Imperial College London

ACSE9 - IRP Project Plan

Department of Earth Science & Engineering
Author: Yuxuan Liu
Supervisor: Michael Warner
June 28, 2019

Introduction

Many problems in image processing, computer graphics, and computer vision can be posed as "translating" an input image into a corresponding output image. Just as a concept may be expressed in either English or French, a scene maybe rendered as an RGB image, a gradient field, an edge map, a semantic label map, etc. In analogy to automatic language translation, we define automatic image-to-image translation as the task of translating one possible representation of a scene into another, given sufficient training data. Traditionally, each of these tasks has been tackled with separate, special-purpose machinery, despite the fact that the setting is always the same: predict pixels from pixels.

The community has already taken significant steps in this direction, with convolutional neural nets (CNNs) becoming the common workhorse behind a wide variety of image prediction problems. However, in recent years, a novel approach - the Generative Adversarial Networks (GANs), gained great popularity in image-to-image translation (Figure 1). GANs learn a loss that tries to classify if the output image is real or fake, while simultaneously training a generative model to minimize this loss. Because GANs learn a loss that adapts to the data, they can be applied to a multitude of tasks that traditionally would require very different kinds of loss functions [1].

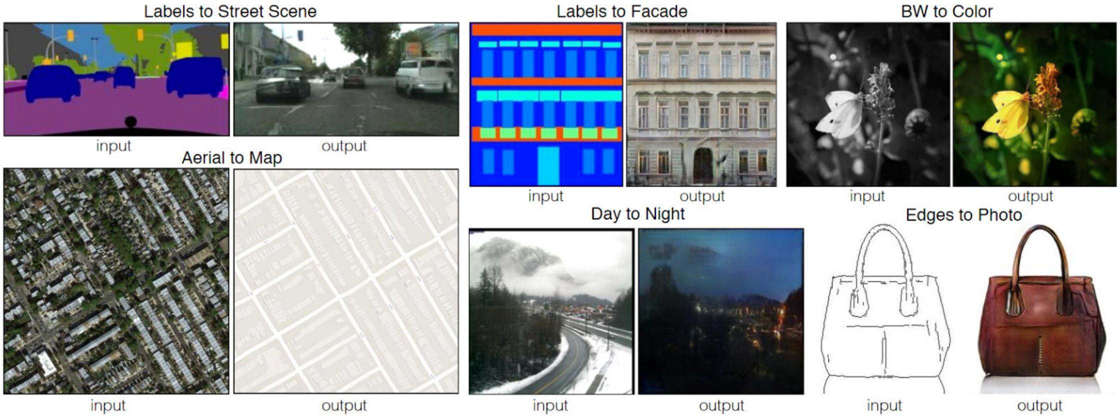


Figure 1: Image-to-Image Translation with Conditional Adversarial Networks [1]

In this project, I would explore GANs in the conditional setting. Just as GANs learn a generative model of data, conditional GANs (cGANs) learn a conditional generative model [2]. This makes cGANs suitable for image-to-image translation tasks, where we condition on an input image and generate a corresponding output image. Similar approach has been applied to seismic data performing the translations between acoustic domain and elastic domain hydrophone data to enable low-cost elastic data modelling and to remove elastic effects from real observed data before acoustic FWI (Figure 2).

GANs have been vigorously studied in the last few years and many of the techniques I will explore in this project have been previously proposed. Nonetheless, earlier papers have barely focused on ocean seismic data applications, and it has remained unclear how effective image-conditional GANs can be as a general-purpose solution for domain translation. Therefore, my primary goal is to demonstrate that on the geophysical domain translation problems, conditional GANs produce reasonable results.

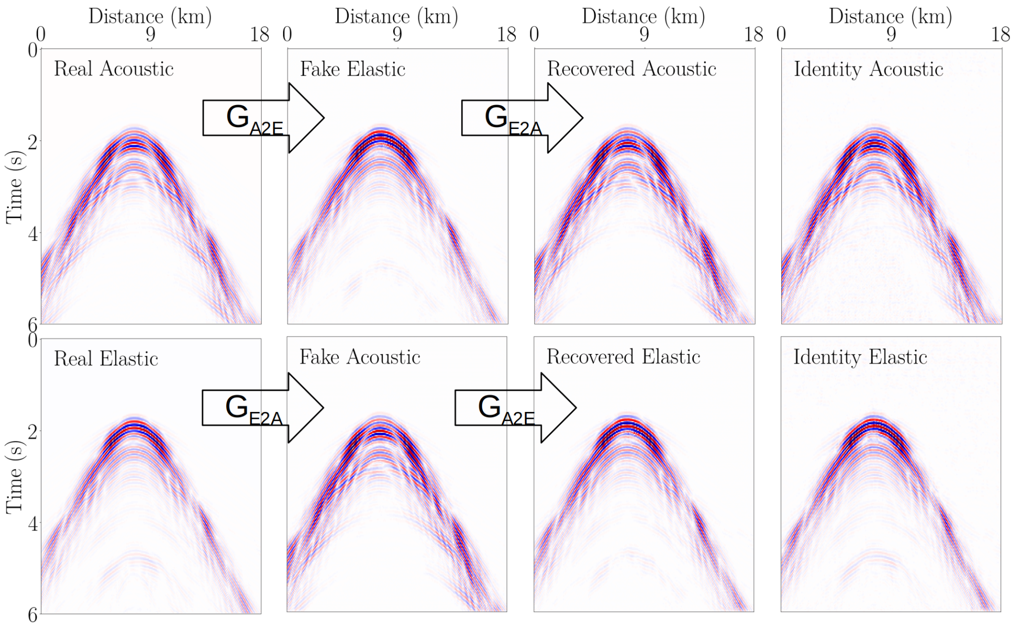


Figure 2: Seismic data translations between acoustic domain and elastic domain with cycleGAN [3]

Background

Ocean bottom acquisition is a marine seismic acquisition approaches popular in the latest decades [4]. There are two main types of ocean bottom recording: ocean bottom cable (OBC) and ocean bottom node (OBN). In both OBC and OBS acquisition most modern systems make use of four component (4C) sensors, consisting of a 3C geophone and a hydrophone [5]. The hydrophone records only the scalar pressure response and does not distinguish between up and down wavefields, whereas the geophone records the vector displacement of the seabed that is different for the up and down wavefield [6]. The source is typically an airgun fired from near the sea surface. This particular seismic acquisition method, known as Ocean Bottom Seismograph (OBS), typically yields superior images of the sub-surface than can be obtained by conventional towed-streamer data, but ocean-bottom data acquisition is significantly more expensive. This project aims to use machine learning to reduce its acquisition cost.

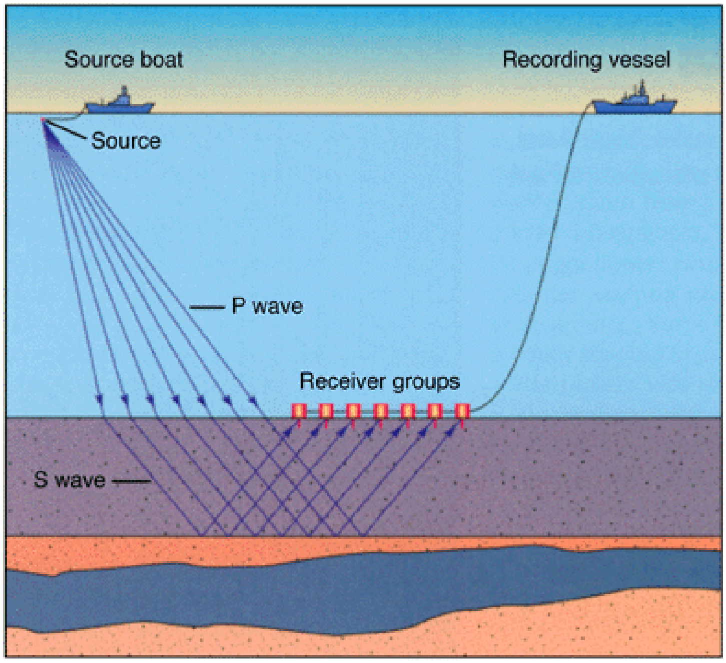


Figure 3: Ocean Bottom Seismograph [7]

OBS seismic data provide the information to obtain compressional-wave images of the subsurface. These images usually have fewer water-bottom multiples than streamer seismic data. Because of the nature of the OBS data acquisition, combining the vertical component (Z-component) of the vector wavefield and the pressure component (hydrophone), a process known as PZ summation, produces PZ seismic sections, which are ideally free of water-bottom multiples [8].

The PZ summation takes advantage of the simultaneous recording of the pressure and vertical velocity wavefields to separate the up-going and down-going wavefields and to attenuate the water bottom receiver-side peg-legs. In actual experiments, the pressure data is easy to obtain, whereas the acquisition of the vertical velocity wavefields data is relatively expensive and difficult since the geophones are buried into the hole on the seabed individually, usually by a remote operated vehicle (ROV) and with a typical receiver spacing of 100-400m [5]. Therefore, we naturally comes up with an interesting idea that using machines learning method to obtain the vertical velocity wavefields data from the pressure data without deploying all the physical geophones. That is to say, translating the seismic data from P (pressure) domain to PZ (pressure + Z-component velocity) domain via the trained model so as to greatly reduce the data acquisition cost and to make this process more efficient since we only need to collect a small portion of the original data, say 10%, to train the model, and we can get results with the same accuracy or even better.

Data set

The field data set for this study was seismic data taken from the Tommeliten Alpha field in the North Sea by OBC. Tommeliten Alpha is a gas condensate discovery located 25 km southwest of the Ekofisk field in the Norwegian North Sea, Block 1/9. The reservoir consists of two fractured chalk formations, Ekofisk and Tor, situated at the crest of a broad anticline, approximately 3000 m below the surface. A large part of the reservoir is located in a seismic-obscured area, caused by the presence of gas in the overlying section of interbedded silt and sandstone within the 1000–2000 m depth range [9].

A high-density, full-azimuth, 3D, 4C, ocean-bottom-cable survey was acquired in 2005 with the aim of improving images of the reservoir beneath the gas cloud. The data were acquired using three swaths, each composed of eight parallel cables, in water depths of around 75 m (Figure 4). The cables were 6 km in length; the inline receiver spacing was 25 m, and the crossline spacing between cables was 300 m (Figure 5). Flip-flop shooting using two airgun arrays, each of 3930 cubic inches towed at 6 m depth, was orthogonal to the cables, and used a 75 m cross-track and 25 m along-track separation (Figure 5). For each receiver swath, the shooting patch measured 10 × 12 km, and together the three patches covered a survey area of about 180 km^2. In total, the survey employed 5760 4C receivers and about 96,000 sources [10].

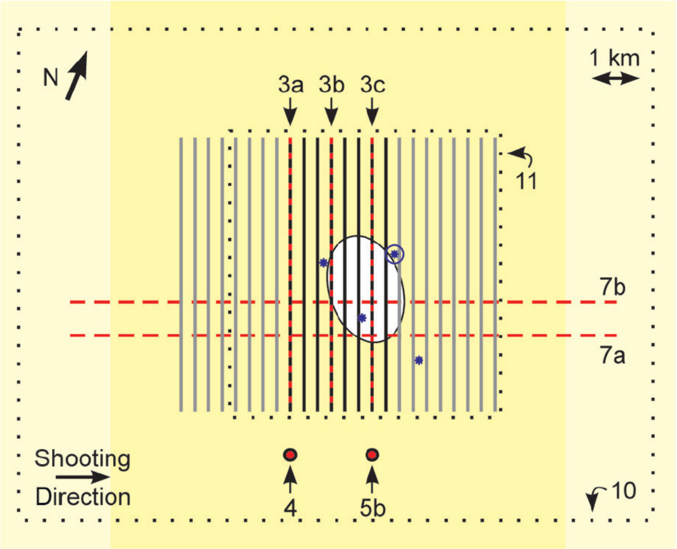


Figure 4: Experimental geometry for the OBC survey across the Tommeliten Alpha field [10].

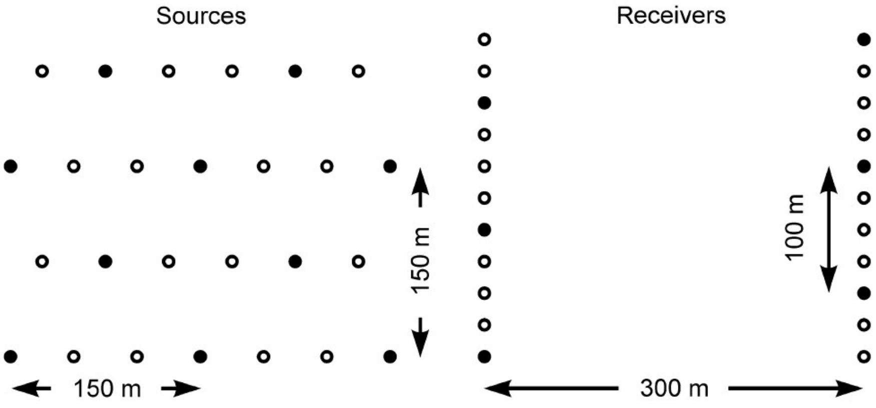


Figure 5: Detail of the source and receiver geometry. Circles show the nominal location of sources and receivers in the original data set [10].

Method

Since in this project, the raw seismic data have already been paired up in P and PZ domain manually, this explains why can we apply cGANs to domain translation of the seismic data here.

GANs are generative models that learn a mapping from random noise vector z to output image y, G: $z \rightarrow y$. In contrast, conditional GANs learn a mapping from observed image x and random noise vector z, to y, G: $\{x, z\} \rightarrow y$. The generator G is trained to produce outputs that cannot be distinguished from "real" images by an adversarially trained discriminator, D, which is trained to do as well as possible at detecting the generator's "fakes". This training procedure is diagrammed in Figure 6 [1].

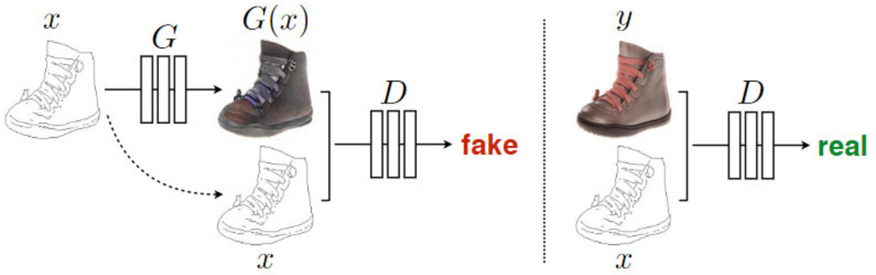


Figure 6: Training a conditional GAN to map edges→photo. Unlike an unconditional GAN, both the generator and discriminator observe the input edge map [1].

Network architectures

I would adapt the generator and discriminator architectures from those in [11]. Both generator and discriminator use modules of the form convolution-BatchNorm-ReLu [12]. Details and key features of the architecture will be discussed below.

Generator with skips

A defining feature of image-to-image translation problems is that they map a high resolution input grid to a high resolution output grid. In addition, for the problems we consider, the input and output differ in surface appearance, but both are renderings of the same underlying structure. Therefore, structure in the input is roughly aligned with structure in the output [1]. I would design the generator architecture around these considerations.

Many previous solutions [13–17] to problems in this area have used an encoder-decoder network [18]. In such a network, the input is passed through a series of layers that progressively downsample, until a bottleneck layer, at which point the process is reversed. Such a network requires that all information flow pass through all the layers, including the bottleneck. For many image translation problems, there is a great deal of low-level information shared between the input and output, and it would be desirable to shuttle this information directly across the net. For example, in the case of image colorization, the input and output share the location of prominent edges.

To give the generator a means to circumvent the bottleneck for information like this, we add skip connections, following the general shape of a "U-Net" [19]. Specifically, we add skip connections between each layer i and layer n-i, where n is the total number of layers. Each skip connection simply concatenates all channels at layer i with those at layer n-i.

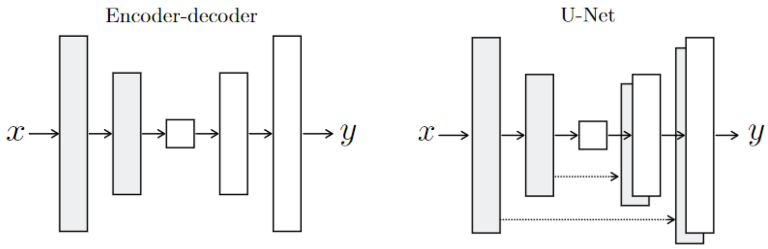


Figure 7: Two choices for the architecture of the generator. The "U-Net" [19] is an encoder-decoder with skip connections between mirrored layers in the encoder and decoder stacks [1].

Markovian discriminator (PatchGAN)

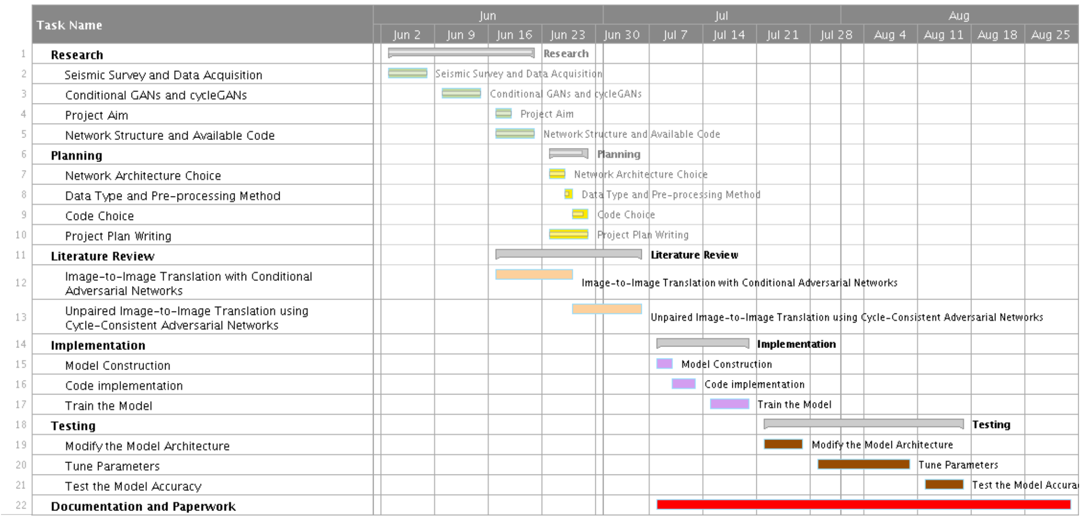
It is well known that the L2 loss – and L1, produces blurry results on image generation problems [20]. Although these losses fail to encourage high-frequency crispness, in many cases they nonetheless accurately capture the low frequencies. For problems where this is the case, we do not need an entirely new framework to enforce correctness at the low frequencies. L1 will already do.

This motivates restricting the GAN discriminator to only model high-frequency structure, relying on an L1 term to force low-frequency correctness. In order to model high-frequencies, it is sufficient to restrict our attention to the structure in local image patches. Therefore, we design a discriminator architecture – which we term a *Patch*GAN – that only penalizes structure at the scale of patches. This discriminator tries to classify if each $N \times N$ patch in an image is real or fake. We run this discriminator convolutionally across the image, averaging all responses to provide the ultimate output of D [1].

Such a discriminator effectively models the image as a Markov random field, assuming independence between pixels separated by more than a patch diameter. This connection was previously explored in [21], and is also the common assumption in models of texture [22, 23] and style [24–27]. Therefore, our PatchGAN can be understood as a form of texture/style loss.

Future Plan

The key steps in the project are to design, implement, test and refine a deep learning neural network that can recover accurate PZ-summed seismic data as output when presented with only pressure seismic data as input. The input training data, which may be used either paired or unpaired with the output data, will consist of a subset of the field dataset where both P and PZ-summed data are available for about 3,000,000 two-dimensional test examples. I expect to use about 10% of these data in training, but will test the minimum volume that is required for good performance. The detailed future plan and goals are presented in the following Gantt chart.



	Task Name
1	**Research**
2	Seismic Survey and Data Acquisition
3	Conditional GANs and cycleGANs
4	Project Aim
5	Network Structure and Available Code
6	**Planning**
7	Network Architecture Choice
8	Data Type and Pre-processing Method
9	Code Choice
10	Project Plan Writing
11	**Literature Review**
12	Image-to-Image Translation with Conditional Adversarial Networks
13	Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks
14	**Implementation**
15	Model Construction
16	Code implementation
17	Train the Model
18	**Testing**
19	Modify the Model Architecture
20	Tune Parameters
21	Test the Model Accuracy
22	**Documentation and Paperwork**

Figure 8: Timeline and future plan

References

[1] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-Image translation with conditional adversarial networks. *Computer Vision and Pattern Recognition*, November 2016.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *NIPS*, June 2014.

[3] J. Yao, L. Guasch, M. Warner, D. Davies, and A. Wild. Removing elastic effects in FWI using supervised cycled generative adversarial networks. *EAGE*, May 2019.

[4] https://wiki.seg.org/wiki/Ocean-bottom_node.

[5] Yi Wang, Sergio Grion, and Richard Bale. What comes up must have gone down the principle and application of up-down deconvolution for multiple attenuation of ocean bottom data. *RECORDER*, December 2009.

[6] https://www.slb.com/services/seismic/geophysical-processing/multicomponent/pz-processing.aspx.

[7] http://www.peakseismic.com/content/ocean-bottom-seismic.asp.

[8] http://sep.stanford.edu/sep/daniel/daniel_thesis/paper_html/node1.html.

[9] Granli, J. R., B. Arntsen, A. Sollid, and E. Hilde. Imaging through gas-filled sediments using marine shear-wave data. *Geophysics*, June 1999.

[10] Michael Warner, Andrew Ratcliffe, Tenice Nangoo, and et al. Anisotropic 3D full-waveform inversion. *Geophysics*, March 2013.

[11] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *ICLR*, January 2016.

[12] S. Ioffe and C. Szegedy. Batch normalization: Acceleratingdeep network training by reducing internal covariate shift. *ICML*, February 2015.

[13] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A.Efros. Context encoders: Feature learning by inpainting. *CVPR*, April 2016.

[14] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. *ECCV*, March 2016.

[15] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. *ECCV*, March 2016.

[16] Y. Zhou and T. L. Berg. Learning temporal transformations from time-lapse videos. *ECCV*, August 2016.

[17] D. Yoo, N. Kim, S. Park, A. S. Paek, and I. S. Kweon. Pixel-level domain transfer. *ECCV*, March 2016.

[18] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, July 2006.

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

[20] A. B. L. Larsen, S. K. Sønderby, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. *ICML*, December 2016.

[21] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. *ECCV*, April 2016.

[22] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. *ICCV*, September 1999.

[23] L. A. Gatys, A. S. Ecker, and M. Bethge. Texture synthesis using convolutional neural networks. *NIPS*, May 2015.

[24] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. *SIGGRAPH*, April 2001.

[25] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H.Salesin. Image analogies. *SIGGRAPH*, April 2001.

[26] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. *CVPR*, June 2016.

[27] C. Li and M. Wand. Combining markov random fields andconvolutional neural networks for image synthesis. *CVPR*, January 2016.