future internet

MDPI

Article

View Synthesis with Scene Recognition for Cross-View Image Localization

Uddom Lee [1], Peng Jiang [1,2], Hongyi Wu [3] and Chunsheng Xin [1,2,*]

[1] School of Cybersecurity, Old Dominion University, Norfolk, VA 23529, USA
[2] Department of ECE, Old Dominion University, Norfolk, VA 23529, USA
[3] Department of ECE, University of Arizona, Tucson, AZ 85721, USA
* Correspondence: cxin@odu.edu

Abstract: Image-based localization has been widely used for autonomous vehicles, robotics, augmented reality, etc., and this is carried out by matching a query image taken from a cell phone or vehicle dashcam to a large scale of geo-tagged reference images, such as satellite/aerial images or Google Street Views. However, the problem remains challenging due to the inconsistency between the query images and the large-scale reference datasets regarding various light and weather conditions. To tackle this issue, this work proposes a novel view synthesis framework equipped with deep generative models, which can merge the unique features from the outdated reference dataset with features from the images containing seasonal changes. Our design features a unique scheme to ensure that the synthesized images contain the important features from both reference and patch images, covering seasonable features and minimizing the gap for the image-based localization tasks. The performance evaluation shows that the proposed framework can synthesize the views in various weather and lighting conditions.

Keywords: image-based localization; deep learning; generative neural networks; style transfer

check for updates

Citation: Lee, U.; Jiang, P.; Wu, H.; Xin, C. View Synthesis with Scene Recognition for Cross-View Image Localization. *Future Internet* **2023**, *15*, 126. https://doi.org/10.3390/fi15040126

Academic Editors: Wei Yu, Weixian Liao, Fan Liang and Francesco Buccafurri

Received: 27 February 2023
Revised: 20 March 2023
Accepted: 23 March 2023
Published: 28 March 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Image-based localization, also known as cross-view geo-localization, involves finding corresponding points or regions between images taken from different viewpoints at the same location. It plays an important role in many applications, including remote sensing and localization in unmanned aerial vehicles (UAVs) [1–3], vehicle localization [4–8], wide-area augmented reality, and so on. The goal is to establish correspondences between images captured from different camera view angles, such as query images captured by a dashcam, mobile camera, etc., versus references views captured by satellite, UAV, and more, which allows the images to be aligned and compared to each other.

Recent advances in computer vision and deep learning have enabled significant improvements in cross-view image localization performance [5,8]. However, it is still a common practice to make both query and reference images look similar to improve performance [9,10]. Therefore, it is still a challenge to address the variability of image appearances caused by differences in lighting, scene structure, and weather conditions. For example, the Google Maps Platform provides public access to Google Street Views [11] and Satellite Views at given locations, which can serve as reference points for cross-view geo-localization tasks [4,5,8,9,12]. However, these photos are taken globally and uploaded to the Google Street View database at infrequent times. Many images that are used in Google Street View can be outdated, from by about a month to years, depending on the location. This is not to mention that the reference images are usually taken in a single light condition, i.e., clear light, whereas real-world users may capture a photo in any light condition. Figure 1 shows an example of when different Google Street View images are taken at different timestamps at the same location. Apparently, if a mobile

user tries to match a cell phone image taken at this location with a Google Street View captured at a completely different time or season, the matching performance would be degraded. Experiments in [12] have evaluated the performance degradation caused by the misalignment of matching pairs.

(**a**) (**b**)

Figure 1. (**a**) Google Street View taken in Queens, NYC, in November 2016; (**b**) Google Street View taken at the same location in July 2017.

To alleviate the differences between the matching pairs caused by various lighting, scene structure, and weather conditions, in this paper we propose a framework to convert outdated reference images, such as Google Street View, with a patch image that provides desired weather/light information. The patch images are collected from Flickr, an image-sharing platform, where we can obtain a set of images contributed by people who live in the area of interest. The reconstructed output images synthesize critical features from the original reference images, as well as the weather/light embedding from the patch images. To accomplish the goal of view synthetization, the proposed framework contains three parts: a patch image generator, an autoencoder-based style exchanger [13], and an auxiliary deep learning model, Places365 [14], to provide the semantic understanding of both reference and patch images.

Although several works have already discussed latent space manipulation [13,15–17], i.e., transferring styles between images, the standard approach to determine the quality of transferred images does not apply to our scenario as we need to consider whether the information is equally carried over from patch and reference images. In most cases, users do not have full control of the style transfer, as the parameters and algorithms of the deep neural network determine the transfer process. This can make it challenging to achieve the desired result, especially when transferring styles between images with significant differences in content and style. Therefore, in most cases transfer results are evaluated in the form of similarity instead of fairness. To address this issue and produce a more generative framework for the view systems task, we analyzed the different attributes extracted from patch and reference images before and after the synthesis. A fairness score was calculated from the attributes extracted from the synthesized outputs compared with the individual attributes obtained from the reference image and the patch image.

In summary, the contributions of this work are as follows:

1. We proposed a deep generative network-based view synthesis framework to address the challenges in the existing cross-view matching tasks. It was accomplished by updating the reference images, which are usually updated in an infrequent manner, with a weather/time patch. The synthesized reference view contains features from reference views, current time, and weather conditions.

2. We propose a novel evaluation metric to measure the quality of the view synthesis instead of a subjective judgment. With the assistance of an auxiliary attribute extraction network, we can effectively select the best synthesis results by comparing the attributes before and after the view synthesizing under various testing conditions.

The rest of the paper is organized as follows. Section 2 discusses the related works. Section 3 presents the main design of the proposed framework. Section 4 discusses the ways to evaluate the performance, and Section 5 concludes the paper.

2. Related Work

Image-Based Localization. Cross-View-Matching (CVM) techniques have been widely adopted in the application that requires image localization [4–6,18–20], which relies on matching query images (from ground view) to a set of geo-tagged reference images. Depending on the scenarios for the image localization, query images can be 360-panorama ground view images, dashcam images with limited view angles [12], or aerial images captured by UAVs [21]. Reference images can also be aerial/satellite views or Google Street View [12,22]. Cross-View-Matching is achieved by adopting two Siamese networks with identical network designs to extract the features from the same structure that appeared in the different views. Recent models either take advantage of polar transformation [9,23,24] or advanced Vision Transformer Networks [8] to achieve a better performance in matching stationary panorama images with satellite images in good light/weather conditions. SAFA [9] proposes a design of polar transformation to regenerate the reference image and produce a similar layout with prior knowledge of the two views. Refs. [5,8] adopted vision transformers as more effective feature extractors to align objects from two views. However, such models were incapable of matching query images captured from mobile devices such as vehicle dashcams or mobile phones. Ref. [12] adopted an Autoencoder to transfer the styles between Google Street Views and dashcam images to mitigate the performance degradation caused by the various light/weather conditions. However, such one-to-one transformation has a high computation cost, a lack of control regarding the degree of transformation, and the consideration of unqualified patch images. In additional, indoor localization [25–27] also relies on the features learnt from the images. Ref. [25] proposed the use of a clustering algorithm and dynamic compensation to enhance the accuracy of indoor positioning. Ref. [26] presented a graph-based image matching approach for indoor localization, which was based on the identification of common visual features in images of indoor environments. Ref. [27] proposed a high-accuracy recognition and localization method for moving targets in an indoor environment using binocular stereo vision.

Style Transfer (Table 1). Style transfer enables the creation of impressive and realistic synthesized images from images of different domains. DCGAN [28] is the first milestone work that proposes using Generative Adversarial Networks (GANs) to manipulate styles between images. StyleGAN and StyleGAN2 [15,29] are developed for synthesizing high-resolution photorealistic images with fine details, such as hair, fur, and textures, as well as high-level features, such as facial features and body posture. Ref. [16] addressed issues when combining GAN and the Variational Autoencoder (VAE), which had issues discarding high-frequency details but had stable training dynamics. To fix the issues of both models, an Introspective Adversarial Network (IAN), a combination of the GAN and VAE models, was proposed, which used an interpolating mask with multiscale dilated convolution blocks and orthogonal regularization to produce small quality changes on pre-existing images. Ref. [13] proposed a variant of the autoencoder architecture that used a swapping module to swap features between two input images, allowing for precise and fine-grained control over the manipulation of image attributes such as structure and texture.

Table 1. Major notations.

AE	Attribute extractor
ST	Style transfer
I_P, A_P	Patch image and the corresponding attributes
I_R, A_R	Reference image and the corresponding attributes
I_O, A_O	Output image and the corresponding attributes
M_p	Pretrained model for style transfer

3. System Design

After introducing the related techniques for style transferring and attribute extraction, in this section we describe the technical details of our system in transferring the features from a patch image to a reference image with lower updating frequency.

3.1. Framework Overview

Figure 2 illustrates the proposed framework, which is composed of the following components: patch image generator, reference image, style transfer, attribute extractor, and comparison. The fundamental idea was to generate a synthesis view for the reference images, such as Google Street View, with lower updating frequency by transferring the style from patch images, which contain the latest weather/season information from an external data source, such as Flickr [30]. Flickr is an online photo management and sharing application that covers various images with different lighting and weather from anywhere in the world. Most importantly, the large number of active users ensures that the image database is updated at a very high frequency. Once we obtain the qualified patch images by providing a time frame and a geo-location near the desired reference images, we can pass patch images and reference images to the style transfer component to produce an image that contains the features from both images. The auxiliary deep learning model contains an attribute extractor, which can recognize the scenes when the model is given images. The attribute extractor then produces a set of attributes to best describe the images. The set of attributes is then passed to the comparison component with the performance evaluator to determine the results of the attributes extracted and transferred.

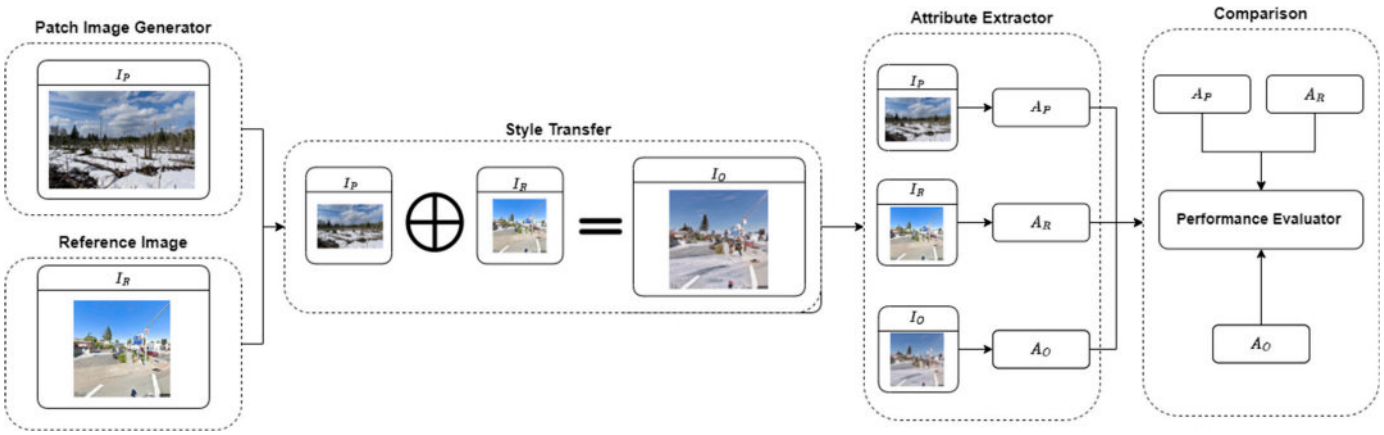

Figure 2. The overall design of the framework and a visual representation of how each component relates to the others.

3.2. Patch Image Generator

To patch the reference images such as Google Street View, which are updated irregularly, we need to select images that reflect the changes of weather and season in the area of interest by giving a timestamp and a geo-location using the Flickr API with a specific keyword, such as indoor/outdoor, forest, cloudy, etc., and the date of selection. However, such filtering may not provide the appropriate images to accurately reflect the weather and light condition if the owner of the image provided the wrong tag for the images. For

example, Figure 3 shows two search results of an attempt to search for photos uploaded near Virginia Beach on 7 February 2023, with the tag "outdoor". However, Figure 3b does not return an outdoor image due to mislabeling by the system. It is possible to be led in the wrong direction if we solely rely on keyword searching to retrieve the patch images with the latest weather and seasonable information. Therefore, a more effective approach is needed to select patch images from a pool of Flickr images returned by the search engine.

(**a**) (**b**)

Figure 3. Images were uploaded to Flickr by the author and retrieved on 7 February 2023, with the Flickr search API. (**a**) is an outdoor image tagged with a correct label. (**b**) is an indoor image but tagged as "outdoor". The copyright of both images was obtained from the city of Virginia Beach.

3.3. Attribute Extractor

In order to precisely select the patch images from the search engine to provide accurate weather and seasonal information, we selected an image classifier to extract the attributes associated with each image. This was accomplished by using Places365-CNN, a convolutional neural network (CNN) trained on a public dataset, Places365 [14], which contains 1.8 million images from 365 scene categories. Ref. [14] used a group of subjects to define the different scenic attributes which provided part of the training dataset for the CNN model. The CNN model trained on this dataset can effectively recognize the scene associated with it by showing the probability of each category. Therefore, we used the categories with high probabilities as the dominant category associated with the image. As a comparison, Table 2 shows the prediction results for both Figure 3a and Figure 3b, respectively. Clearly, we can easily remove images that do not qualify for the patch image from the search results.

Table 2. Scene category and attribute outputs from Places365-CNN.

	Figure 3a	**Figure 3b**
Type of environment	Outdoor	Indoor
Scene categories	Beach, swamp, lagoon, coast	Alley
Scene attributes	Natural light, open area, natural, trees, foliage, sunny, leaves, faraway horizon, vegetation	Plaza, shopping mall/indoor, gym/indoor, atrium/public, construction site

In general, we used A_i to denote a set of attributes extracted from the image I_i, as expressed in Equation (1), where AE represents the Attribute Extractor.

$$A_i = AE(I_i) \quad (1)$$

Additionally, the attribute extractor was also used in the performance evaluation, which measures the output attributes from the transferred images. A good transformation should contain the attributes from both patch images and reference images. A full list of attributes can be found in [14].

3.4. Style Transfer

Style transfer is a computer vision technique that involves transferring the style of one image to another. This can be achieved by using deep neural networks, such as convolutional neural networks (CNNs), to learn an image's texture and appearance and then apply this information to another image to give it a similar style. A style transfer algorithm usually involves two parts: the content and the style. The content of an image is the basic structure or shape of the image, while the style is its texture, color, and appearance. The process of style transfer involves preserving the content of one image and applying the style of another image to it, resulting in a new image that combines the content of one image with the style of another. There are several variations of style transfer, each with its own set of algorithms and techniques. Some popular style transfer algorithms include neural style transfer, which uses a deep neural network to transfer the style of one image to another, and fast style transfer, which uses a simplified neural network architecture to transfer the style of an image in real-time.

In the case of a swapping autoencoder proposed in the paper [13], the network was trained to reconstruct an image by first encoding it into a compact representation, called a latent code, and then decoding the latent code back into an image. The key difference with a swapping autoencoder is that the network is trained to swap the latent codes between two images and then decode the swapped codes to produce a new image. This allows the network to transfer the style or appearance of one image to another. There are three pre-trained models provided in [13], i.e., the mountain model, the church model, and the bedroom model, which are trained by LSUN Church, Bedroom [31], and Flickr Mountains, respectively. Figure 4 shows the view synthesis results when we attempted to transfer the weather/light information extracted from a forest image to a Google Street View Image with each pre-trained model.

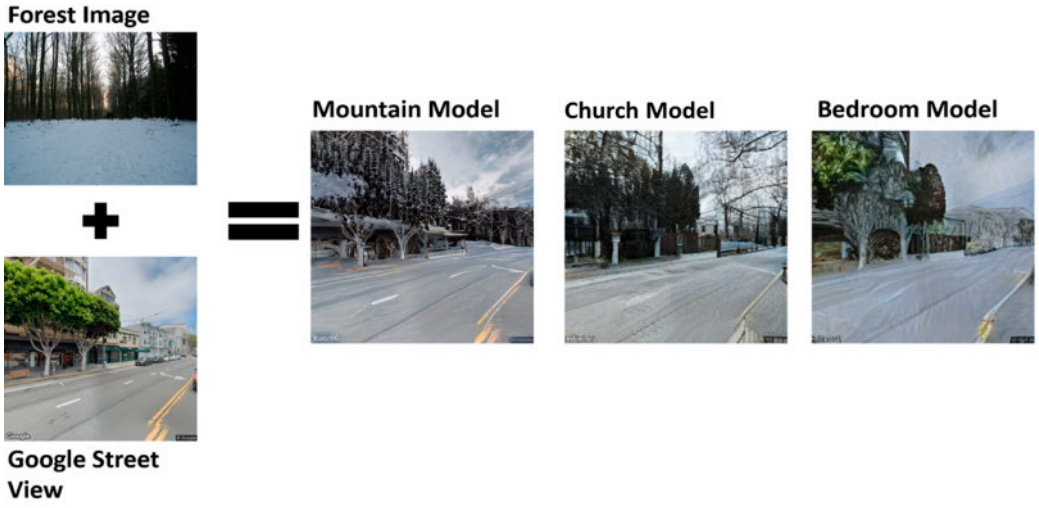

Figure 4. Forest images are combined with a reference image with three pre-trained models with swapping autoencoder [13].

The transformation can be expressed in Equation (2), where $ST(\oplus)$ is a function transferring the style from a patch image (I_P) to a reference image (I_R) with the pre-trained model M_p. I_O denotes the synthesized view after the style transformation.

$$ST(I_P \oplus I_R,\ M_p) = I_O \tag{2}$$

4. Performance Evaluation for View Synthesis

4.1. Dataset and Experimental Setup

The framework was developed with Python running on Ubuntu 20.04, and the deep learning model for style transfer and attribute extractor were implemented in PyTorch with Nvidia GPU 1080Ti. To evaluate the performance of the proposed image patching framework, we selected Google Street View as the reference image data source, which is also used [12,22] for various cross-view geo-localization tasks. Google Street View images are sampled from the GPS locations reported BDD100K [32], one of the largest self-driving datasets, with 100,000 vehicle driving trajectories from diverse locations under different weather conditions and different times of the day. We sampled one GPS location from each vehicle trajectory to represent the area of interest, then acquired the patch images from the same area by using the Flickr API accordingly.

4.2. Subjective Evaluation

Existing similarity measures, such as self-similarity distance [33] and single-image FID [34], care about the similarity in deep feature spaces based on the features that are represented in each image. However, such methods lack judgment on how good the transformation is at keeping the structure features from the reference images while adding weather/season information.

Figure 5 shows the style transfer results when we tried to patch images generated from Flickr to three Google Street View images with three pre-trained models, i.e., the mountain (first row), church (second row), and bedroom (third row) models.

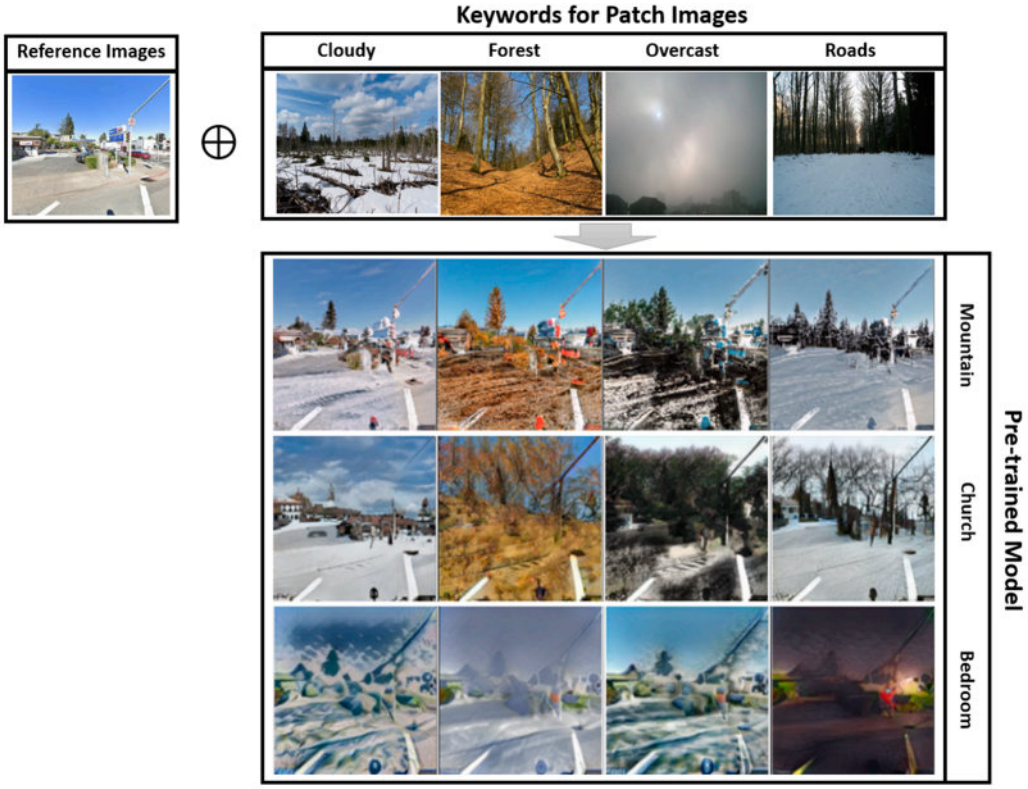

Figure 5. View synthesis performance with the same reference image (first column) on three pre-trained models, i.e., mountain (first row), church (second row), and bedroom (third row). Columns 2–5 represent transfer results from the same set of patch images filtered by keywords "Cloudy", "Forest", "Overcast", and "Roads", respectively.

Apparently, the synthesized images produced by the pre-trained models have a combination of the structure from Google Street View and weather information from the patch image, i.e., snow. However, models achieved a different level of transformation in terms

of the texture of objects in the original Google Street View image, such as roads, trees, and buildings. Specifically, the mountain and bedroom models accurately changed the color of trees and road surfaces from green to white but failed to keep the buildings on the roadsides in the transfer results. The church model keeps a building on the roadside but does not update the color of the trees. Therefore, we needed to find a numerical approach to effectively measure the quality of the view synthesis to determine which style transfer output can better reflect the changes from the inputs.

Table 3 shows a subjective evaluation of the results in Figure 5, which takes a closer look at how the reference and patch images are combined to create new images. The mountain, forest, and bedroom models created similar images. The three models kept the road and the trees within the images. These three models also had trouble keeping the buildings behind the trees in the image. Upon closer inspection of the three model images, the texture of the road and sky is different from the original Google Street View image. Additionally, the forest model image adds extra branches in the top right corner. While the images are not perfect, it can be inferred that the original Google Street View image was changed to include snow on a cloudy day.

Table 3. Subjective evaluation of the quality of the synthesized view on different pre-trained models in Figure 5. The "X" shows that the model did not retain the original structure. The "O" shows that the model kept the original structure of the reference image.

	Keywords for Filtering Patch Images			
Keyword / M_p	**Cloudy**	**Forest**	**Overcast**	**Roads**
Mountain	O	O	X	O
Church	O	O	X	X
Bedroom	X	X	X	X

4.3. Evaluation Using Jain Index

To determine if the new images created by Style Transfer were accurately and equally carrying the features from the reference image and patch image, we used the Jain Index [35] to measure the fairness of each attribute extracted from the images.

For I_P, I_R, and I_O, we obtained the top five attributes from A_P, A_R, and A_O , respectively. Since Style Transfer has three different models, as mentioned, i.e., mountain, church, and bedroom, each model had four sets of patch images with the matching keyword: cloudy, forest, overcast, and roads. From these image sets, twelve images were taken with five attributes, for a total of sixty attributes across the twelve images. If a previous attribute from A_P or A_R was seen in A_O, that attribute was added to the total amount of correct attributes carried over, which defined the rule of a ***strict*** set of matching. Only attributes that were in the previous images A_P or A_R would be tallied. Then, they would be divided by the total amount of attributes to obtain the average accuracy of the model, which is expressed in (3).

$$Accuracy = \frac{\sum_{n=1}^{N} \sum_{k=1}^{K} (A_{nk})}{N \times K} \tag{3}$$

If any attributes related to the respective image groups were seen, those attributes were added to the total number of matched attributes carried over. For example, if the A_O in the forest image set had any attributes related to forests, flora, or fauna, those attributes would be added to the correct attributes. Once the total matched attributes carried over had been tallied, they were then divided by the total amount of attributes in the image set to produce the average that the respective model would have in carrying an attribute over to A_O.

With this, we can find the average accuracy percentage in each table. $\sum_{n=1}^{N} \sum_{k=1}^{K} (A_{nk})$ represents the summation of all the matched attributes from each image in the category, where N represents the total number of images and K represents the total number of

attributes. $A_R(n)$, $A_P(n)$, and $A_O(n)$ are the attribute lists of the nth reference, patch image, and output image, respectively. A_{nk} is a binary value, which is equal to 1 if the Kth attribute of the nth image belongs to the attribute list of the nth reference image or the nth patch image.

$$O_R(n) = \sum_{k=1}^{K} G_{nk} \tag{4}$$

$$O_P(n) = \sum_{k=1}^{K} F_{nk} \tag{5}$$

$$\begin{aligned} R(n) &= Jain(O_R(n), O_P(n)) \\ &= \frac{(O_R(n)+O_P(n))^2}{2\left(O_R(n)^2+O_P(n)^2\right)^2} \end{aligned} \tag{6}$$

In Equation (4), G_{nk} is equal to 1 if Kth $A_R(n)$ belongs to $A_O(n)$. Otherwise, G_{nk} is equal to 0. In Equation (5), F_{nk} is equal to 1 if Kth $A_P(n)$ belongs to $A_O(n)$. Otherwise, F_{nk} is equal to 0.

Equation (6), which is known as the Jain Index, judges the fairness of how many attributes from A_R and A_P carried over to A_O. Fairness means the attributes of the output image came from the attributes of the reference image and patch image equally. For example, if the total attributes of A_O were four, then two attributes should have come from A_R and two attributes should have come from A_P.

Table 4 shows the average accuracy when only one attribute is selected from the image. "Sub." means subjective ruling, and "Str." means strict ruling. Table 5 shows the average accuracy when five attributes are selected from the image.

Table 4. Average accuracy when only one attribute is selected from each image.

	Cloudy		**Forest**		**Overcast**		**Roads**	
	Sub.	**Sub.**	**Sub.**	**Sub.**	**Sub.**	**Sub.**	**Sub.**	**Sub.**
Bedroom	0%	0%	0%	0%	0%	0%	0%	0%
Church	15%	15%	5%	0%	10%	10%	8.33%	8.33%
Mountain	10%	5%	5%	3.33%	8.33%	1.67%	0%	0%

Table 5. Average accuracy when five attributes are selected from each image.

	Cloudy		**Forest**		**Overcast**		**Roads**	
	Sub.	**Sub.**	**Sub.**	**Sub.**	**Sub.**	**Sub.**	**Sub.**	**Sub.**
Bedroom	5%	5%	28.33%	11.67%	1.67%	5%	5%	28.33%
Church	33.33%	31.67%	50%	28.33%	20%	33.33%	31.67%	50%
Mountain	41.67%	41.67%	40%	40%	46.67%	41.67%	41.67%	40%

In Figure 6, the data points are broken into three different colors and four different shapes. The data points in red come from the mountain model, the data points in green come from the church model, and the data points in blue come from the bedroom model. Then, the data points with the circle symbol are from the overcast image group, the data points with the star symbol are from the forest image group, the data points with the diamond symbol are from the cloudy image group, and the data points with the square symbol are from the road image group.

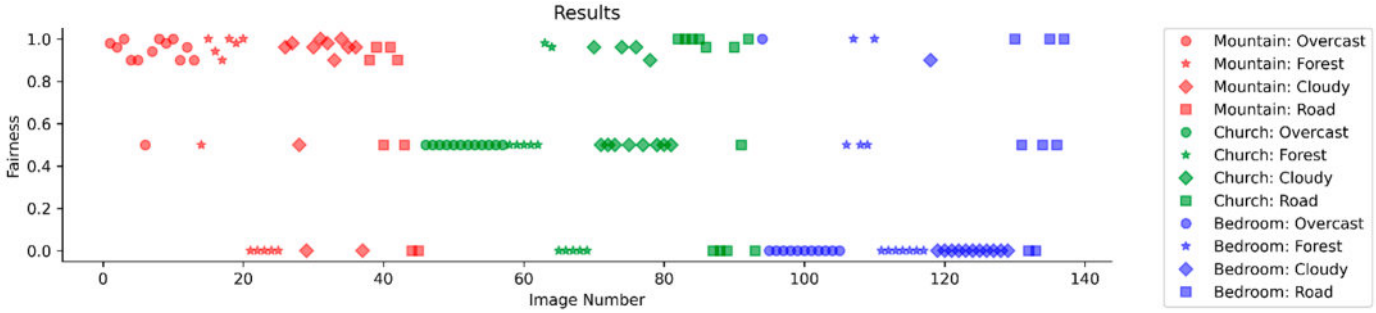

Figure 6. Jain index results of the mountain, church, and bedroom model.

From the Jain index results from all the images that were tested, the results that were the closest to 1.0 showed the image had an even distribution of attributes from both patch images from the Flickr dataset and reference images from Google Street View. Results that had a Jain index result of 0.5 or less showed an uneven distribution of attributes from both Flickr and Google Street View. Based on the results, the mountain model created more images with a Jain index closer to 1.0 than the church and bedroom models. In contrast, the bedroom model performed the worst out of the three models by having the most images with a Jain index result of 0. The church model had the most images with a Jain index of 0.5, and if the model was trained more the church model would have an opportunity to perform just as well as the mountain model.

The discussion on each pre-trained model is as follows:

1. Mountain Model. The mountain model had the most trouble with the forest image group. The mountain model also performed very well with the overcast and cloudy image group. Overall, the mountain model with more training could achieve higher Jain index results. However, since the model was mostly trained on images consisting of different mountains, the model would struggle to adapt to images with many buildings.
2. Church Model. The church model had trouble with the overcast image group. All the overcast images only had a Jain index of 0.5. With more training, the church model could achieve results similar to the mountain model. However, the church model was primarily trained on buildings, so any images that do not have many buildings would pose a problem for the church model.
3. Bedroom Model. The bedroom model struggled with all the image groups. Only a few of the images had a Jain index of 1. Moreover, even fewer images had a Jain index of 0.5, and many images had a Jain index of 0. However, this was not too surprising since the model was trained on images based on indoor scenes and features. Therefore, it was not a good fit to transfer styles for reference images that were mostly outdoor images.

Additionally, Figures 7–9 showed a 95% confidence interval graph for each individual model. Each model had image sets of overcast, forest, cloudy, and road categories. The results showed that the bedroom model had the worst performance, since the overcast and cloudy image sets had most of their fairness data points at 0, and the forest and road image sets had a wide range of data points. The mountain model performed the best compared to the church and bedroom models. The mountain model's overcast image set had most of its data points above 0.8. While the mountain model did not perform well with the forest image set showing a wide range of data points, the cloudy image set from the mountain model performed better than the forest model by having a better average of data points above 0.8.

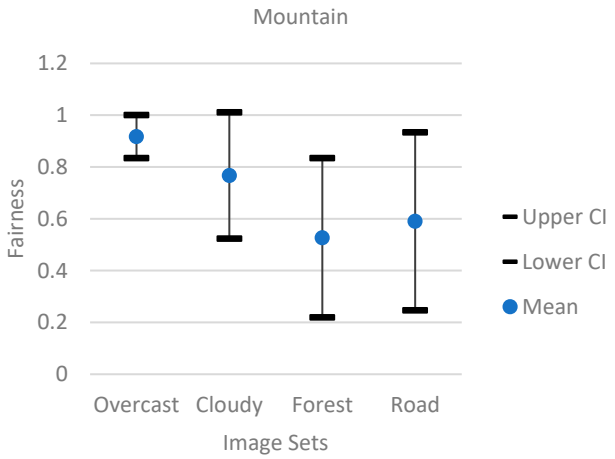

Figure 7. The 95% confidence interval graph for the mountain model.

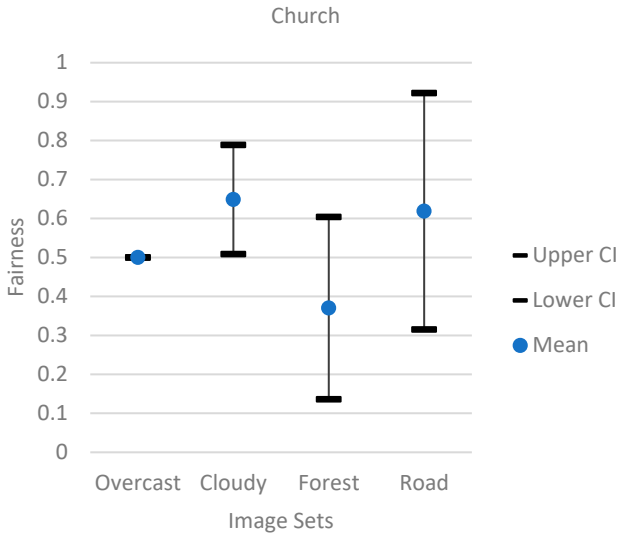

Figure 8. The 95% confidence interval graph for the church model.

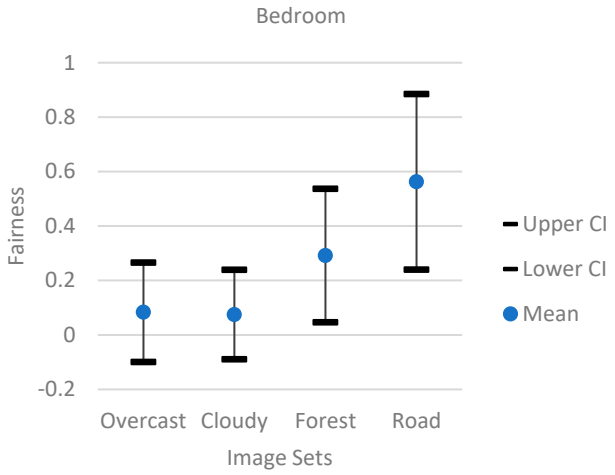

Figure 9. The 95% confidence interval graph for the bedroom model.

5. Conclusions

In this paper, we proposed a view synthesis framework to alleviate inconsistency between the query image and reference images in the existing cross-view geo-localization system. Our framework was composed of a patch image generator, attribute extractor,

and style transfer to update the reference images with a lower update frequency, with the latest weather/season information within the area of interest. A Jain fairness index was used to evaluate the performance of view synthesis outputs to determine if the attributes from both the patch image and the reference image were carried over equally without a significant bias. We evaluated the extensive performance with various models, and the experiment results indicate that it can effectively update reference images under complex weather conditions.

Author Contributions: Conceptualization, U.L. and P.J.; methodology U.L. and P.J.; software, U.L.; validation, U.L., P.J. and C.X.; formal analysis, U.L. and P.J.; investigation, U.L. and P.J.; resources, P.J. and C.X.; data curation, P.J.; writing—original draft preparation, U.L.; writing—review and editing, P.J.; visualization, U.L.; supervision, H.W. and C.X.; project administration, H.W. and C.X.; funding acquisition, H.W. and C.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the National Science Foundation under Grants CNS-2120279, CNS-1950704, CNS-2244902, CNS-2245250, and DUE-1742309, the National Security Agency under Grants H98230-22-1-0275, H98230-21-1-0165, and H98230-21-1-0278, the Air Force Research Lab under Grant FA8750-19-3-1000, and the Commonwealth Cyber Initiative.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to the data containing information that could compromise the privacy of research participants.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ding, L.; Zhou, J.; Meng, L.; Long, Z. A Practical Cross-View Image Matching Method Between UAV and Satellite for UAV-Based Geo-Localization. *Remote Sens.* **2020**, *13*, 47. [CrossRef]
2. Zhuang, J.; Dai, M.; Chen, X.; Zheng, E. A Faster and More Effective Cross-View Matching Method of UAV and Satellite Images for UAV Geolocalization. *Remote Sens.* **2021**, *13*, 3979. [CrossRef]
3. Shetty, A.; Gao, G.X. UAV Pose Estimation Using Cross-View Geolocalization with Satellite Imagery. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019.
4. Hu, S.; Feng, M.; Nguyen, R.M.; Hee Lee, G. CVM-Net: Cross-View Matching Network for Image-Based Ground-to-Aerial Geo-Localization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
5. Lu, X.; Zhu, Y. Cross-View Geo-Localization with Layer-to-Layer Transformer. In Proceedings of the Thirty-Fifth Conference on Neural Information Processing Systems, Online, 7–10 December 2021.
6. Tian, Y.; Chen, C.; Shah, M. Cross-View Image Matching for Geo-Localization in Urban Environments. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
7. Xia, Z.; Booij, O.; Manfredi, M.; Kooij, J.F. Cross-View Matching for Vehicle Localization by Learning Geographically Local Representations. *IEEE Robot. Autom. Lett.* **2021**, *6*, 5921–5928. [CrossRef]
8. Zhu, S.; Shah, M.; Chen, C. TransGeo: Transformer Is All You Need for Cross-view Image Geo-localization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
9. Shi, Y.; Liu, L.; Yu, X.; Li, H. Spatial-Aware Feature Aggregation for Image Based Cross-View Geo-Localization. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 10090–10100.
10. Tian, X.; Shao, J.; Ouyang, D.; Shen, H.T. UAV-Satellite View Synthesis for Cross-View Geo-Localization. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *32*, 4804–4815. [CrossRef]
11. Anguelov, D.; Dulong, C.; Filip, D.; Frueh, C.; Lafon, S.; Lyon, R.; Ogale, A.; Vincent, L.; Weaver, J. Google Street View: Capturing the World at Street Level. *Computer* **2010**, *43*, 32–38. [CrossRef]
12. Jiang, P.; Wu, H.; Zhao, Y.; Zhao, D.; Xin, C. SEEK: Detecting GPS Spoofing via a Sequential Dashcam-Based Vehicle Localization Framework. In Proceedings of the IEEE International Conference on Pervasive Computing and Communications, Atlanta, GA, USA, 13–17 March 2023.
13. Park, T.; Zhu, J.-Y.; Wang, O.; Lu, J.; Shechtman, E.; Efros, A.; Zhang, R. Swapping Autoencoder for Deep Image Manipulation. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 7198–7211.

14. Zhou, B.; Lapedriza, A.; Khosla, A.; Oliva, A.; Torralba, A. Places: A 10 Million Image Database for Scene Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 1452–1464. [CrossRef] [PubMed]
15. Abdal, R.; Qin, Y.; Wonka, P. Image2stylegan: How to Embed Images into the Stylegan Latent Space? In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 29 October 2019.
16. Brock, A.; Lim, T.; Ritchie, J.M.; Weston, N. Neural Photo Editing with Introspective Adversarial Networks. *arXiv* **2016**, arXiv:1609.07093.
17. Yeh, R.A.; Chen, C.; Yian Lim, T.; Schwing, A.G.; Hasegawa-Johnson, M.; Do, M.N. Semantic Image Inpainting with Deep Generative Models. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
18. Zhu, S.; Yang, T.; Chen, C. Revisiting Street-to-Aerial View Image Geo-Localization and Orientation Estimation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Nashville, TN, USA, 20–25 June 2021.
19. Cai, S.; Guo, Y.; Khan, S.; Hu, J.; Wen, G. Ground-to-Aerial Image Geo-Localization with a Hard Exemplar Reweighting Triplet Loss. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 29 October 2019.
20. Liu, L.; Li, H. Lending Orientation to Neural Networks for Cross-View Geo-Localization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
21. Xue, N.; Niu, L.; Hong, X.; Li, Z.; Hoffaeller, L.; Pöpper, C. DeepSIM: GPS Spoofing Detection on UAVs Using Satellite Imagery Matching. In Proceedings of the Annual Computer Security Applications Conference, Online, 7–11 December 2020.
22. Regmi, K.; Shah, M. Video Geo-Localization Employing Geo-Temporal Feature Learning and Gps Trajectory Smoothing. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021.
23. Shi, Y.; Yu, X.; Campbell, D.; Li, H. Where am I Looking at? Joint Location and Orientation Estimation by Cross-View Matching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
24. Toker, A.; Zhou, Q.; Maximov, M.; Leal-Taixé, L. Coming Down to Earth: Satellite-to-Street View Synthesis for Geo-Localization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021.
25. Bi, J.; Huang, L.; Cao, H.; Yao, G.; Sang, W.; Zhen, J.; Liu, Y. Improved Indoor Fingerprinting Localization Method Using Clustering Algorithm and Dynamic Compensation. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 613. [CrossRef]
26. Manzo, M. Graph-Based Image Matching for Indoor Localization. *Mach. Learn Knowl. Extr.* **2019**, *1*, 46. [CrossRef]
27. Ding, J.; Yan, Z.; We, X. High-Accuracy Recognition and Localization of Moving Targets in an Indoor Environment Using Binocular Stereo Vision. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 234. [CrossRef]
28. Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv* **2015**, arXiv:1511.06434.
29. Viazovetskyi, Y.; Ivashkin, V.; Kashin, E. Stylegan2 Distillation for Feed-Forward Image Manipulation. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020.
30. Flickr. Available online: https://www.flickr.com/photos/tags/flicker/ (accessed on 7 February 2023).
31. Yu, F.; Seff, A.; Zhang, Y.; Song, S.; Funkhouser, T.; Xiao, J. Lsun: Construction of a Large-Scale Image Dataset Using Deep Learning with Humans in the Loop. *arXiv* **2015**, arXiv:1506.03365.
32. Yu, F.; Xian, W.; Chen, Y.; Liu, F.; Liao, M.; Madhavan, V.; Darrell, T. BDD100K: A Diverse Driving Video Database with Scalable Annotation Tooling. *arXiv* **2018**, arXiv:1805.04687.
33. Kolkin, N.; Salavon, J.; Shakhnarovich, G. Style Transfer by Relaxed Optimal Transport And Self-Similarity. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
34. Shaham, T.R.; Dekel, T.; Michaeli, T. Singan: Learning a Generative Model from a Single Natural Image. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 29 October 2019.
35. Jain, R.K.; Chiu, D.-M.W.; Hawe, W.R. *A Quantitative Measure of Fairness and Discrimination*; Eastern Research Laboratory, Digital Equipment Corporation: Hudson, MA, USA, 1984; pp. 1–38.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.