Synopsis

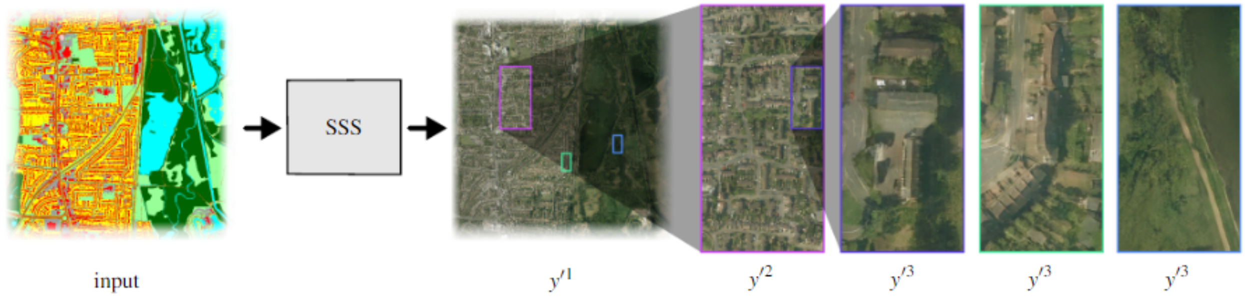

Figure 1: *Our system, SSS, takes input vector cartographic data (left), and generates realistic and seamless high-resolution satellite images (right; the image y'^3 is 8,192 pixels wide, approximately 67 megapixels) which are continuous over space and scale. Given the input, SSS generates massive spatially seamless images over a range of scales (y'^1, y'^2, and y'^3) by blending the results of lower resolution networks.*

Abstract:

We introduce Seamless Satellite-image Synthesis (SSS), a novel neural architecture to create scale-and-space continuous satellite textures from cartographic data. While 2D map data is cheap and easily synthesized, accurate satellite imagery is expensive and often unavailable or out of date. Our approach generates seamless textures over arbitrarily large spatial extents which are consistent through scale-space. To overcome tile size limitations in image-to-image translation approaches, SSS learns to remove seams between tiled images in a semantically meaningful manner. Scale-space continuity is achieved by a hierarchy of networks conditioned on style and cartographic data. Our qualitative and quantitative evaluations show that our system improves over the state-of-the-art in several key areas. We show applications to texturing procedurally generated maps and interactive satellite image manipulation.

Introduction:

Satellite images have revolutionized the way we visualize our environment. Massive, high-resolution, satellite images are essential to many modern endeavors including realistic virtual environments and regional- or city-planning. However, these images can be prohibitively expensive, out of date, or of low quality. Further, continuous satellite images covering large areas are often unavailable due to limited sensor resolution or orbital mechanics. In contrast, cartographic (map) information is frequently accessible and available over massive continuous scales; further, it can be easily manually edited or synthesized. In this paper we address the problem of generating satellite images from map data at arbitrarily large resolutions.

Convolutional neural networks are state of the art for image toimage translation; such networks can translate from cartographic images to satellite images. However, these networks are unable to generate outputs of arbitrary areas – the resolution of these synthetic satellite image tiles remains limited by available memory and compute. Arranging many such small tiles to cover a large map results in inconsistencies (seams) between the tiles. Further, if a larger number of tiles are joined together and viewed at multiple scales, the result lacks continuity and style coordination.

Inspired by convolutional image-to-image translation systems such as Pix2Pix [IZZE17] and SPADE [PLWZ19], Seamless Satellite-image Synthesis (SSS, Figure 1) creates satellite image tiles by conditioning generation on cartographic data. These tiles are of fixed resolution (256 ×256 pixels in our implementation) and exhibit good continuity internally, but would contain spatial and scale artifacts if naïvely tiled over a larger area. Spatial discontinuities between adjacent tiles are caused by a lack of coordination between network evaluations, resulting in lack of continuity for properties not determined by the cartographic input (e.g., color of building roof or species of tree). We introduce a network which removes such spatial discontinuities by leveraging learned semantic domain knowledge as well as the cartographic data. Scale discontinuities between a larger set of synthetic satellite tiles arise from unnatural style variations over larger areas (e.g., distribution of cars throughout a city or different camera sensors). To address this, SSS uses a hierarchy of networks synchronized by color guidance.

The SSS system contains three components – two convolutional neural networks to generate tiles of different resolutions (map2sat), and remove seams (seam2cont) between those tiles, as well as an interface allowing the generated textures to be explored interactively.

SSS contributes:

•A neural approach to generate massive scale-and-space continuous images conditioned on cartographic data.

•A novel architecture to train and evaluate image-to-image generation networks at high resolutions with reduced memory use.

•An application to texture procedurally generated cartographic maps – allowing the synthesis of textures for entirely novel areas following user style guidance.

•An application to interactive exploration of endless satellite images generated on-the-fly from cartographic data.

The textures generated by SSS have many applications, including interactive city planning, virtual environments, and procedural modeling. During city planning, the textures may help stakeholders (e.g., homeowners, planners, and motorists) understand proposed developments. For example, sit-down sessions in which stakeholders interactively edit a map and are shown birds-eye-view (satellite) “photos” within seconds. Further, virtual environments may be constructed using SSS’s output (e.g., a flight simulator) to texture massive terrains on-the-fly, or the streets and open areas of urban procedural models [PM01].

Objective:

High resolution imaging satellites have limited swath and images from adjacent passes are stitched together, However, theme images may have different illumination and atmospheric conditions leading to visible differences at the border of images. The goal of this problem statement is to develop a fully automated algorithm to automatically stitch adjacent (multiple) satellite images without human intervention to create a seamless mosaic. The algorithm needs to be fast enough for on-the-fly stitching for web based visualization.

Scope:

The goal of this problem statement is to develop a fully automated algorithm to automatically stitch adjacent (multiple) satellite images without human intervention to create a seamless mosaic. The algorithm needs to be fast enough for on-the-fly stitching for web based visualization.

Study of Existing Systems:

J. Zhu & T. Kelly / Seamless Satellite-image Synthesis

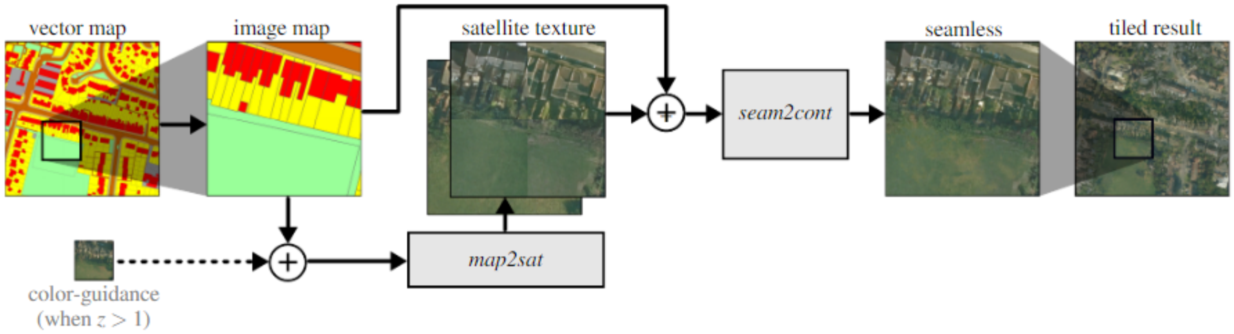

Figure 2: *SSS uses a hierarchy of networks for different scales. A pair of networks is shown above, which process input cartographic maps to large seamless images at a single scale level, z. First, a vector map of arbitrary extent is processed to create an image tile of map data. The* map2sat *network then performs a translation to the satellite texture domain, before the* seam2cont *network removes visible tilling artifacts to create a seamless image and the result is combined with neighboring tiles to create an image of arbitrary resolution (right). Subsequent, smaller scale,* map2sat *networks use previous outputs as scale-guidance (bottom left).*

Texture synthesis is the generation of texture images from exemplars or lower dimension representations. We refer the reader to Wei et al. [WLKT09] for an overview. Non-neural approaches from the AI Winter create images with local and stationary properties using techniques such as Markov Random Fields to generate pixels [EL99,WL00] or patches [LLX∗01,EF01] from an exemplar image. Variations on these themes generate larger, more varied textures [DSB∗12]. Procedural languages for texturing [MZWVG07] are well adapted for urban domains and can be used to guide synthesis [GAD∗20].

With the advent of Convolutional Neural Networks (CNNs),

texture synthesis has increasingly become data-driven, creating images by gradient descent in feature space [GEB15] or with Generative Adversarial Networks (GANs) [ZZB∗18]. Another approach to texture generation is to transform an input image to a different style or domain while retaining the structure. Such image-to-image translation is implemented by the influential UNet architecture [RFB15] which passes the input through an “hourglass” shaped network of convolutional layers. This allows the network to aggregate and coordinate features within a bottleneck. UNets have been applied to generate satellite images [IZZE17] from map data, however the resolution of the output tiles was limited by available memory. Image-to-image networks typically disentangle the structure from style to enable style transfer [GEB16]. These have been applied to generate terrains [GDG∗17] and within chained sequences [KGS∗18] for architectural texture pipelines.

Label-to-image translation is a variation of image-to-image translation; the input is a label-map with each pixel having a specific class, such as cartographic data containing, streets, buildings, and vegetation. Again, we note the application of hourglass networks [WLZ∗18]. We build on Semantic Image Synthesis with Spatially-Adaptive Normalization (SPADE) [PLWZ19] which uses a simpler V-shaped architecture with a bottleneck at the start, rather than center, of the network. From a latent code, successive SPADE Residual Blocks generate feature-maps of increasing resolution. Within these blocks, a SPADE unit introduces learned per-label style information using Spatially-Adaptive Normalization. Texture blending or inpainting approaches give a way to combine multiple smaller images into a larger one. These may match small image patches [LLX∗01,EF01,BSFG09], or minimize a function over a grid of pixels using seam energy minimization [AS07,KSE∗03,RKB04], exemplars [CPT03], or a Poisson formulation [PGB03]. Although such local approaches produce excellent pixel-level results, they do not contain the higher, semantic-level knowledge to blend textures and images with complex structures. We may overcome this limitation with user input [ADA∗04], large datasets [HE07], and distributions learned by neural architectures [ZZL19,ZWS20,BSP∗19]. These learn distributions for the blended area using techniques such as guided Poisson constraints [WZZH19] and symmetry [DCX∗18]. Domain specific image blending has been studied for domains such as microscopy [LCH∗15], façade texturing [KGS∗18], and satellite generation. TileGAN [FAW19] searches a library of latent encodings of adjacent micro-tiles to create seamless images, rather than learning a domain specific blending function.In contrast to relatively recent satellite images, applications of cartography date back to antiquity. Today, there are high-quality, publicly available maps of much of the world [Fou21,Ser21]. We may also generate maps with a variety of techniques, including procedural systems; see Vanegas et al. [VAW∗10] for an overview. In particular, we may use rule and parameter controlled geometric algorithms to recreate the distribution of streets [CEW∗08], buildings [PM01], and city parcels [VKW∗12]. These procedural systems can generate massive quantities of 3D city geometry from random seeds; such geometry can be used to generate cartographic maps by projecting the 3D meshes to 2D images and applying a constant color for each category (e.g., red buildings). Procedurally generated cartography has the advantage that it can be easily edited with a user-interface or created from databases.

Project Description:

We propose a novel method for map-to-satellite-image translation over arbitrarily large areas. Two neural networks work together to first generate satellite image tiles from map data, and then remove seams and discontinuities from these images. To generate an area, SSS (Figure 2) first generates two overlapping sets of satellite image tiles with the map2sat network. Then the seam2cont network removes the discontinuities and seams between adjacent tiles by learning to generate a mask to blend the overlapping satellite images. Joining together such tiles results in a single seamless image with a large resolution and scale.To achieve our goals, we solve problems such as the training of a mask generator to remove seams while realistically blending tiles, the coordination of networks over different areas to create continuous results in scale-space, and reducing the memory usage for large image synthesis networks.

Methodology:

Evaluating the quality of composite synthesized images for neural networks is a difficult problem; there is no canonical method to assess the performance of image stitching. Traditional edge detection technology, such as Sobel filter detection and Canny edge detection, can be used for seams detection but are not optimized for tile-based systems where the seam locations can be anticipated. In addition, these algorithms output images with edges rather than comparable values.

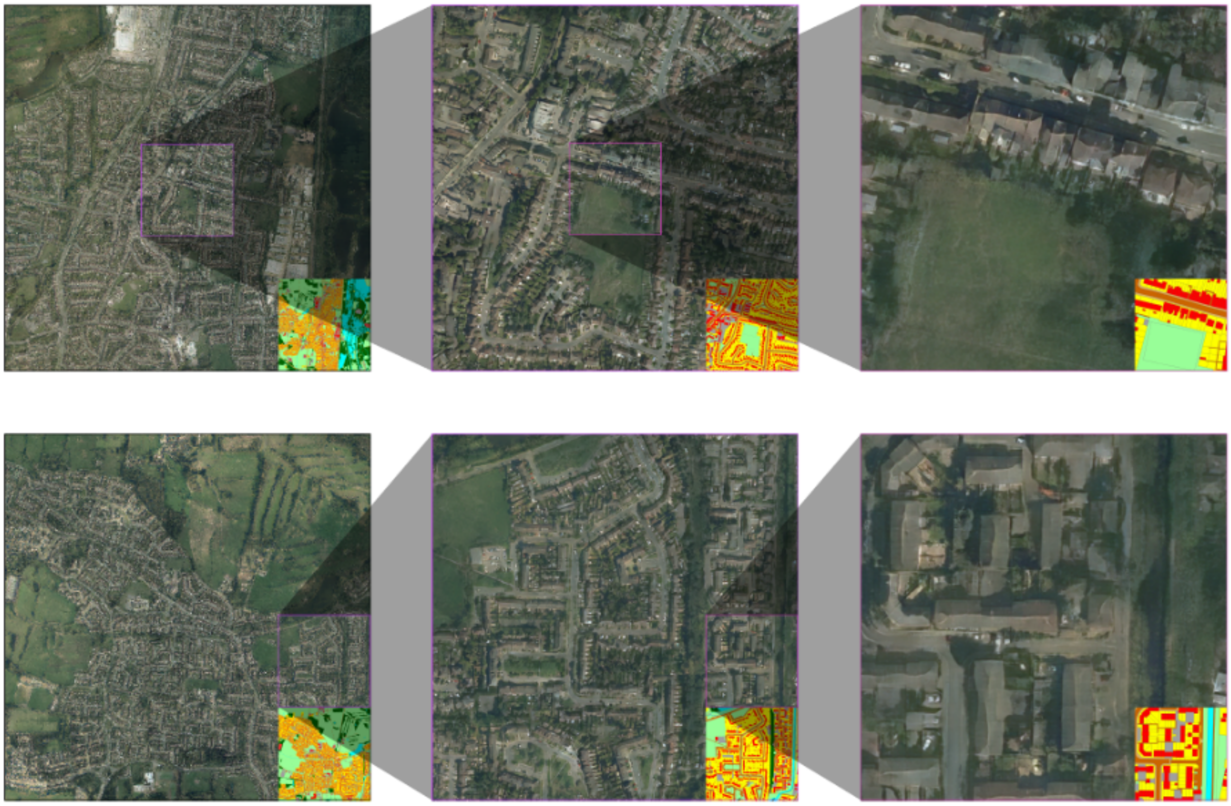

Figure 9: *Results from the SSS pipeline. Rows: examples of generated satellite images from real-world map data. Columns: left to right show scales $z = 1, 2, 3$. Inset are the corresponding map inputs.*

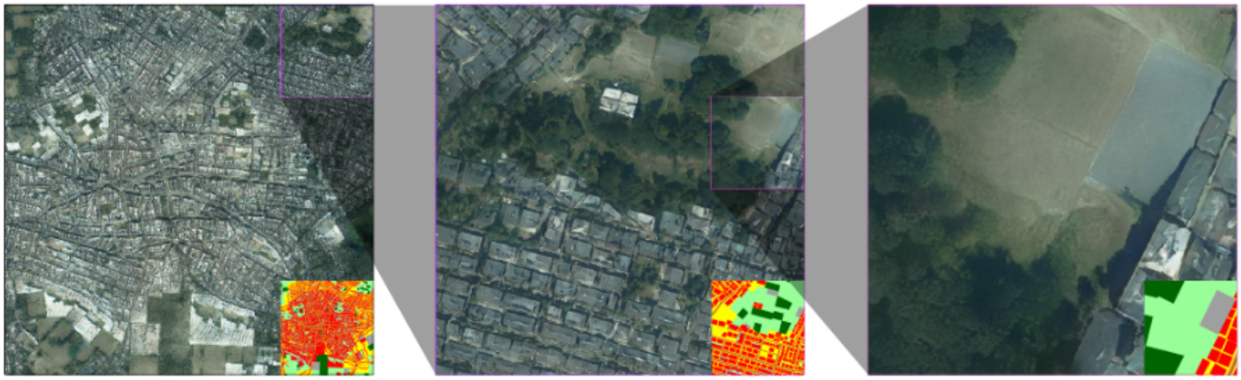

Figure 10: *Result of our system on synthetic procedurally generated map data.*

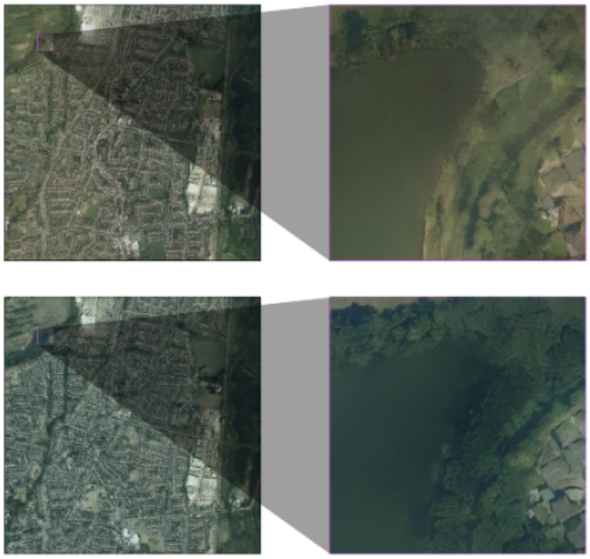

Figure 11: *Style generation of z^1 (left) and z^3 (right) images in a 3 scale level SSS system. Each row has a different latent encoding, e, input to* map2sat^1, *giving different styles which remain consistent throughout the scale levels.*

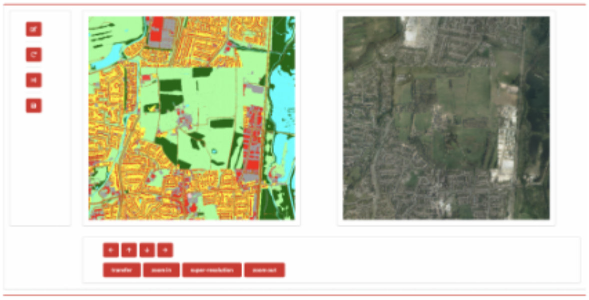

Figure 12: *User interface for our interactive system. In this figure, we show a result of generated satellite image (right) by corresponding map (left). Here the user has edited the map by rotating the central tile. Our system is able to generate realistic results even with the discontinuities in the map data.*

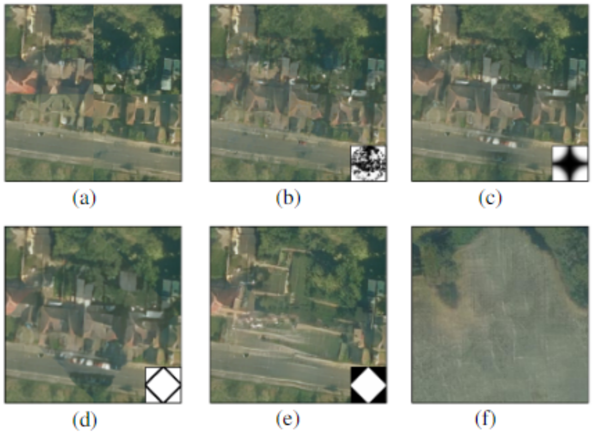

Figure 13: *This figure illustrates results from selected baseline techniques. (a) t from our* map2sat *network; (b) the output of our* seam2cont *network; (b, inset) the calculated mask, m; (c) the result of fixed-mask (c, inset) image blending. (d) the Graphcut algorithm calculates four minimum energy seams within margins (d, inset); (e) Generative Image Inpainting is not conditioned on map data, but is used to inpaint a diamond area (e, inset); (f) TileGAN conditioned on a lower-resolution satellite image.*

To evaluate our seam2cont network results, we use two methods. First, we design a qualitative user study survey. We ask users to identify the best system for removal of seams between tiles to examine the effectiveness of our seam2cont network against other algorithms. Second, we introduce a novel algorithm, MoT (Mean
over Tiles), to measure the ability to remove seams in images. We evaluate against the following baselines: traditional image blending with a soft mask, Graphcut [KSE*03], Generative Image Inpainting [YLY*19], and TileGAN [FAW19].

For traditional image blending, we use sas input and blends with tby a soft mask shown in the right bottom corner of Figure 13 (c) to obtain an image without seams. Graphcut removes seams by splicing the center part of tinto selecting a cut through a buffer area.

We use a small area, 10 ×10 pixels, shown in the right bottom corner of Figure 13 (d) as our buffer area. We train the Generative
Image Inpainting model, which uses gated convolutions [YLY*19] for the image inpainting task; it is trained with our generated seams dataset sand the diamond shape mask shown in the right bottom corner of Figure 13 (e). This is to let the model learn to generate the center diamond area which covers the seams. Generative Image Inpainting does synthesize reasonable results, but it often changes the original content structure in sas it is not conditioned on map
data. Last, we train PGGAN [KALL17] with ground-truth satellite images of 256×256 pixels resolution for TileGAN. TileGAN can generate large seamless images by searching the latent space of the pre-trained PGGAN model to create tiles that are stitched into a large image. It is not strictly an image-to-image translation system, but can take a small, low-resolution, image as guidance to generate large, high-resolution, images.
Figure 13 shows selected representative output tiles for each of the baselines. Traditional image blending in Figure 13 (c) does a good job generally, but since it uses a fixed mask, the rough shape of the mask can sometimes be observed. This is more obvious when we apply image blending on green fields. Figure 13 (d) is created by the Graphcut algorithm, which clearly shows visible seams. Figure 13 (e) is generated by Generative Image Inpainting; it does synthesize good results, however, it changes the original structure of Figure 13 (a); traces of the diamond mask can also be observed.
Figure 13 (f) is a tile from a large image synthesized by TileGAN; an 8,320 ×8,320 pixel image is guided by a low-resolution image which is cropped into the corresponding tile. TileGAN creates an overly “blobby” result with poor large scale features. While TileGAN is true to the guidance image, the guidance is too low resolution to effectively guide the generation of crisp man-made features such as streets. When TileGAN is trained on satellite images of urban areas, the results can be poor because of the difficulty in searching for a latent encoding which matches large, sharp features over tile boundaries. This is illustrated in the additional materials file
baselines_8k_images/tilegan.jpg. We provide large area results (8k images) of all baselines in baselines_8k_images/. We also evaluate against the following variants and later use these to form an ablation study: photographic ground-truth, groundtruth with seams (created by flipping each tile along the diagonal), original SPADE [IZZE17] with no seam2cont networks, our
map2sat results without seam2cont network, and our results which are generated by procedural map data. The User study survey evaluates user’s perceived quality of seam removal. We designed a user questionnaire which shows an
image with seams, and ask which of two techniques remove the seams in the most natural way. We compare against 4 baselines: Graphcut, Generative Image Inpainting, image blending with a soft mask, and the dataset without seams that we use for training. Comparison trials for one baseline are performed over 10 pairs of
images. We do not include TileGAN in the survey because it does not support generating satellite imagery of a specific area. We asked a total of 20 participants “Which of these two images below removes the seams in the most natural way?” (see misc/user_study.png

in additional materials). We also collected information on participant's digital image editing experience. Among our participants, 55% process digital images every month, while the remaining 45% have no relevant experience.

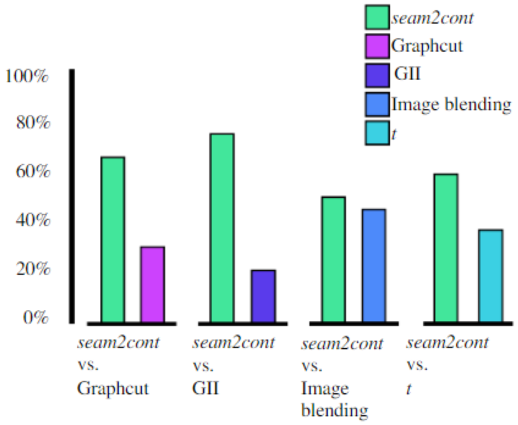

Figure 14: *User study results. We show results of SSS (our approach), Graphcut, Generative Image Inpainting (GII) and t* (map2sat *output, without* seam2cont*). In all scenarios, users rate our seam-removal as more natural than alternatives.*

Figure 14 shows the results of our user study. SSS removes seams in a more natural way than all the baselines in the majority of trials. Our advantage is particularly strong against Graphcut and Generative Image Inpainting. We believe this is because Graphcut cannot complete the stitching task well over a limited buffer area, and Generative Image Inpainting removes seams while changing the original content of the image. We found it surprising that when our results are compared with t, we still have an advantage. This may be because the tile generation near the border (t) may be less accurate, so showing more images from the center of adjacent tiles (y0) may create more realistic results.

The MoT algorithm compares differences between horizontally and vertically adjacent pixels over generated images to highlight any sudden color differences at known tile boundaries. To compute the MoT response we firstly crop the input image by removing (tileWidth/2) pixels from each edge. Let Ca be all pixels in columns {0+a,tileWidth +a,tileWidth *2+ a,...,imageWidth−tileWidth+a}from a cropped image. Similarly Rafor all rows. The MoT response at pixel position, x,y, is the absolute difference of the averages of differences of adjacent pixels:

$$D(x,y) = |Cx-Cx+1|+|Ry-Ry+1| \quad (15)$$

We use tileWidth =256 to match our generated tile size, x,y∈ 0...til Width −1, and a 8,192 ×8,192 pixel image to calculate the MoT values.Figure 15 shows the results of the MoT algorithm. The horizontal axis shows discontinuities at each pixel. The map2sat results show an obvious response peak at tile boundaries, while after the seam2cont network our results do not have this peak. This illustrates that our network has learned to remove the seams over the image. This algorithm also works for other baselines and variants; table 1shows the absolute difference between max(D(x,y)) and mean(D(x,y)) over x,y for the baselines and variants. When the difference value is greater than 0.002 a sharp vertical or horizontal seam is present.

Expected Outcome:

In the following results we use SSS with zmax =3 and a scale factor of f=4 to create a 512×512 pixels image at z1, a 2,048×2,048 pixels image at z2, and a 8,192×8,192 pixels image at z3. Figure 9shows the output of our technique for real-world map guidance images; further results are given in our additional materials (see folder sss_8k_images/). Figure 10 shows the output of our technique for synthetic map labels generated using a procedural modeling system. Figure 11 illustrates the style continuity of SSS pipeline. Finally, our interactive system is introduced in figure 12. We allow users to move around in the map, zoom in or out, and generate corresponding satellite images of different styles. We also provide some operations to modify the map, such as randomly replacing a single tile in the map input. All networks (map2sat and seam2cont) for 3 scales can be loaded into the 12GB memory of a Nvidia Titan V graphics card. Generation times for a 512 ×512 pixels window vary between 2.4 and 3.42 seconds depending on the scale. Smaller scales require additional network evaluations.

Limitations:

There are several limitations of our SSS pipeline. First, our map2sat is based on SPADE [PLWZ19] and Pix2Pix [IZZE17] which means that the quality of the generated satellite images is limited by these architectures. We also tried Progressive Growing of GANs when we implementing TileGAN for comparison, but there is still a gap between the generated satellite image and the ground-truth satellite image. Another limitation is that we only remove the linear seams between adjacent tiles. The seam2cont network also leaves theoretical singular point discontinuities at the midpoints of each tile edge after blending between sand t. We found this had minimal impact on the quality of seam removal in practice, but can envisage an additional network which removes such singular discontinuities. Finally, we produce strong results because SSS is conditioned on cartographic data, but compare to some baselines which can not use this data fully (e.g., TileGAN), or were not designed for this task (e.g., Graphcut).

Conclusion:

We have proposed the SSS pipeline to synthesize large satellite images that are continuous both spatially and throughout scale-space at arbitrary size for given cartographic map data while not requiring large memory use. It also supports procedurally generated cartographic maps for entirely novel areas which allow users to create textures on demand. This is achieved by the map2sat network architecture that can keep style continuity in specified scale space
and seam2cont network architecture to splice small images seamlessly into one large image. We also implement a interactive system to let users explore of an boundless satellite images generated conditioned on cartographic data.

J. Zhu & T. Kelly / Seamless Satellite-image Synthesis

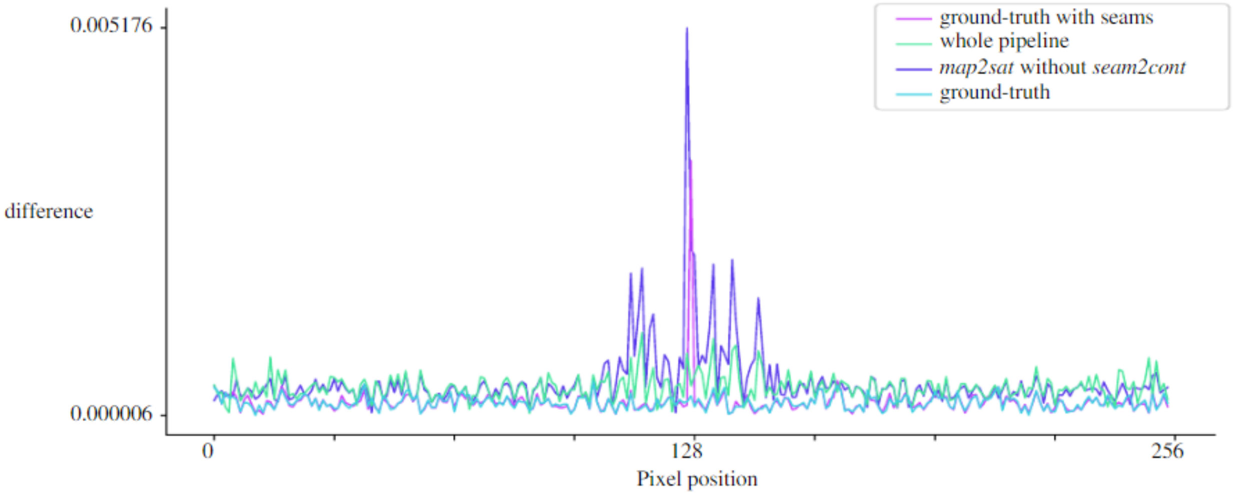

Figure 15: *Selected MoT algorithm results: difference value response for ground-truth, ground-truth with seams, our* map2sat *with no* seam2cont *and our whole pipeline. Images with strong vertical and horizontal seams have a strong response on the tile boundary at 128 pixels.*

In future work, we would like to study the impact of different edge constraint images (c) in our seam2cont network by replacing with lob-shaped masks and using other conditioning for map2sat, such as multi-spectral images, pencil sketches, and low resolution images. Another avenue of exploration is the scale factor of 4 between z; we would like to try different values with a perceptual evaluation. In addition, we suspect that using different offsets at different scales would reduce regularity and improve the quality of results. We also want to try to modify the neural network structure used; for example, adding adaptive discriminator augmentation [KAH*20]. Finally, we want to extend our work to 3D voxel based networks using transformers to synthesize a 3D virtual environment from cartographic map data.

References:

[ADA*04] Agarwala A., Dontcheva M., Agrawala M., Drucker S., Colburn A., Curless B., Salesin D., Cohen M.: Interactive digital photomontage. In ACM SIGGRAPH. 2004, pp. 294–302. 2

[AKM*21] Alsallakh B., Kokhlikyan N., Miglani V., Yuan J., Reblitz-Richardson O.: Mind the pad – {cnn}s can develop blind spots. In ICLR (2021). URL: https://openreview.net/forum?id=m1CD7tPubNy.4

[AS07] Avidan S., Shamir A.: Seam carving for content-aware image resizing. In ACM SIGGRAPH (New York, NY, USA, 2007), Association for Computing Machinery, p. 10–es. URL: https://doi.org/10.1145/1275808.1276390,doi:10.1145/1275808.1276390.2

[BSFG09] Barnes C., Shechtman E. , Finkelstein A., Goldman D. B.: PatchMatch: A randomized correspondence algorithm for structural image editing. ACM SIGGRAPH 28, 3 (Aug. 2009). 2

[BSP*19] Bau D., Strobelt H., Peebles W., Wulff J., Zhou B., Zhu J.-Y., Torralba A.: Semantic photo manipulation with a generative image prior. ACM TOG 38, 4 (July 2019). URL: https://doi.org/10.1145/3306346.3323023,doi:10.1145/3306346.3323023.2

[CEW*08] Chen G., Esch G., Wonka P., Müller P., Zhang E.: Interactive procedural street modeling. In ACM SIGGRAPH. 2008, pp. 1–10. 2

[CPT03] Criminisi A., Perez P., Toyama K.: Object removal by exemplar-based inpainting. In IEEE CVPR (2003), vol. 2, IEEE, pp. II–II. 2

[DCX*18] Deng J., Cheng S., Xue N., Zhou Y., Zafeiriou S.: Uv-gan: Adversarial facial uv map completion for pose-invariant face recognition. In IEEE ICCV (2018), pp. 7093–7102. 2

[DSB*12] Darabi S., Shechtman E., Barnes C., Goldman D. B., Sen P.: Image melding: combining inconsistent images using patch-based synthesis. ACM SIGGRAPH 31, 4 (2012), 1–10. 2

[EF01] Efros A. A ., Freeman W. T.: Image quilting for texture synthesis and transfer. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques (2001), pp. 341–346. 2

[EL99] Efros A. A ., Leung T. K.: Texture synthesis by non-parametric sampling. In IEEE ICCV (1999), vol. 2, IEEE, pp. 1033–1038. 2

[FAW19] Frühstück A., Alhashim I., Wonka P.: Tilegan: synthesis of large-scale non-homogeneous textures. ACM SIGGRAPH 38, 4 (2019), 1–11. 2,4,7

[Fou21] Foundation O.: OpenStreetMap, 2021. URL: https://www.openstreetmap.org.2

[GAD*20] Guehl P., Allegre R., Dischler J.-M., Benes B., Galin E.: Semi-procedural textures using point process texture basis functions. In CGF (2020), vol. 39, Wiley Online Library, pp. 159–171. 2

[GDG*17] Guérin É., Digne J., Galin É., Peytavie A., Wolf C., Benes B., Martinez B.: Interactive example-based terrain authoring with conditional generative adversarial networks. ACM TOG 36, 6 (2017), 1–13. 2

[GEB15] Gatys L., Ecker A. S., Bethge M .: Texture synthesis using convolutional neural networks. In NIPS (2015), pp. 262–270. 2

[GEB16] Gatys L. A., Ecker A. S., Bethge M.: Image style transfer using convolutional neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (2016), pp. 2414–2423. 2

[geo] Geopackage. https://www.geopackage.org/. Accessed: 2010-09-30. 3

[HE07] Hays J., Efros A. A.: Scene completion using millions of photographs. ACM SIGGRAPH 26, 3 (July 2007), 4–es. URL: https://doi.org/10.1145/1276377.1276382,doi:10.1145/1276377.1276382.2

[IZZE17] Isola P., Zhu J.-Y., Zhou T., Efros A. A.: Image-to-image translation with conditional adversarial networks. IEEE CVPR (2017). 1,2,4,9,10

[KAH*20] Karras T., Aittala M., Hellsten J., Laine S., Lehtinen J., Aila T.: Training generative adversarial networks with limited data. arXiv preprint arXiv:2006.06676 (2020). 11

[KALL17] Karras T., Aila T., Laine S., Lehtinen J.: Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196 (2017). 7

[KB15] Kingma D. P., Ba J.: Adam: A method for stochastic optimization. In ICLR (2015), Bengio Y., LeCun Y., (Eds.). URL: http://arxiv.org/abs/1412.6980.4

[KGS*18] Kelly T., Guerrero P., Steed A., Wonka P., Mitra N. J.: Frankengan: guided detail synthesis for building mass models using style-synchonized gans. ACM TOG 37, 6 (2018), 1–14. 2

[KSE*03] Kwatra V., Schödl A., Essa I., Turk G., Bobick A.: Graphcut textures: image and video synthesis using graph cuts. ACM SIGGRAPH 22, 3 (2003), 277–286. 2,4,7

[LCH*15] Legesse F., Chernavskaia O., Heuke S., Bocklitz T., Meyer T., Popp J., Heintzmann R.: Seamless stitching of tile scan microscope images. Journal of microscopy 258, 3 (2015), 223–232. 2

[LLX*01] Liang L., Liu C., Xu Y.-Q., Guo B., Shum H.-Y.: Real-time texture synthesis by patch-based sampling. ACM Trans. Graph. 20, 3 (July 2001), 127–150. URL: https://doi.org/10.1145/501786.501787,doi:10.1145/501786.501787.2

[Mas21] Mastermap O. S.: Ordnance, 2021. URL: https://www.ordnancesurvey.co.uk/business-government/products/mastermap-topography.11

[MZWVG07] Müller P., Zeng G., Wonka P., Van Gool L.: Image-based procedural modeling of facades. ACM SIGGRAPH 26, 3

(2007), 85. 2

[PAAEZ20] Park T., Alexei A. Efros R. Z., Zhu J.-Y.: Contrastive learning for unpaired image-to-image translation. ECCV (2020). 4

[PGB03] Pérez P., Gangnet M., Blake A.: Poisson image editing. In ACM SIGGRAPH 2003 Papers. 2003, pp. 313–318. 2

[PLWZ19] Park T., Liu M.-Y., Wang T.-C., Zhu J.-Y.: Semantic image synthesis with spatially-adaptive normalization. In IEEE CVPR (2019), pp. 2337–2346. 1,2,3,5,6,10

[PM01] Parish Y. I. H., Müller P.: Procedural modeling of cities. In Proceedings of SIGGRAPH 2001 (2001), pp. 301–308. doi:10.1145/383259.383292.2

[RFB15] Ronneberger O., Fischer P., Brox T.: U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (2015), Springer, pp. 234–241. 2,4

[RKB04] Rother C., Kolmogorov V., Blake A.: " grabcut" interactive foreground extraction using iterated graph cuts. ACM SIGGRAPH 23, 3 (2004), 309–314. 2

[Ser21] Service D. O. S. W. M.: Digimap, 2021. URL: https://digimap.edina.ac.uk/.2,11

[TSK*19] Teterwak P., Sarna A., Krishnan D., Maschinot A., Belanger D., Liu C., Freeman W. T.: Boundless: Generative adversarial networks for image extension. In IEEE ICCV (2019), pp. 10521–10530. 4

[VAW*10] Vanegas C. A., Aliaga D. G., Wonka P., Müller P., Waddell P., Watson B.: Modelling the appearance and behaviour of urban spaces. In CGF (2010), vol. 29, Wiley Online Library, pp. 25–42. 2

[VKW*12] Vanegas C. A., Kelly T., Weber B. , Halatsch J., Aliaga D. G ., Müller P.: Procedural generation of parcels in urban modeling. In CGF Eurographics (2012), vol. 31, Wiley Online Library, pp. 681–690. 2

[WL00] Wei L.-Y., Levoy M.: Fast texture synthesis using tree-structured vector quantization. In Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (USA, 2000), SIGGRAPH '00, ACM Press/Addison-Wesley Publishing Co., p. 479–488. URL: https://doi.org/10.1145/344779.345009,doi:10.1145/344779.345009.2

[WLKT09] Wie L.-Y., Lefebvre S., Kwatra V., Turk G.: State of the Art in Example-based Texture Synthesis. In CGF Eurographics (2009), Pauly M., Greiner G., (Eds.), The Eurographics Association. doi:10.2312/egst.20091063.2

[WLZ*18] Wang T.-C., Liu M.-Y., Zhu J.-Y., Tao A., Kautz J., Catanzaro B.: High-resolution image synthesis and semantic manipulation with conditional gans. In IEEE CVPR (2018), pp. 8798–8807. 2

[WZZH19] Wu H., Zheng S., Zhang J., Huang K.: Gp-gan: Towards realistic high-resolution image blending. In Proceedings of the 27th ACM International Conference on Multimedia (2019), pp. 2487–2495. 2

[YLY*19] Yu J., Lin Z., Yang J., Shen X., Lu X., Huang T. S.: Free-form image inpainting with gated convolution. In IEEE ICCV (2019), pp. 4471–4480. 7

[ZWS20] Zhang L., Wen T., Shi J.: Deep image blending. In The IEEE Winter Conference on Applications of Computer Vision (2020), pp. 231–240. 2

[ZZB*18] Zhou Y., Zhu Z., Bai X., Lischinski D., Cohen-Or D., Huang H.: Non-stationary texture synthesis by adversarial expansion. ACM SIGGRAPH 37, 4 (July 2018). URL: https://doi.org/10.1145/3197517.3201285,doi:10.1145/3197517.3201285.2

[ZZL19] Zhan F., Zhu H., Lu S.: Spatial fusion gan for image synthesis. In IEEE CVPR (2019), pp. 3653–3662. 2

[ZZP*17] Zhu J.-Y., Zhang R., Pathak D. , Darrell T., Efros A. A., Wang O., Shechtman E .: Toward multimodal image-to-image translation. In NIPS (2017). 4