping from input images to output images. A lot of work has been done in image-to-image translation in recent years. In [4], P. Isola et al. use a Conditional GANs framework capable of translating images from arbitrary input to

arbitrary output domains, exploiting the fact that the loss function (the Discriminator network) quantifying the difference between generated and real images, is also learned. The proposed Pix2Pix architecture consists of a Generator based on a U-Net [9] encoder-decoder architecture for generating images conditioned on input images, and a PatchGAN/Markovian Convolutional Discriminator as described in detail in [5], which works by classifying smaller 70×70 patches in an image as "real" or "fake" instead of classifying the entire image. The L1 loss between Generated and Real images was also added to the cGAN adversarial loss, to capture pixel-level differences. This work was one of the major breakthroughs in image-to-image translation and established a strong baseline for comparison in future papers. Later, Wang et al. proposed a major improvement [7], over the Pix2Pix baseline in their Pix2PixHD architecture, which addresses the problem of unstable training and poor quality when generating higher resolution images (greater than 256×256). This architecture is one of the best state-of-the-art methods existing today for high-resolution conditional image synthesis, which we also use as a baseline for our work. The details of the architecture and the specific modifications we did over it for our task are discussed in the Method section.

Lastly, in their paper [8], Milz et al. use the same Pix2Pix cGAN as in [4] on Multispectral Aerial Images. They do various interesting aerial image-to-image translation experiments that are relevant to remote-sensing applications. These include translation from one spectral domain to the other and vice-versa, such as Aerial RGB↔Semantic Map, Aerial RGB↔LiDAR Height Map, Aerial RGB↔LiDAR Elevation Map and 2D Object Box labels↔Aerial RGB. These experiments demonstrate that cGANs can extract features from not just RGB but also other spectral channels. Our work is similar to this idea but we don't just work with a single spectral domain in inputs and outputs, but demonstrate that aerial images with multiple spectral channels can be generated using cGANs, conditioned again on multispectral images. Such multispectral image-to-image translation with cGANs hasn't been explored before and is a novel area of research which can lead to development of new techniques that are useful for various remote-sensing applications.

Method

As mentioned before, we use the Pix2PixHD [7] architecture as a baseline and modify it as per our needs. This decision was made after initial experiments with several different architectures, and among them the Pix2PixHD architecture gave the best qualitative results. The architecture details are as follows. The architecture uses a coarse-to-fine Generator and a multi-scale Discriminator. The coarse-to-fine Generator G consists of a global generator network G_1 and one or more local enhancer networks $\{G_2, G_3, ...\}$. The Generator G is hence given as $G = \{G_1, G_2, G_3, ...\}$. If there are n such networks, the last network G_n operates on the original image of resolution $N \times N$, while each preceding network operates on a down-sampled image of resolution $\frac{1}{4}^{th}$ that of its succeeding one. Thus, G_{n-1} operates at $\frac{N}{2} \times \frac{N}{2}$, G_{n-2} at $\frac{N}{4} \times \frac{N}{4}$, and so on. This way, the first network- the global generator- extracts features at the coarsest scale while the last local enhancer network extracts features at the finest scale. Each G_i consists of a Convolutional front-end, followed by a number of Residual blocks [6], and ending with a Deconvolutional or Transposed Convolutional back-end. In addition to these 3 components, a final convolution layer is applied only to the output of the last local-enhancer G_n to get the output image. For global generator G_1, the image is simply passed as input and processed sequentially through these three components to obtain the output, but for each local enhancer $G_i; i \geq 2$, the output of the front-end is element-wise added with the output of the last feature-map of G_{i-1}'s back-end before feeding into Residual blocks. This addition lets us add information extracted by G_{i-1} at a smaller scale to G_i before it extracts its own features. Such coarse-to-fine structure helps accumulate information from image at multiple-scales, which works very well for generating high-resolution images.

A similar idea is used in the multi-scale Discriminator, which consists of multiple networks, given as $D = \{D_1, D_2, D_3, ...\}$. All D_i have the same PatchGAN architecture similar to the one described in Pix2Pix [4], but operate on different scales. D_1 is trained to classify image as "fake" or "real" at scale $N \times N$, D_2 at $\frac{N}{2} \times \frac{N}{2}$, D_3 at $\frac{N}{4} \times \frac{N}{4}$, and so on. The losses of all the $D_i s$ are added together, which captures the classification loss at various scales from coarsest to finest.

In our final architecture used for training, we make several changes to the original architecture used in [7]. These changes were made taking various factors into account, such as the resolution of images, number of spectral channels in images, training times and observed qualitative results. In our architecture, we use two local enhancer networks in the coarse-to-fine Generator in addition to the global generator, and three PatchGAN networks in the multi-scale Discriminator. Each network in the Generator consists of three Downsampling and three Upsampling layers in the Convolutional front-end and Deconvolutional back-end respe-

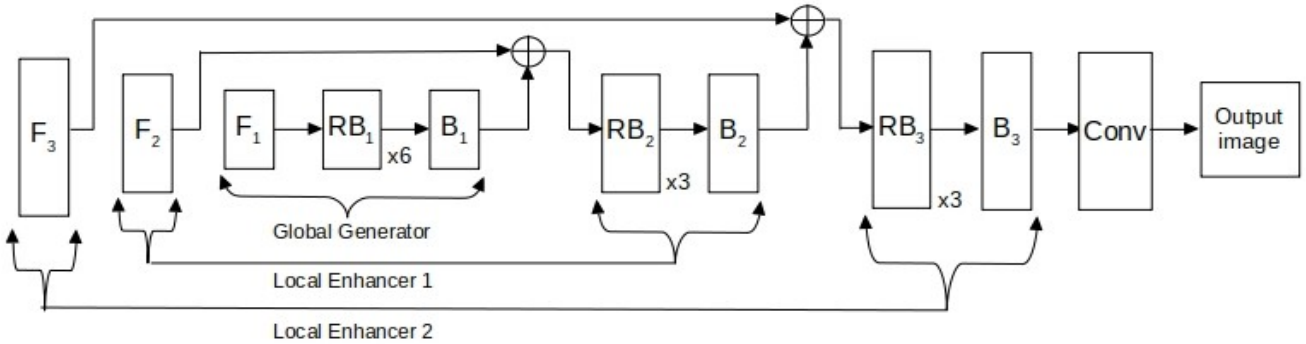


Figure 1. The Generator architecture. F_i, RB_i and B_i denote the Convolutional front-end, Residual Blocks, and the Deconvolutional back-end of each network respectively. "Conv" denotes the final convolution layer which gives the output image.

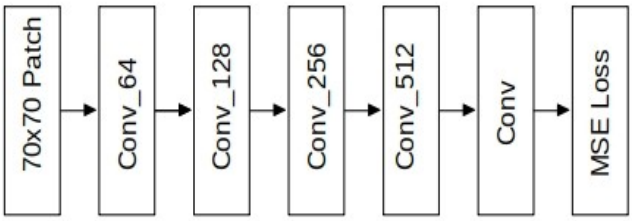


Figure 2. Architecture of the three Discriminators. Input is a 70×70 patch of the image. Conv_k denotes a Convolution-BatchNorm-LeakyReLU layer with k filters, 4x4 kernel, stride=2. As an exception, BatchNorm is not used in the first layer. The leakyReLUs have a slope=0.2. The last "Conv" layer gives a 1-d output which is used for MSE calculation.

ctively. The global generator contains six Residual blocks in between the front-end and back-end layers, while each local enhancer contains three such blocks. The schematic diagrams of the Generator and Discriminator architectures are shown in Figure 1 and 2 respectively.

The objective function contains two components. Lets say the dataset contains pairs $\{x_i, y_i\}$ of images, where x_i is the input low-quality image and y_i is the corresponding high-quality image. The first component is the normal cGAN adversarial loss, which for three discriminators in our case can be written as:

$$\sum_{i=1,2,3} V_{Adv}(G, D_i) \tag{1}$$

where the function $V_{Adv}(G, D_i)$ is given by:

$$V_{Adv}(G, D_i) = \mathbb{E}_{(x,y)}[\log D_i(x,y)] + \mathbb{E}_x[\log(1 - D_i(x, G(x)))] \tag{2}$$

In addition, we also use a feature matching loss taken from Discriminator as recommended by the authors in [7]. This loss is calculated by computing the L1 loss between the intermediate layer features of Discriminators for real and generated images. When the Generator is penalized on this loss, it is further encouraged to produce realistic-looking images. If there are T total layers in each Discriminator and the k^{th} layer in i^{th} Discriminator is represented as $D_i^{(k)}$, this loss is given by:

$$V_{FM}(G, D_i) = \mathbb{E}_{(x,y)} \sum_{k=1}^{T} \frac{1}{N_k} \parallel D_i^{(k)}(x,y) - D_i^{(k)}(x, G(x)) \parallel_1 \tag{3}$$

This loss is added to the adversarial loss, and the final objective function is given by:

$$\sum_{i=1,2,3} V_{Adv}(G, D_i) + \lambda \sum_{i=1,2,3} V_{FM}(G, D_i) \tag{4}$$

where λ is a scalar weight assigned to the feature matching loss, and $\lambda = 20$ in our case. For training this architecture, the whole objective needs to be minimized with respect to the Generator and the adversarial loss needs to be maximized with respect to the Discriminators. Thus, the final learning problem is given as:

$$\min_G \left(\max_{D_1, D_2, D_3} \left(\sum_{i=1,2,3} V_{Adv}(G, D_i) \right) + \lambda \sum_{i=1,2,3} V_{FM}(G, D_i) \right) \tag{5}$$

Experiments and results

Datasets

We have created two datasets and conduct experiments on both. In both the datasets, we have pairs $\{x_i, y_i\}$ of 512×512 images, where each x_i and y_i is a low-quality and the corresponding high-quality image respectively. These images are generated using a physics-driven scene simulation software called DIRSIG5 [10]. The images generated are randomly extracted from a realistic simulation of a large aerial scene of an urban setting consisting of houses, trees, terrain, vehicles, buildings, and several other object types, each with multiptle different, diverse geometries. In the first dataset, the images contain R, G, B channels, and also Edge maps and Semantic maps. The Edge maps contain 1s on the pixels corresponding to edges of the objects and 0s everywhere else. These were extracted from the corresponding instance maps, and are useful to guide the Generator in producing sharp object edges. Similarly, the Semantic maps are per-pixel labelled maps, wherein each pixel is labelled with one of the 140 geometries in the scene. These help the Generator in generating specific objects in the images more accurately. The low-quality images in this dataset have a poor quality of textures across all the objects in the scene, while the high-quality images have the best quality, realistic textures. We will refer to this dataset as Dataset1.

In the second dataset, the images contain R, G, B, Near-infrared, Red-edge, LiDAR channels as well as Edge and Semantic maps. DIRSIG5 uses ray-tracing which traces the light rays reaching a pixel to their sources, to solve for the value of that pixel. This solution is effectively the Monte-Carlo integral of all the light paths reaching the pixel, and hence the fidelity of the solution depends on the number of paths traced [10]. Using less no. of paths per pixel (or samples) gives a lower quality, approximate but faster solution. On the other hand, using a large no. of paths per pixel gives a more accurate, high-fidelity solution at the cost of higher computation time. The low-quality images in this dataset were obtained by fixing the no. of such samples taken per pixel to one. The high-quality images on the other hand were obtained with the default DIRSIG5 adaptive sampling, wherein the no. of samples may range from 20 to 100, depending on the complexity of the pixel. The time taken for generating 2000 512×512 low-fidelity images was approximately 1 hour and 10 minutes, as compared to upwards of 9 hours for high-fidelity images. Hence, it can be said that a cGAN model capable of accurately translating the low-fidelity images to high-fidelity can potentially save hours of computation time taken to generate high-fidelity images using software such as DIRSIG5. We will refer to this second dataset as Dataset2. Both these datasets have a total of 2000 image pairs, out of which 85% (1700) are used for training, and the rest 15% (300) are used for testing.

Experiments

We present visual qualitative results obtained on both the datasets. In both the cases we start with a learning rate of $1e-4$ and train with this rate for first 20 epochs. After 20 epochs, we linearly decay the learning rate every epoch and train for 80 more epochs. A batch size of 2 was used in both the cases. We also use Batch Normalization layers in both the Generator and Discriminator networks. For Dataset1, the task is to translate the images with poor texture quality to those with higher texture quality. Some examples of input, generated and ground-truth images are shown

in Figure 3. It can be observed from these visualizations that the model successfully learns to fill the low-quality texture pixels with higher quality, realistic-looking textures. For Dataset2, the task is more complex and challenging. Here, we take low-quality input images with 8 channels: {R, G, B, Near-infrared, Red-edge, LiDAR, Edge-maps, Semantic-maps} and try to generate higher-quality images with 5 channels:{R, G, B, Near-infrared, Red-edge}. The Edge-maps and Semantic-maps are only used as inputs to guide the Generator and are not produced in the outputs. After training, the model learns to map the low quality 8-channel inputs to the higher quality 5-channel outputs. We also tried training the network with several other combinations of input and output channels for this dataset and observed their results. Among all these experiments, the aforementioned experiment was the most complex one, and produces good results on par with the results of other, simpler experiments. Hence we only show the results of this experiment. The composite as well as individual channels of a sample test input, generated and ground-truth images are shown in Figure 4. From these visualizations, it can be observed that the images generated by the model are more realistic than the low-fidelity input images. These however are still not as perfect as the ground-truth images, which leaves a scope of improvement and further research in this area.

Conclusion

In previous works, Conditional GANs have been demonstrated to work on image-to-image translation tasks involving RGB images. In this paper, we explored their use for multispectral image-to-image translation and demonstrate their feasibility on this task. We specifically focused on low-quality to high-quality translation of aerial images of an urban scene, but cGANs being a general purpose architecture, this approach can be generalized to work on any type of multispectral images given sufficient data. The qualitative results show that the presented method produces images with a reasonably good quality, however, further scope of improvement remains in order for such approach to be a complete alternative to present methods of high-fidelity image generation.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, Generative Adversarial Networks, NIPS (2015).

[2] M. Mirza and S. Osindero, Conditional Generative Adversarial Networks, arXiv preprint arXiv:1411.1784 (2014).

[3] A. Radford, L. Metz and S. Chintala, Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, arXiv preprint arXiv:1511.06434 (2016).

[4] P. Isola, J. Zhu, T. Zhou and A. Efros, Image-to-Image Translation with Conditional Adversarial Network, CVPR (2017).

[5] J. Long, E. Shelhamer and T. Darrell, Fully Convolutional Networks for Semantic Segmentation, IEEE TPAMI, Volume: 39, Issue: 4 (2017).

[6] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, CVPR (2016).

[7] T. Wang, M. Liu, J. Zhu, A. Tao, J. Kautz and B. Catanzaro, High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs, CVPR (2018).

[8] S. Milz, T. Rüdiger and S. Süss, Aerial GANeration: Towards Realistic Data Augmentation Using Conditional GANs, ECCV (2018).

[9] O. Ronneberger, P. Fischer and T. Brox, U-net: Convolutional networks for biomedical image segmentation, MIC-CAI (2015).

[10] A. Goodenough and S. Brown, DIRSIG 5: core design and implementation, Proc. SPIE 8390, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XVIII, 83900H (2012).

[11] M. Arjovsky, S. Chintala and L. Bottou, Wasserstein GAN, arXiv preprint arXiv:1701.07875 (2017).

[12] T. Karras, T. Aila, S. Laine and J. Lehtinen, Progressive Growing of GANs for Improved Quality, Stability, and Variation, ICLR (2018).

[13] A. Brock, J. Donahue and K. Simonyan, Large Scale GAN Training for High Fidelity Natural Image Synthesis, ICLR (2019).

Author Biography

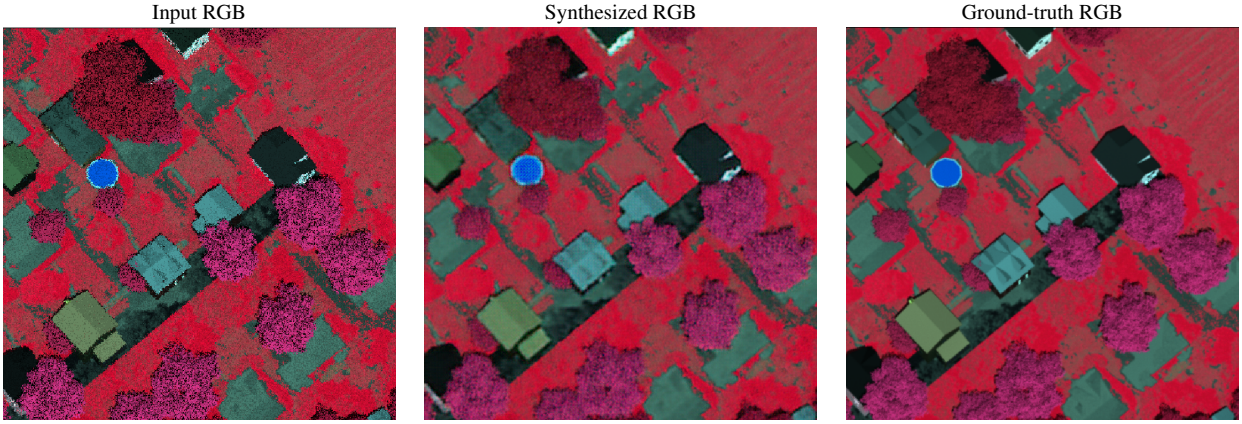
Ayush Soni is currently a second year Master's student at Rochester Institute of Technology. He obtained his Bachelor's degree in Information Technology Engineering in India and joined RIT's CS MS program thereafter to pursue his specialization in Intelligent Systems. His primary areas of interest include Applied Machine Learning, Computer Vision and Cloud Computing. His recent projects have revolved around practical application of Computer Vision methods for Classification, Segmentation and Conditional Image Generation to real-world problems.

Figure 3. (Continued on next page.)

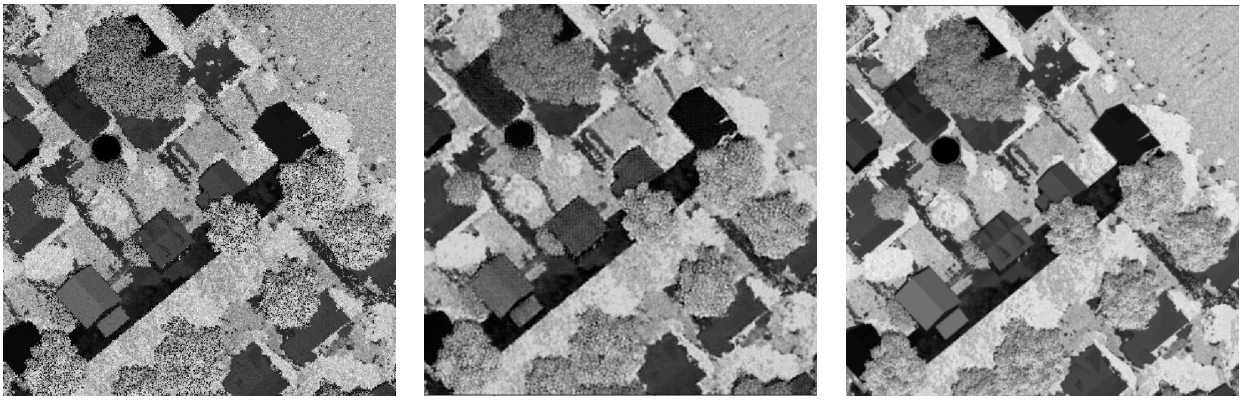
Input RGB | Synthesized RGB | Ground-truth RGB

Figure 3. Two examples of synthesized RGB images(center) from input low texture quality RGB images(left) of Dataset1 test samples. The corresponding high texture quality ground-truth images(right) are shown for comparison.

Input RGB | Synthesized RGB | Ground-truth RGB



Input {Near-infrared, R, G} composite | Synthesized {Near-infrared, R, G} composite | Ground-truth {Near-infrared, R, G} composite



Input Near-Infrared | Synthesized Near-Infrared | Ground-truth Near-Infrared

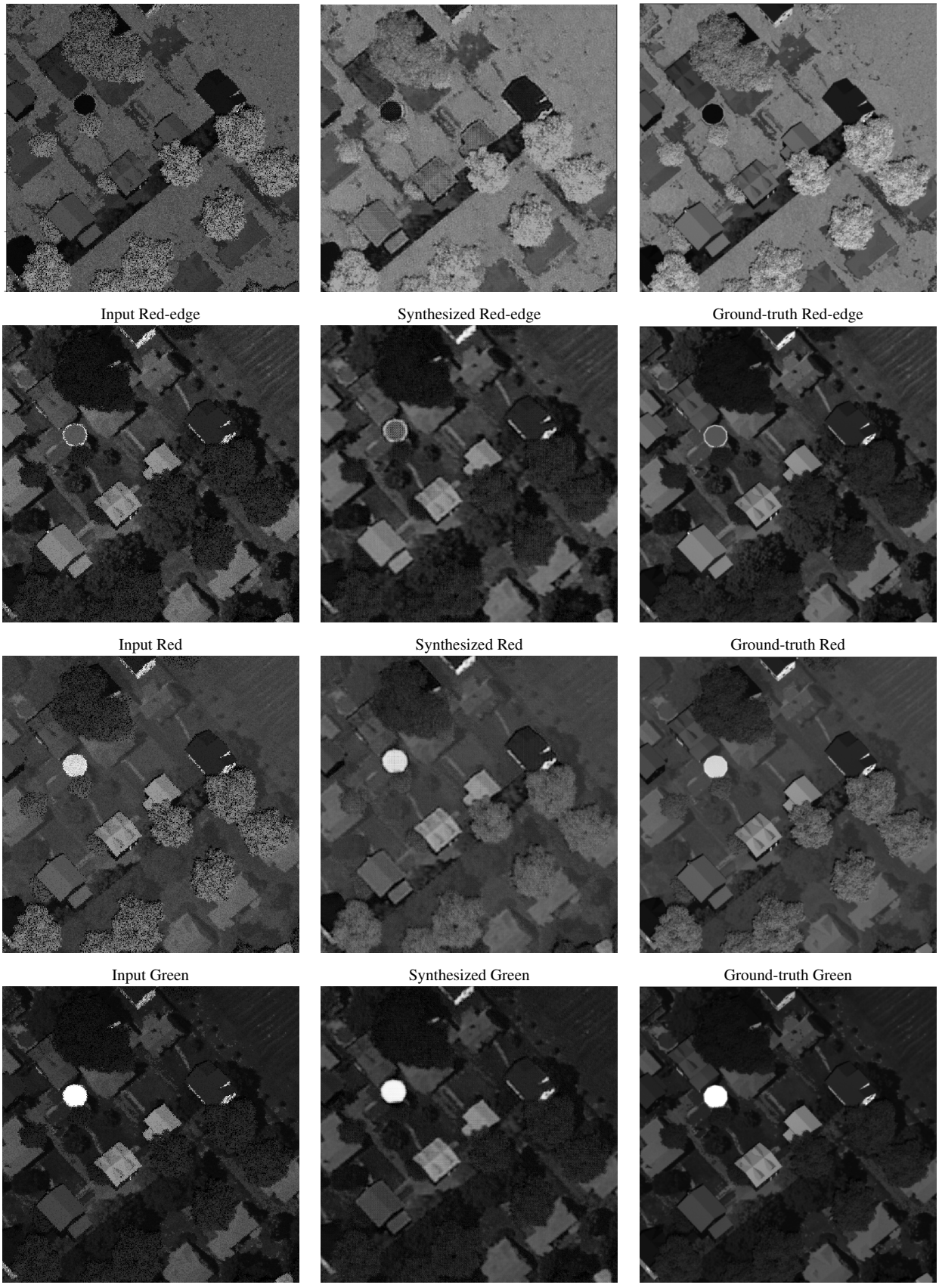


Figure 4. Composite and single channel visualizations of a synthesized multi-spectral image(center) from an input low-quality image of Dataset2 test set. Corresponding ground-truth high-quality visualizations are also shown(right).

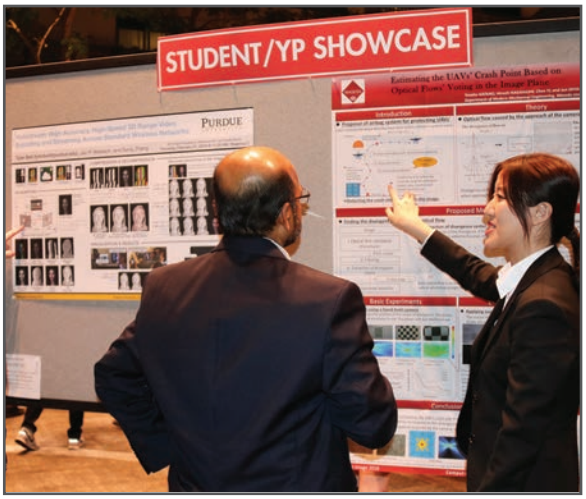
JOIN US AT THE NEXT EI!

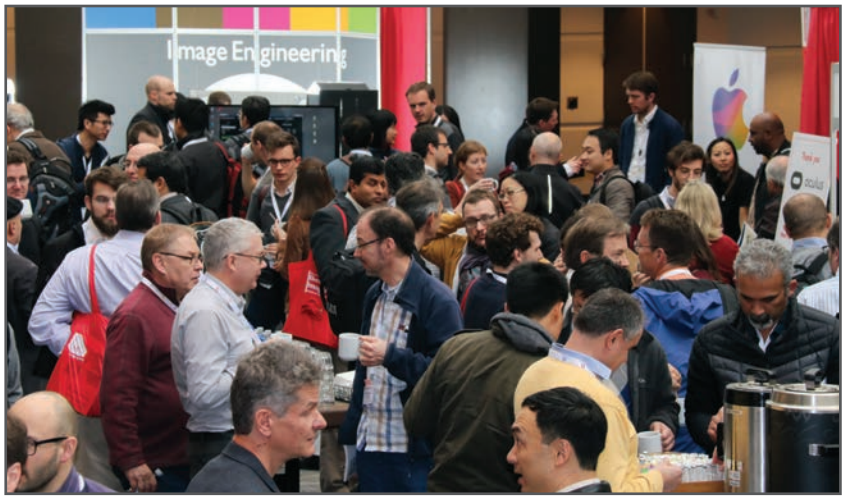
IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Imaging across applications . . . Where industry and academia meet!





• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

IS&T
imaging.org