Eye For An Eye: A Deep-Learning and Analytical Method to Spatializing Stereoscopic Images

Jake Yoshinaka, Alex Dils & Leo McDonnell

Received October 02, 2024
Accepted March 18, 2025
Electronic access April 30, 2025

Many recent developments in artificial intelligence (AI) have been inspired by biological processes in the human brain. One of these processes, stereopsis, is how humans use slightly offset images from each eye as well as depth estimation to evaluate an environment. Inspired by this, we propose a hybrid architecture capable of transforming an image captured from one perspective into a corresponding image from an offset perspective. To test the effectiveness of the hybrid model, we explore three approaches to stereopsis: (1) a lone Pix2Pix generative adversarial network, (2) a hybrid method combining Pix2Pix with an analytical depth map and pixel shift, and (3) a version incorporating interpolation with the hybrid method. The version with interpolation yields the best results, achieving an SSIM score of 0.81, a 6% improvement from other methods. This architecture has the potential to enhance various fields, including 3D viewing experiences, medical imaging, and assistive technologies for individuals with impaired vision. Our work aims to advance research in vision augmentation and model design by exploring both analytical and deep learning methods.

Introduction

Recently, brain-inspired computing models, mimicking various mechanisms in the human brain, have brought great advances in our understanding of artificial intelligence (AI)[1]. For example, attention-based transformer models have led to groundbreaking innovations such as OpenAI's ChatGPT[2]. In this paper, we seek to mimic another of the brain's processes: stereopsis, the brain's ability to perceive three-dimensional depth by combining the two slightly different images received from each eye.

In the context of computing, our new approach could lead to more insight into how our brain understands the three-dimensional world so effectively. This work seeks to explore stereopsis through a series of approaches, where each approach attempts to learn the spatiality of an environment by understanding the depth and perspective from different viewpoints. To do this, we intend to find an approach that translates an image from one perspective (left eye) to another offset perspective (right eye).

A hybrid approach inspired by stereopsis would improve efficiency and accuracy by incorporating a preprocessing step that encodes depth information into the input image. Instead of relying on a model to approximate depth internally—where depth-related computations may be scattered and inaccessible—a pre-trained depth estimation model can be used. This ensures more reliable depth calculations, freeing the model to focus entirely on its primary function. Additionally, providing the model with depth-encoded inputs brings it closer to the expected output, reducing errors and improving overall performance.

Unlike attention-based models, which focus on particular parts of the input to enhance accuracy, our approach concentrates on developing a more holistic and intricate system of analysis. Stereoscopic image transformation itself relies on capturing depth and spatial details in small areas rather than finding global patterns, meaning attention mechanisms are likely unnecessary for this task and only add extra complexity.

The development of neural networks for image generation and transformation has progressed significantly. Conditional adversarial networks, which can transform images from one form to another, like turning a sketch into a detailed image, have shown great promise in image-to-image translation tasks. One well-known model for this is Pix2Pix. The Pix2Pix model, introduced by Isola et al. (2017), has been effective in generating accurate and realistic images by learning from paired datasets[3]. Research has demonstrated the utility of GANs in various applications, including medical imaging, art creation, and image enhancement[4]. This versatility highlights their potential for solving the challenge of perspective transformation in vision augmentation.

There have also been several advancements made in the analytical approaches to morphing images based on perspective. These technologies allow for smooth transitions between images, simulating changes in viewpoint and object structure, and can be applied to photographs, drawings, and rendered scenes. For example, Seitz and Dyer's paper (1996) enhanced image morphing in perspective by addressing unnatural distortions caused by differences in object pose or viewpoint[5]. Their method, called view morphing, uses projective geometry principles to pre-warp

and post-warp images, enabling realistic 3D transformations without requiring 3D shape information. The technique combines image morphing dramatic shape transformations with the ability of the view morphing approach to achieve changes in viewpoint, resulting in visually convincing 3D effects. Techniques like these have been improved on through many years in order to ensure the most convincing and realistic results from the morph.

The potential impact of this technology spans several fields. In virtual and augmented reality, it could improve the realism and immersion of 3D environments. In medical imaging, it could enhance the interpretation of complex scans by providing additional visual insights. Furthermore, assistive technologies for individuals with impaired vision could benefit from improved depth perception and spatial awareness, offering significant advancements in their quality of life.

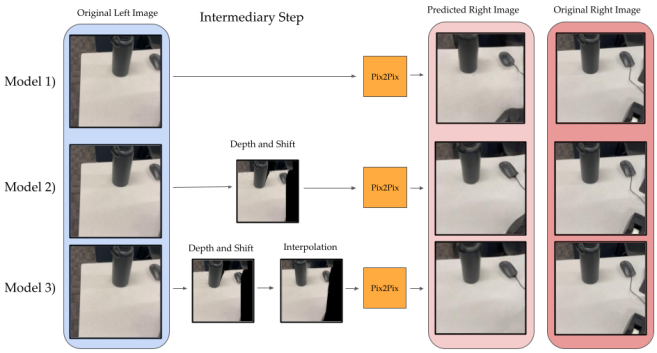

Fig. 1 Comparison of our models' inference process. Model 1) Shows the Pix2Pix method (without shifted pixels). B) Shows the analytical depth-and-shift method (without any use of Pix2Pix). C) Shows the composite model with the intermediary deformation step.

Related Work

Our work builds on advancements in image transformation and depth-aware models. While GAN-based methods like Pix2Pix are effective for image-to-image translation, they can introduce artifacts and often struggle with fine details [3]. Other GAN architectures, like CycleGAN and StyleGAN, focus on different tasks, unpaired translation and high-resolution synthesis, respectively, but do not directly handle depth-based transformations [6,7].

Analytical methods, like Seitz and Dyer's view morphing, use projective geometry to shift perspectives but tend to generate gaps and distortions [5]. More recent depth estimation models, such as MiDaS and Depth Anything v2, have improved monocular depth estimation, providing more accurate depth maps for transformations [8,9].

By combining GAN-based synthesis and depth-based analytical transformations, our composite approach mitigates the weaknesses of both methods, maintaining structural accuracy while minimizing artifacts.

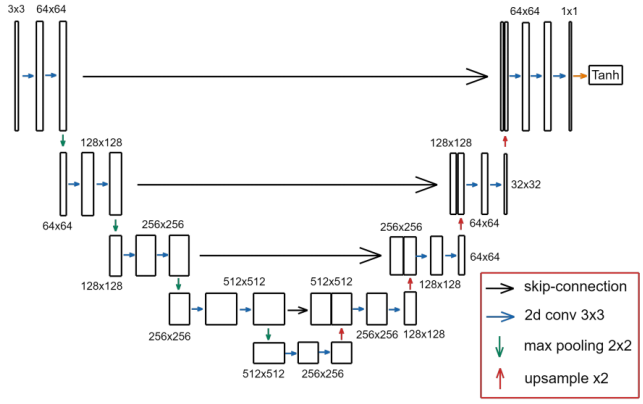

Fig. 2 Diagram of the Pix2Pix model architecture, including all of the layers involved.

Methodology

Pix2Pix Model

The Pix2Pix GAN architecture is specifically designed for image-to-image translation tasks and, hence, is applicable to this problem. Its architecture consists of two primary components: the U-Net generator and the PatchGAN discriminator [3]. The U-Net generator creates the transformed image by processing the input and recreating a version of it with changes that match the desired output (e.g., shifting perspective). The PatchGAN discriminator evaluates sections (patches) of the image to ensure that the generated image is realistic and consistent.

The combination of the U-Net generator and the PatchGAN discriminator makes the Pix2Pix architecture particularly relevant for perspective transformation tasks. The UNet's ability to learn and reconstruct detailed features from the input image ensures that the generated output closely matches the desired perspective. Simultaneously, the PatchGAN discriminator's focus on local image patches ensures that the generated images are realistic and free from artifacts. This dual-component architecture enables the effective translation of one eye's perspective to the other, achieving high-quality results in image generation. The Pix2Pix architecture, illustrated in Figure 2, combines Adversarial loss (GAN loss) and L1 (Reconstruction loss) [10].

As we used the same Pix2Pix model for each of the three experiments, we did not report GPU usage because the differences between running each experiment were minimal.

However, as noted in past literature, generative imagery can introduce inaccuracies and bias [11]. Generators are known to generate artifacts in images that are not represented in reality [12]. This can lead to limitations in the generated image's realism and accuracy.

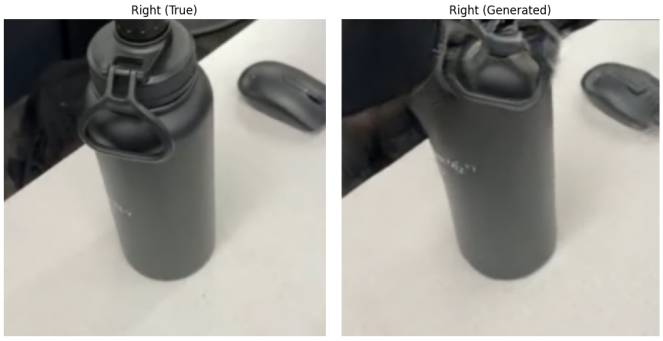

Fig. 3 Diagram comparing an example true right image and output image from the Pix2Pix. This output exemplifies an unrealistic distortion of the environment.

Depth Analytical Approach

Another approach to applying the perspective transformation is by applying a more analytical approach not based on a generator model. Instead, this approach uses a combination of a depth map (implemented as Depth-Anything-V2) and a manual shift of pixels based on the depth map[9].

We first generate a depth map of the left image using a pre-trained monocular depth estimation model. This model processes the left image and produces a depth map that encodes the distance of each pixel from the camera, represented by the pixel's brightness. Then, we apply a deformation to the left image by multiplying each pixel by its estimated depth, replicating the perspective shift required to recreate the view from the other eye. This approach intends to shift regions further away while regions closer to the camera are shifted more.

It is given that we have three images, I(x,y) (input image), O(x,y) (output image), and D(x,y) (depth map image), and each has the same dimensions, where $x \in [0, \text{image width})$, $y \in [0, \text{image height})$ and x and y are both integers.

For each pixel in the output image O(x,y), the color value of the pixel will be determined by the corresponding pixel value in the input image $I(x-\Delta x, y)$. In other words, for each (x,y) in the image dimensions:

$$O(x-\Delta x, y) = I(x,y)$$

This relation essentially results in each pixel in the input image, (x,y), undergoing a horizontal shift, Δx, to form the output image. Specifically, the horizontal shift Δx is given by:

$$\Delta x = \text{round}(k_1 + k_2 \cdot D(x,y))$$

where Δx is the horizontal shift, round() denotes rounding the input of the function to the nearest integer, $D(x,y)$ resembles the depth value at that pixel, and k_1 and k_2 are constants that control how much the depth value affects the shift.

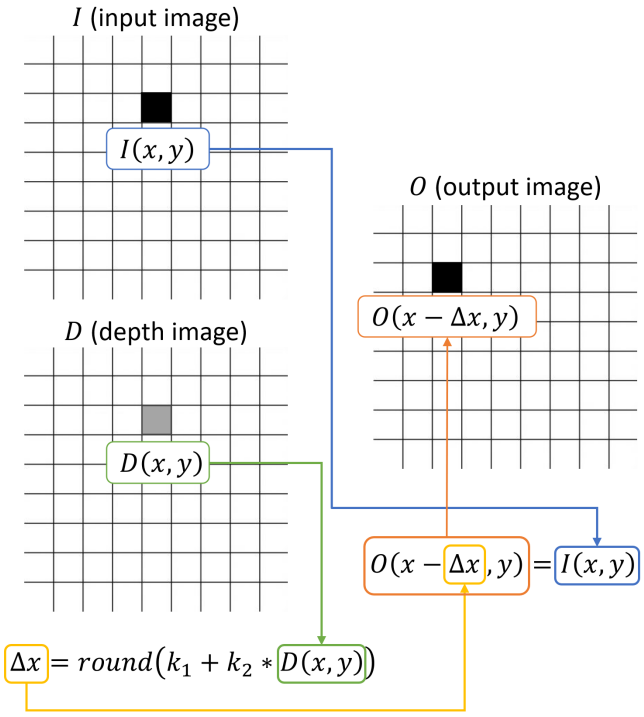

Fig. 4 Diagram demonstrating how the depth-and-shift method works on a single pixel, with mathematical equations included

The displacement of the pixel in the output image is proportional to the depth value D(x, y) and constants k_1 and k_2 , whose optimal values will be found to most realistically mimic real-life situations in order to determine the linear relationship between the shift and the depth value. With this setup, there will be blank spots in the output image where no pixels from the output image are assigned due to the shift. Specifically, if $x-\Delta x < 0$ or $x-\Delta x \geq$ width of the image, the output image at those pixels will remain blank, which means they will take on a value of zero, or black. To find the optimal values of k_1 and k_2, we determine, given the known left, right, and depth images, how much particular pixels shift from the left to the right image based on the depth value.

Consequently, we are able to find the line of best fit between the shift value and the depth value. A few images were chosen from the dataset, and a few real-world points on each image are chosen. We recorded the pixel positions of the points in both the left and right images and also the according depth value in the depth map, and to set up the linear regression, we set the horizontal difference between the left and right images as the predictor variable and the depth value as the response variable. Using linear regression, we are able to find the optimal parameters of the line of best fit.

The equation of best fit is $y = 121.03069x + 61.7295$. This means pixels with a depth value of zero will be shifted by around

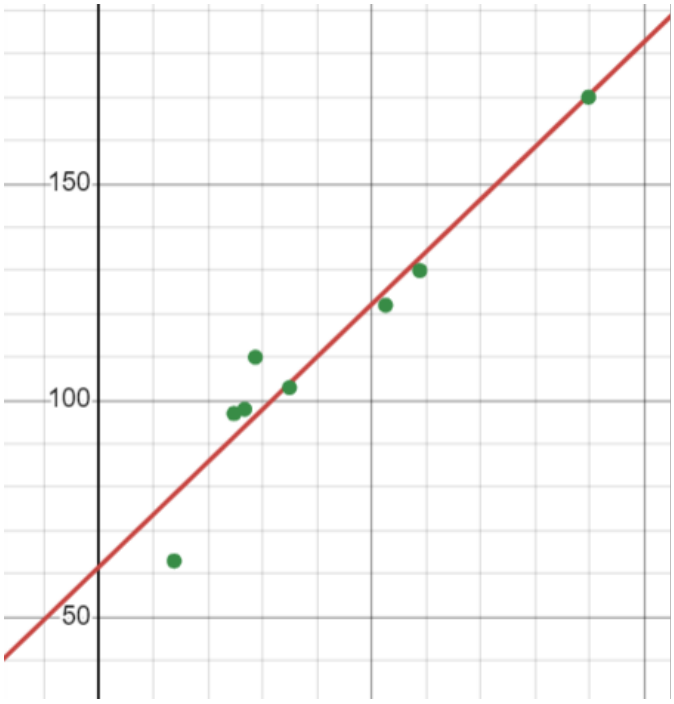

Fig. 5 Linear regression of test points to find k_1 and k_2.

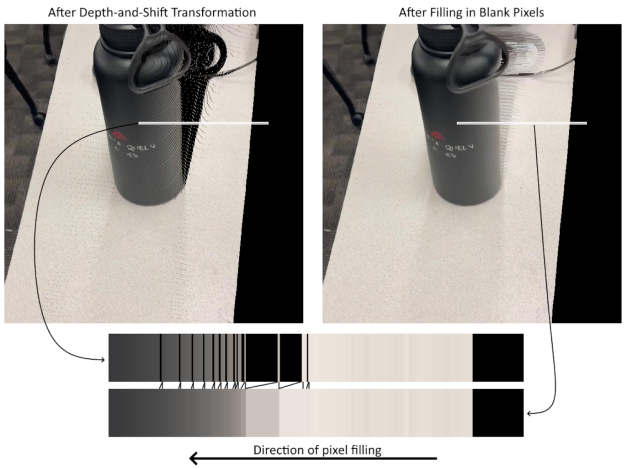

Fig. 6 Diagram of how the pixel-filling/interpolation mechanism functions. The pixel to the right of the blank area in the top strip (before pixel fill) is used to fill the blank area in the bottom strip (after the pixel fill). A row of example pixels is selected to better demonstrate this concept, though the same process is applied to all other pixels in the depth-shifted image.

62 pixels to the left, and that for every 0.1 increment in the depth value (0-1), the pixel will be shifted 12 more pixels to the left.

It is important to note that this relationship only works at a set distance away from the camera, since the DepthAnything models output relative depths and not absolute depths. For this reason, we chose to use images at approximately the same distance from the focus point, though we predict that with an appropriate absolute depth calculation model, this limitation could be unnecessary.

Now we know k_1 and k_2, we can integrate each constant into the following equation, after rounding each constant to 4 decimal places:

$$\Delta x = \text{round}(61.7295 + 121.0307 \cdot D(x,y))$$

$$O(x - \text{round}(61.7295 + 121.0307 \cdot D(x,y)), y) = I(x,y)$$

This equation serves as the mathematical model for the depth-and-shift transformation option, and the equation can be implemented in common programming languages like Python and applied as an image transformation run on the CPU, though we predict that this process could be turned into a parallel process on the GPU. We prioritized accuracy for this method over efficiency, ensuring more precise transformations, even at the cost of increased processing time.

While this method creates a shifted image that preserves the original's detail, it results in areas with missing information due to overlapping or voids created by the shift, as previously mentioned.

Interpolation

To counteract the existence of these blank areas, we can apply an additional transformation on the output of the depth-and-shift process in order to fill in the blank areas left by closer objects shifting more than further away objects. A straightforward example of such a transformation, while not perfect, would be to simply fill in each row of blank pixels using the first non-blank pixel to the right of the blank area.

Unfortunately, this transformation, due to how simple it is, does not yield perfect results. The generated depth map does not perfectly match a theoretical true depth map of the input image, which means that some pixels are shifted more or less than they realistically would. This means some pixels appear in the middle of blank areas and make the filled-in image appear more unnatural.

Additionally, we do not fill in the blank area to the far right of the image. This is because filling in this area would not be an issue of a shift of perspective, but instead the process of outpainting (filling in image information outside the image bounds using prediction), which we are not basing our model evaluations on.

However, overall, this basic interpolation algorithm will be adequate to prove the effectiveness of the depth-and-shift and composite models.

Dataset

Our data collection involved positioning two identical iPhone 12s 75mm apart, mimicking the average distance between human eyes [13]. We captured video footage at 30 frames per second (fps) in HD resolution with a 0.5 field of view (FOV). The setup simulated the distinct perspectives of the left and right eyes. The videos focused on a water bottle placed on a desk, filmed from multiple angles while maintaining the same distance from the focus point. Throughout data collection, we extracted 3960 individual frames from both videos, with each left image paired with a corresponding ground truth right image. These paired frames were used to train, validate, and test the models.

In terms of dataset organization, each input image (left eye perspective) was paired with a ground truth output image (right eye perspective), forming an input-output pair (x_i, y_i), where x_i represents the left eye image, and y_i represents the right eye image. This pairing ensures that the model learns the correlation between the two perspectives, which is crucial for tasks like stereo matching or depth estimation.

The collected data went through two preprocessing steps. First, they were cropped to 512x512. Then images were then randomly split into training (80%), validation (10%), and testing (10%) subsets. These steps ensured consistent input quality and alignment for effective model training and evaluation. The resulting dataset can be mathematically represented as:

$$D = \{(x_i, y_i) \mid x_i \in \mathbb{R}^{512\times512\times3},\ y_i \in \mathbb{R}^{512\times512\times3},\ i = 1, 2, \ldots, 3960\}$$

Each (x_i, y_i) pair is a set of images with dimensions 512×512 and 3 color channels (RGB).

Experimental Design

To test the three different methods for stereoscopic image transformation, we propose an experimental design that includes the following three experimental procedures:

1. 1. Pix2Pix: This experimental procedure involves only the Pix2Pix model being used in order to perform a perspective shift. This procedure can be mathematically modeled as:

   $$I_{\text{output}} = G(I_{\text{input}}; \theta_G)$$

   In this equation, G represents the Pix2Pix conditional GAN network, I_{input} represents the input image with dimensions 512×512 pixels, I_{output} is the output image received as the output of Pix2Pix with the same dimensions, and θ_G represents the parameters of the generator network G, which are learned during 100 epochs of training.

   This experimental procedure was tested by running the training scripts from the Pix2Pix repository with a specified input directory as the dataset's input images and the specified output directory as the dataset's output images, and the loss function was the default provided by the Pix2Pix training scripts. This procedure was run on an NVIDIA GeForce RTX 3090 GPU.

2. Pix2Pix + Depth-and-Shift: This experimental procedure involves using the output of the depth-and-shift approach as the input for the Pix2Pix model. This procedure can be mathematically modeled as:

   $$I_{\text{depth-shift}} = D_{\text{shift}}(I_{\text{input}}; \theta_D)$$

   $$I_{\text{output}} = G(I_{\text{depth-shift}}; \theta_G)$$

   In this equation, along with I_{input} (input image), I_{output} (output image), and θ_G (parameters of the Pix2Pix GAN), $D_{\text{shift}}(I_{\text{input}}; \theta_D)$ represents the output of the Depth-and-Shift model given its parameters θ_D, and $G(I_{\text{depth-shift}}; \theta_G)$ represents the output of the Pix2Pix conditional GAN network run on the depth-and-shift result.

   This experimental procedure was tested by first running the Depth Anything v2 model on all of the input images of the dataset. Then, the depth-and-shift algorithm was run on all of the input images of the dataset using the depth maps generated by the previous step. Finally, the training scripts from the Pix2Pix repository would be run with a specified input directory as the depth-and-shift output images and the specified output directory as the dataset's output images, with the loss function being set as before. The depth-and-shift procedure was run on Google Colab's standard CPU and T4 GPU, and the Pix2Pix training process was run on the same hardware as the first experimental test.

3. Pix2Pix + Depth-and-Shift + Interpolation: This experimental procedure is largely identical to the Pix2Pix + Depth-and-Shift procedure, with the sole difference being that blank pixels in the depth-and-shifted images were filled in using the interpolation mechanism described prior. This step was run using Google Colab's standard CPU directly after performing the depth-and-shift operation, and the Pix2Pix training scripts targeted the interpolated images as the input images for the model.

The base hyperparameters were mostly used when training the Pix2Pix model, and the minimal changes to these settings can be found in Appendix B. No hyperparameter tuning was used, since the Pix2Pix model and its hyperparameters are kept identical between all three methods and should not affect the results. The accuracy metrics after 100 epochs of training were captured after performing each experimental procedure in order to analyze the results.

Results

To evaluate the performance of our proposed approaches, we used four key metrics: Structural Similarity Index (SSIM), Peak Signal-to-Noise Ratio (PSNR), Mean Squared Error (MSE), and Mean Absolute Error (MAE). These metrics were chosen to provide a comprehensive assessment of the image quality and accuracy of the generated perspectives, and each metric (apart from PSNR) for each experimental procedure was calculated by running the input images in the test split and comparing these to their corresponding ground truth images in the test split of the dataset. The PSNR metric was calculated based on only the output images from the method.

- SSIM (Structural Similarity Index): SSIM measures the similarity between two images, taking into account luminance, contrast, and structure. It is particularly useful for assessing the perceptual quality of images. Higher SSIM values indicate better similarity to the ground truth.
- PSNR (Peak Signal-to-Noise Ratio): PSNR is a widely used metric for measuring the quality of reconstructed images. It is expressed in decibels (dB) and represents the ratio between the maximum possible power of a signal and the power of corrupting noise. Higher PSNR values indicate better image quality.
- MSE (Mean Squared Error): MSE quantifies the average squared difference between the generated image and the ground truth. Lower MSE values indicate that the generated image is closer to the original.
- MAE (Mean Absolute Error): MAE measures the average magnitude of errors between the generated image and the ground truth. It is a straightforward metric that provides an absolute measure of error, with lower values indicating better performance.

Table 1 Performance comparison of different models on the test set

Model	Training Dataset	SSIM	PSNR (dB)	MSE	MAE
Pix2Pix	Orignal	0.72	19.9	689.9	11.9
Composite	Combined	0.76	20.3	630.9	11.0
Model 3	Enhanced	0.81	20.6	628.0	10.3

The results of our evaluation are summarized below. Among the tested models, Model 3, trained on the Enhanced dataset, achieved the highest SSIM (0.81) and PSNR (20.6 dB), while maintaining the lowest MSE (628.0) and MAE (10.3). These results suggest that Model 3 produces the most perceptually accurate and visually similar reconstructions to the ground truth.

The Composite model, which utilizes a combined dataset, demonstrated a balance between performance and generalization, achieving competitive SSIM (0.76) and PSNR (20.3 dB) scores and slightly higher MSE (630.9) and MAE (11.0) scores. This indicates that the Composite model benefits from training on a diverse dataset, improving robustness.

The Pix2Pix model, trained on the Original dataset, achieved the lowest SSIM (0.72) and PSNR (19.9 dB), with the highest MSE (689.9) and MAE (11.9). This suggests that training solely on the original dataset may limit the model's ability to generalize and produce high-quality reconstructions.

Overall, the findings indicate that enhancing the training dataset positively impacts the model's performance, with Model 3 outperforming the others across all key metrics.

Discussion

Future Work

The potential impact of this technology spans several fields. For example, our hybrid technique could be combined with modern-day virtual and augmented reality technologies to improve the realism and immersion of 3D environments. To achieve this, two-dimensional videos of three-dimensional environments could be run frame-by-frame through our composite model to create a stereoscopic video of the environment viewable using a virtual reality headset.

This application could be extended to other fields. In medical imaging, it could enhance the interpretation of complex scans like X-rays, MRIs, or CT scans by adding another dimension of view and increasing the ease with which a healthcare provider can understand the results.

Furthermore, assistive technologies for individuals with impaired vision could benefit from improved depth perception and spatial awareness, offering significant advancements in their quality of life. We predict that, while not currently possible within the capabilities of neural implants today, a device implanted into the brain of a person with only one functioning eye could implement our composite model and provide both views to the person's optic nerves.

While our composite model itself offers many possible applications, our hybrid approach also offers many promising applications. In particular, we predict that other models that accept an image as input, such as computer vision models, could be modified to increase their accuracy either by utilizing our depth-and-shift preprocess method or by passing in a depth map of the input along with the image itself (i.e. changing a size $256x256x3$ input to a size $256x256x4$ to include the depth value at each pixel).

Ethical Considerations

The ethical considerations for our project would involve concerns about privacy and consent. Since our technology could potentially capture and generate highly detailed images of an

individual's vision, it is crucial to ensure that users provide informed consent, particularly in cases where personal data or biometric information is involved. Any data collected, especially if it pertains to individuals' personal visual experiences or health conditions, must be handled securely and anonymized to prevent misuse or unauthorized access.

Additionally, the potential for misuse of this technology raises ethical questions around the accuracy of the generated images. If the program is used to simulate or modify visual experiences, there must be transparency about its limitations and potential for errors. This could have significant consequences if the technology is applied in contexts like medical diagnoses, legal settings, or accessibility tools. Furthermore, as with most GANs, bias in the data used to train the GAN could lead to the creation of inaccurate or harmful visual representations for certain groups, reinforcing inequalities or promoting misinformation [11]. Further research must prioritize fairness and inclusivity in the training datasets and ensure rigorous testing to minimize these risks.

Conclusion

Our study explored three approaches to stereoscopic image transformation: (1) a lone Pix2Pix generative adversarial network, (2) a hybrid method combining Pix2Pix with an analytical depth map and pixel shift, and (3) a version incorporating interpolation with the hybrid method. When used individually, the Pix2Pix model demonstrated strong generalization capabilities but introduced artifacts and inconsistencies. The hybrid method improved accuracy by integrating analytical depth mapping and pixel shifting but resulted in missing pixels after the depth-and-shift, slightly reducing the overall accuracy. The final approach, which added interpolation to the hybrid method, achieved the best results by addressing these missing pixels.

By leveraging both deep learning and analytical techniques, the version with interpolation produced the best results, achieving an SSIM score of 0.81, a 6% improvement over the other methods. The deep learning aspect enabled the model to generalize transformations, while the analytical method allowed the model to maintain structural consistency. The addition of interpolation further improved the output by filling in gaps.

These findings highlight the potential of integrating artificial intelligence with analytical methods to enhance applications in a variety of fields, such as 3D viewing and visual assistive technologies.

Acknowledgments

We would like to thank our Professor, Hugh Blair, for teaching us the information and skills necessary to conduct this research, as well as help guide our research process. We would also like to thank Dr. Xiaoyang Yang, Getty George, and Yiting Wang for helping us conduct our research and educate us alongside Hugh Blair.

References

1 S. Schmidgall, R. Ziaei, J. Achterberg, L. Kirsch, S. P. Hajiseyedrazi and J. Eshraghain, *APL Machine Learning*, 2024, **2**, year.

2 A. Radford, K. Narasimhan, T. Salimans and I. Sutskever, *Improving Language Understanding by Generative Pre-Training*, Openai technical report, 2018.

3 P. Isola, J. Zhu, T. Zhou and A. A. Efros, *Image-to-Image Translation with Conditional Adversarial Networks*, https://arxiv.org/abs/1611.07004, 2017.

4 Y. Skandarani, P. Jodoin and A. Lalande, *Journal of Imaging*, 2023, **9**, 69–69.

5 S. M. Seitz and C. R. Dyer, Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, 1996, pp. 21–30.

6 J. Zhu, T. Park, P. Isola and A. A. Efros, *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*, https://arxiv.org/abs/1703.10593, 2017.

7 T. Karras, S. Laine and T. Aila, *A Style-Based Generator Architecture for Generative Adversarial Networks*, https://arxiv.org/abs/1812.04948, 2019.

8 R. Ranftl, K. Lasinger, D. Hafner, K. Schindler and V. Koltun, *Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer*, https://arxiv.org/abs/1907.01341, 2020.

9 L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng and H. Zhao, *Depth Anything v2*, https://arxiv.org/abs/2406.09414, 2024.

10 ashaba1in, *Pix2Pix: Image–to–Image Translation with Conditional Adversarial Nets [Image]*, https://github.com/ashaba1in/pix2pix, 2021.

11 Ananya, *Nature*, 2024, **627**, 722–725.

12 Z. Ahmad, Z. Jaffri, M. Chen and S. Bao, *Multimedia Tools and Applications*, 2024, **83**, 1–30.

13 H. Kobayashi and S. Kohshima, *Nature*, 1997, **387**, 767–768.

A. Performance Metrics

Performance metrics for running the Depth-Anything-v2 "vitb" (base) model on the entire dataset to generate the depth map images using the T4 GPU provided by Google Colab:

1. Setup: 16s
2. To inference all images in the dataset of 1327 images: 24m
3. 1,109MB of GPU memory was used on the T4 GPU after fully initializing the Depth-Anything-v2 model
4. 4. Average GPU usage during the operation: 81.712%, with peaks reaching 100% often (GPU usage percentage was polled every 5 seconds)

Performance metrics for running the depth-and-shift code on the entire dataset to generate shifted and filled images using the CPU provided by Google Colab:

1. To run the depth-and-shift transformation: 1 hour 41 minutes
2. 2. Average CPU usage percentage during the operation: 47.184%, with peaks reaching 100% often (CPU usage percentage was polled every 5 seconds)

B. Model Hyperparameters

Pix2Pix model hyperparameters:
————— Options —————

batch_size: 1
beta1: 0.5
checkpoints_dir: ./gan/checkpoints [default: ./checkpoints]
continue_train: False
crop_size: 512 [default: 256]
dataroot: ./gan/dataset [default: None]
dataset_mode: aligned
direction: AtoB
display_env: main
display_freq: 400
display_id: 1
display_ncols: 4
display_port: 8097
display_server: http://localhost
display_winsize: 512 [default: 256]
epoch: latest
epoch_count: 1
gan_mode: vanilla
gpu_ids: 0
init_gain: 0.02
init_type: normal
input_nc: 3
isTrain: True [default: None]
lambda_L1: 100.0
load_iter: 0 [default: 0]
load_size: 512 [default: 286]
lr: 0.0002
lr_decay_iters: 50
lr_policy: linear
max_dataset_size: inf
model: pix2pix [default: cycle_gan]
n_epochs: 100
n_epochs_decay: 100
n_layers_D: 3
name: gan1 [default: experiment_name]
ndf: 64
netD: basic
netG: unet_256
ngf: 64
no_dropout: False
no_flip: False
no_html: False
norm: batch
num_threads: 4
output_nc: 3
phase: train
pool_size: 0
preprocess: resize_and_crop
print_freq: 100
save_by_iter: False
save_epoch_freq: 5
save_latest_freq: 5000
serial_batches: False
suffix:
update_html_freq: 1000
use_wandb: False
verbose: False
wandb_project_name: CycleGAN-and-pix2pix
————— End —————-

C. Depth-and-Shift Code

The code for the depth-and-shift algorithm can be found here.