



The latest news from Google AI

Seamless Google Street View Panoramas

Thursday, November 9, 2017

Posted by Mike Krainin, Software Engineer and Ce Liu, Research Scientist, Machine Perception

In 2007, we introduced [Google Street View](#), enabling you to explore the world through panoramas of neighborhoods, landmarks, museums and more, right from your browser or mobile device. The creation of these panoramas is a complicated process, involving capturing images from a multi-camera rig called a rosette, and then using image blending techniques to carefully stitch them all together. However, many things can thwart the creation of a "successful" panorama, such as mis-calibration of the rosette camera geometry, timing differences between adjacent cameras, and [parallax](#). And while we attempt to address these

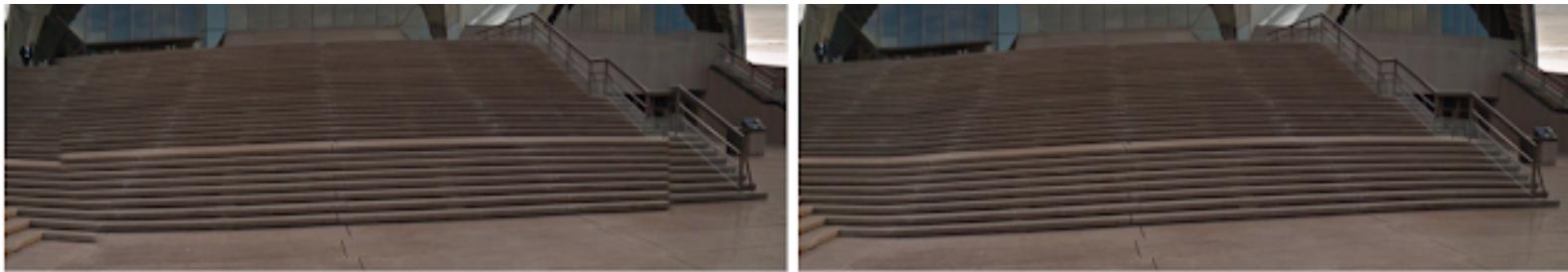
issues by using approximate scene geometry to account for parallax and frequent camera re-calibration, visible seams in image overlap regions can still occur.



Left: A Street View car carrying a multi-camera rosette. Center: A close-up of the rosette, which is made up of 15 cameras.

Right: A visualization of the spatial coverage of each camera. Overlap between adjacent cameras is shown in darker gray.





Left: The Sydney Opera House with stitching seams along its iconic shells. Right: The same Street View panorama after optical flow seam repair.

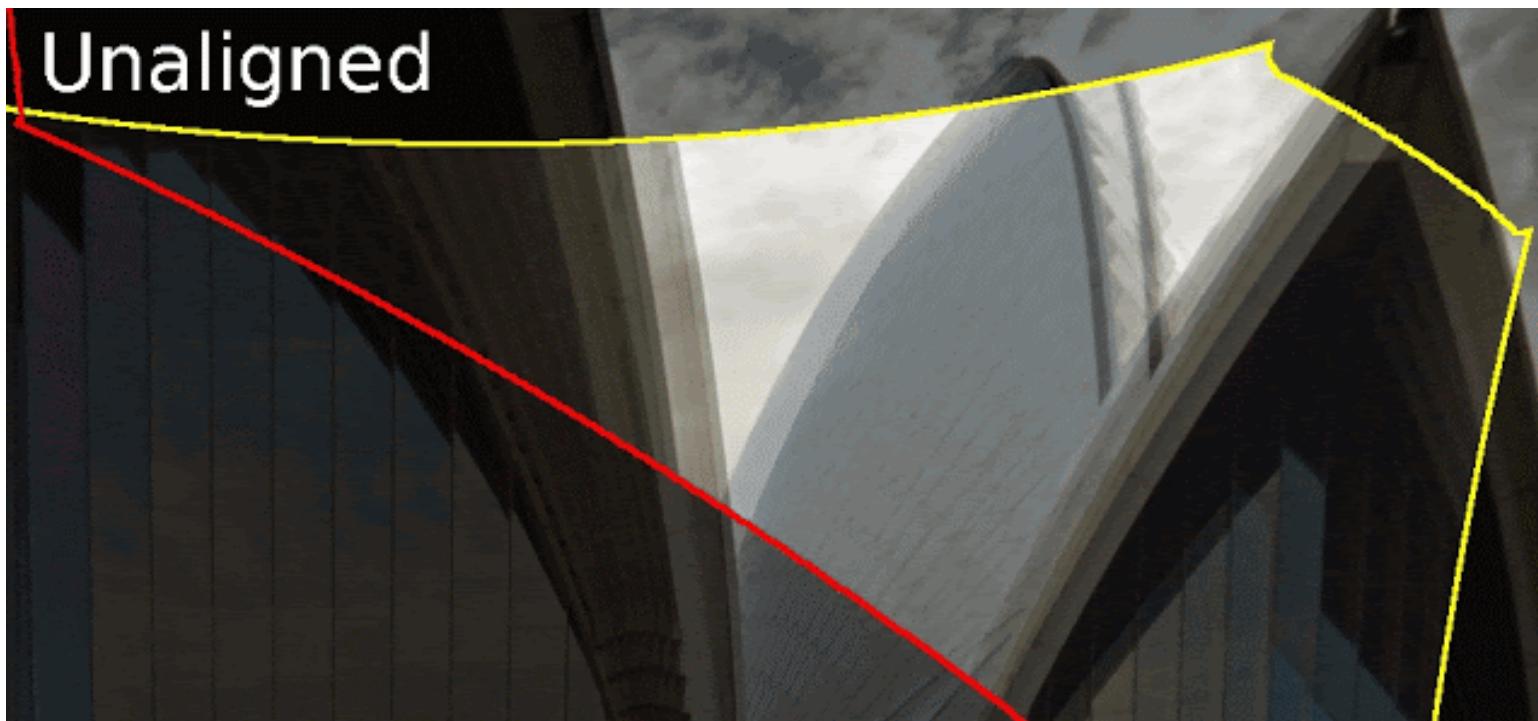
In order to provide more seamless Street View images, we've developed a new algorithm based on [optical flow](#) to help solve these challenges. The idea is to subtly warp each input image such that the image content lines up within regions of overlap. This needs to be done carefully to avoid introducing new types of visual artifacts. The approach must also be robust to varying scene geometry, lighting conditions, calibration quality, and many other conditions. To simplify the task of aligning the images and to satisfy computational requirements, we've broken it into two steps.

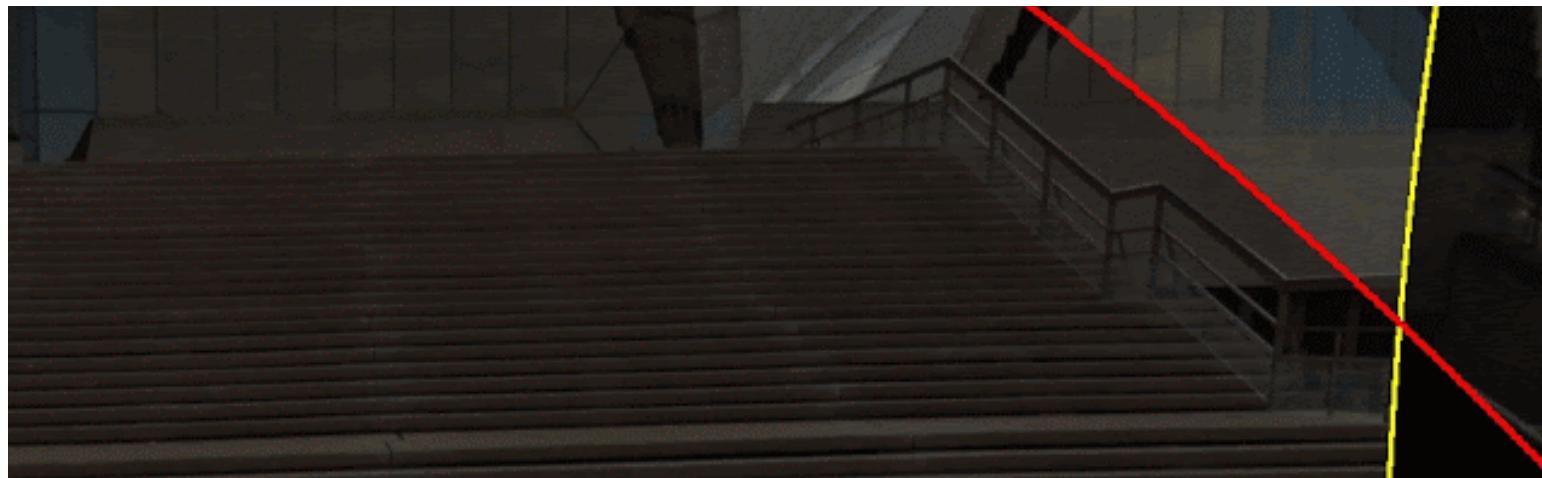
Optical Flow

The first step is to find corresponding pixel locations for each pair of images that overlap. Using techniques described in our [PhotoScan blog post](#), we compute optical flow from one image to the other. This provides a smooth and dense correspondence field. We then downsample the correspondences for computational efficiency. We also discard correspondences where there isn't enough visual structure to be confident in the results of optical flow.

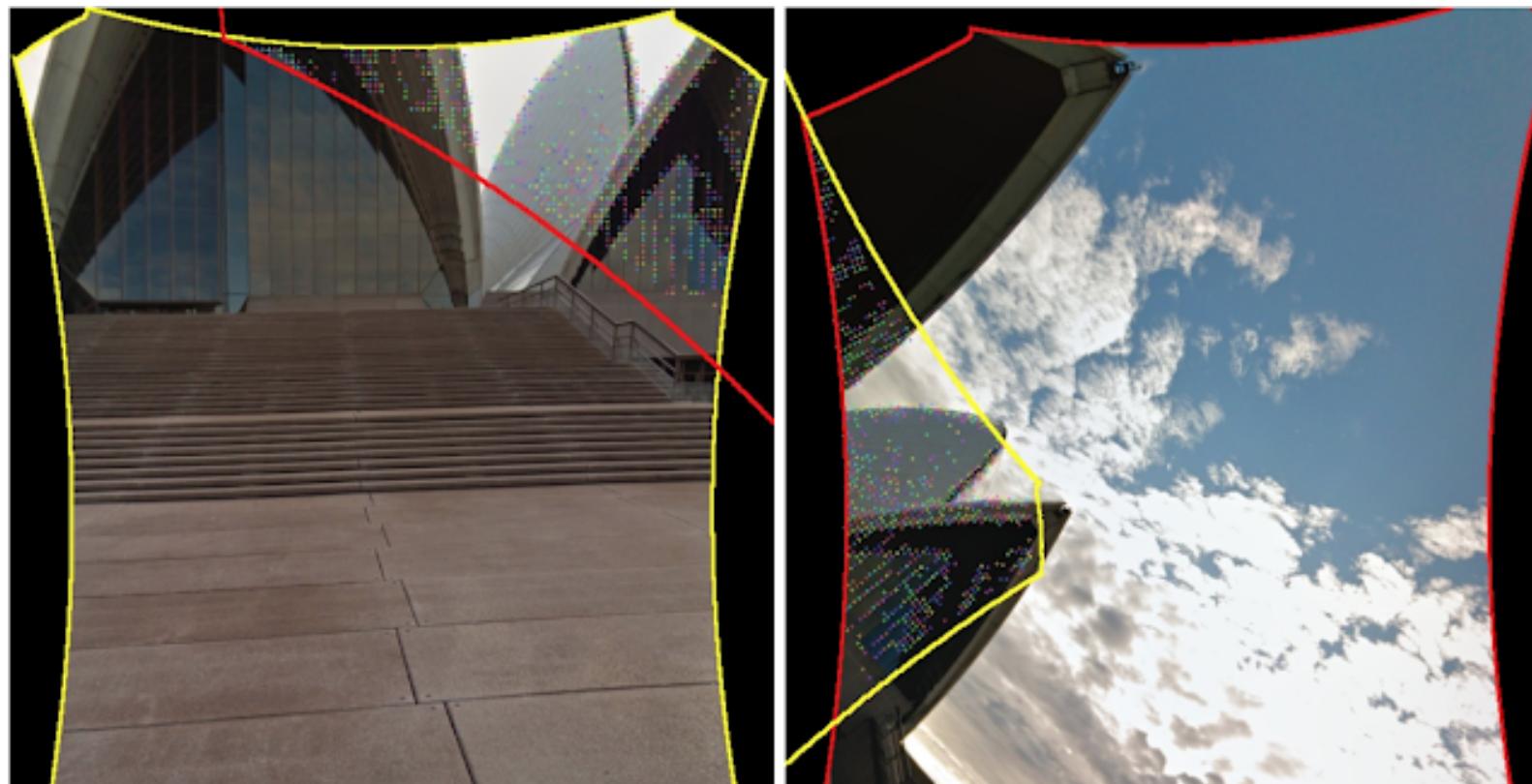


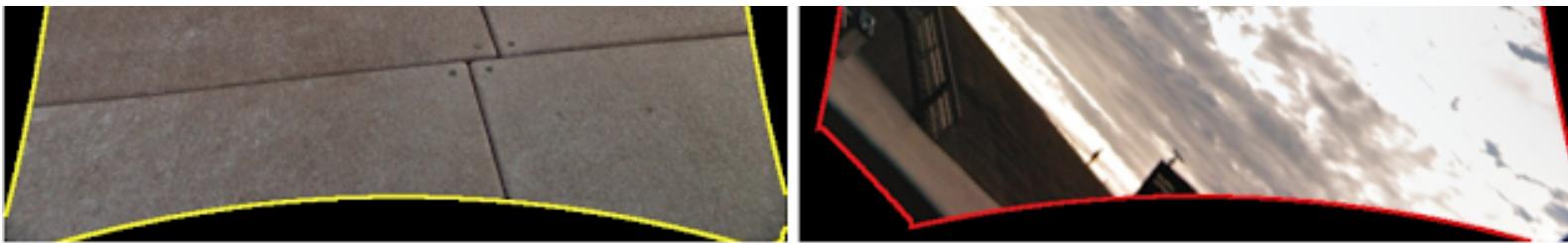
The boundaries of a pair of constituent images from the rosette camera rig that need to be stitched together.





An illustration of optical flow within the pair's overlap region.

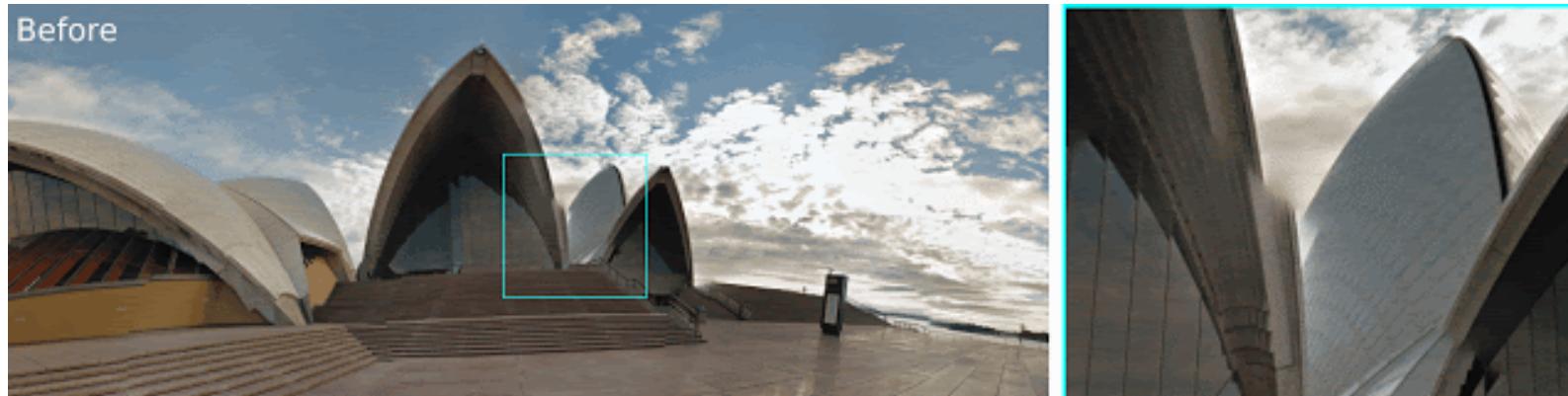


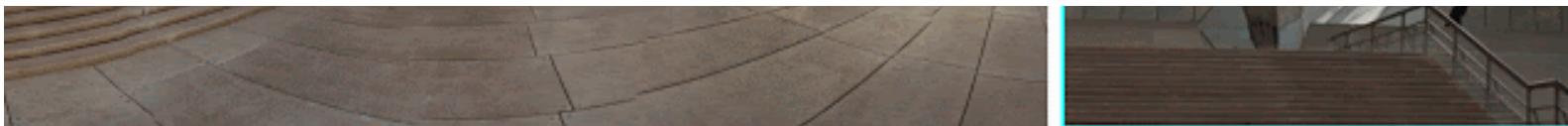


Extracted correspondences in the pair of images. For each colored dot in the overlap region of the left image, there is an equivalently-colored dot in the overlap region of the right image, indicating how the optical flow algorithm has aligned the point. These pairs of corresponding points are used as input to the global optimization stage. Notice that the overlap covers only a small portion of each image.

Global Optimization

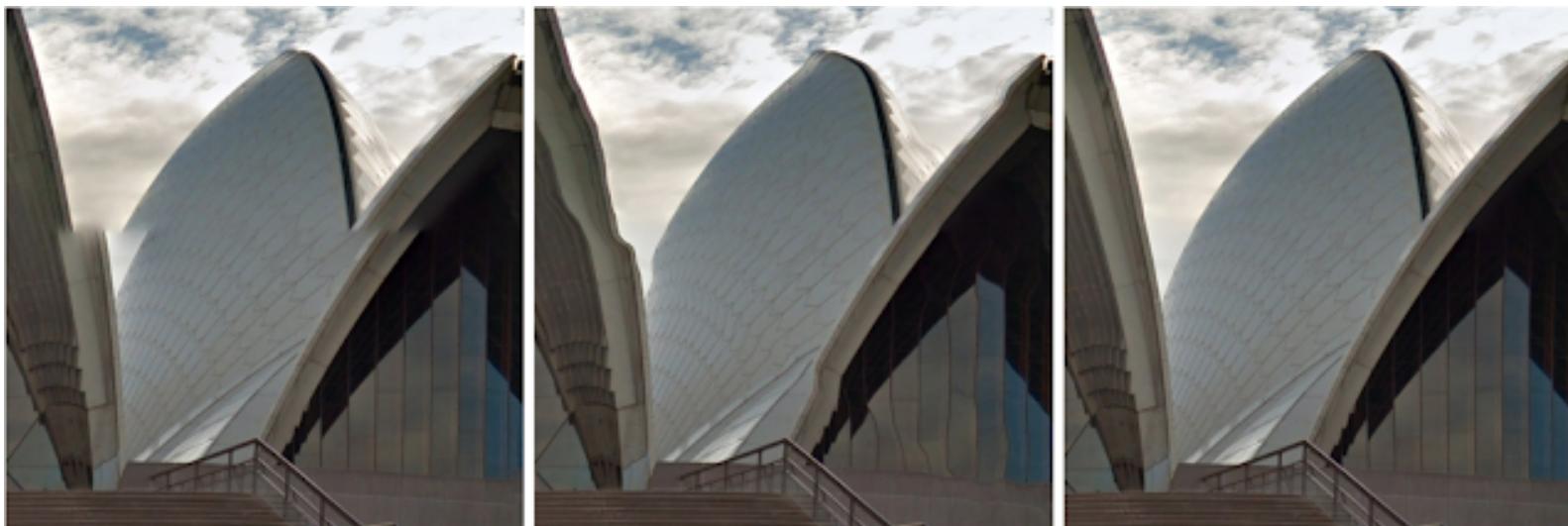
The second step is to warp the rosette's images to simultaneously align all of the corresponding points from overlap regions (as seen in the figure above). When stitched into a panorama, the set of warped images will then properly align. This is challenging because the overlap regions cover only a small fraction of each image, resulting in an under-constrained problem. To generate visually pleasing results across the whole image, we formulate the warping as a [spline](#)-based flow field with spatial regularization. The spline parameters are solved for in a non-linear optimization using Google's open source [Ceres Solver](#).





A visualization of the final warping process. Left: A section of the panorama covering 180 degrees horizontally. Notice that the overall effect of warping is intentionally quite subtle. Right: A close-up, highlighting how warping repairs the seams.

Our approach has many similarities to [previously published work](#) by Shum & Szeliski on “deghosting” panoramas. Key differences include that our approach estimates dense, smooth correspondences (rather than patch-wise, independent correspondences), and we solve a nonlinear optimization for the final warping. The result is a more well-behaved warping that is less likely to introduce new visual artifacts than the kernel-based approach.



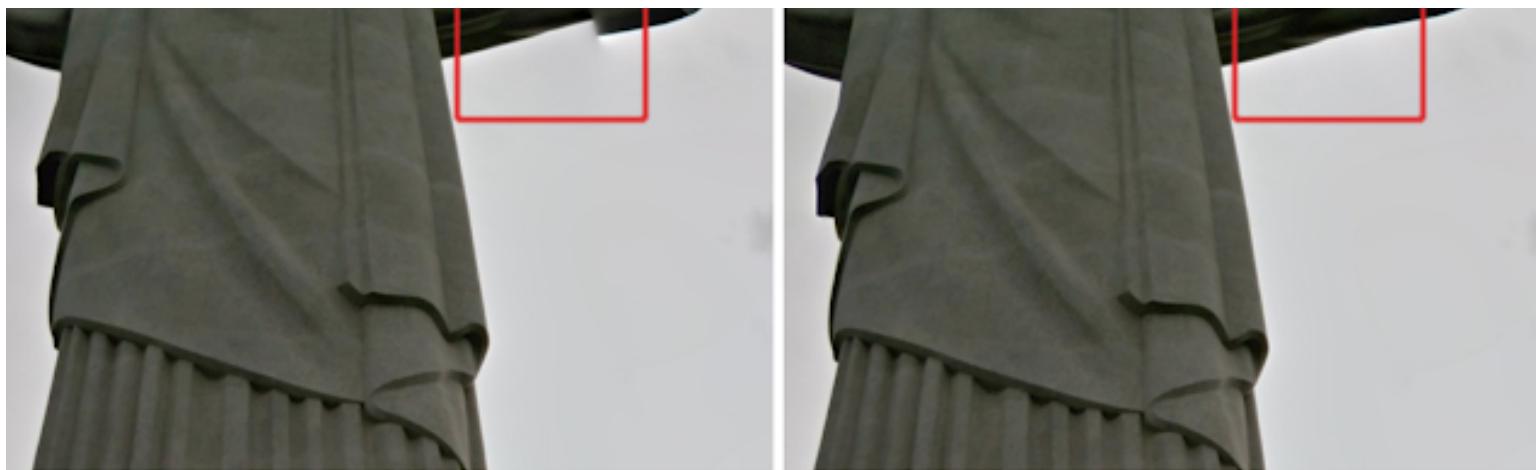
Left: A close-up of the un-repaired panorama. Middle: Result of kernel-based interpolation. This fixes discontinuities but at the expense of strong wobbling artifacts due to the small image overlap and limited footprint of kernels. Right: Result of our global optimization.

THIS IS IMPORTANT BECAUSE OUR ALGORITHM NEEDS TO BE ROBUST TO THE ENORMOUS DIVERSITY IN content in Street View's billions of panoramas. You can see how effective the algorithm is in the following examples:

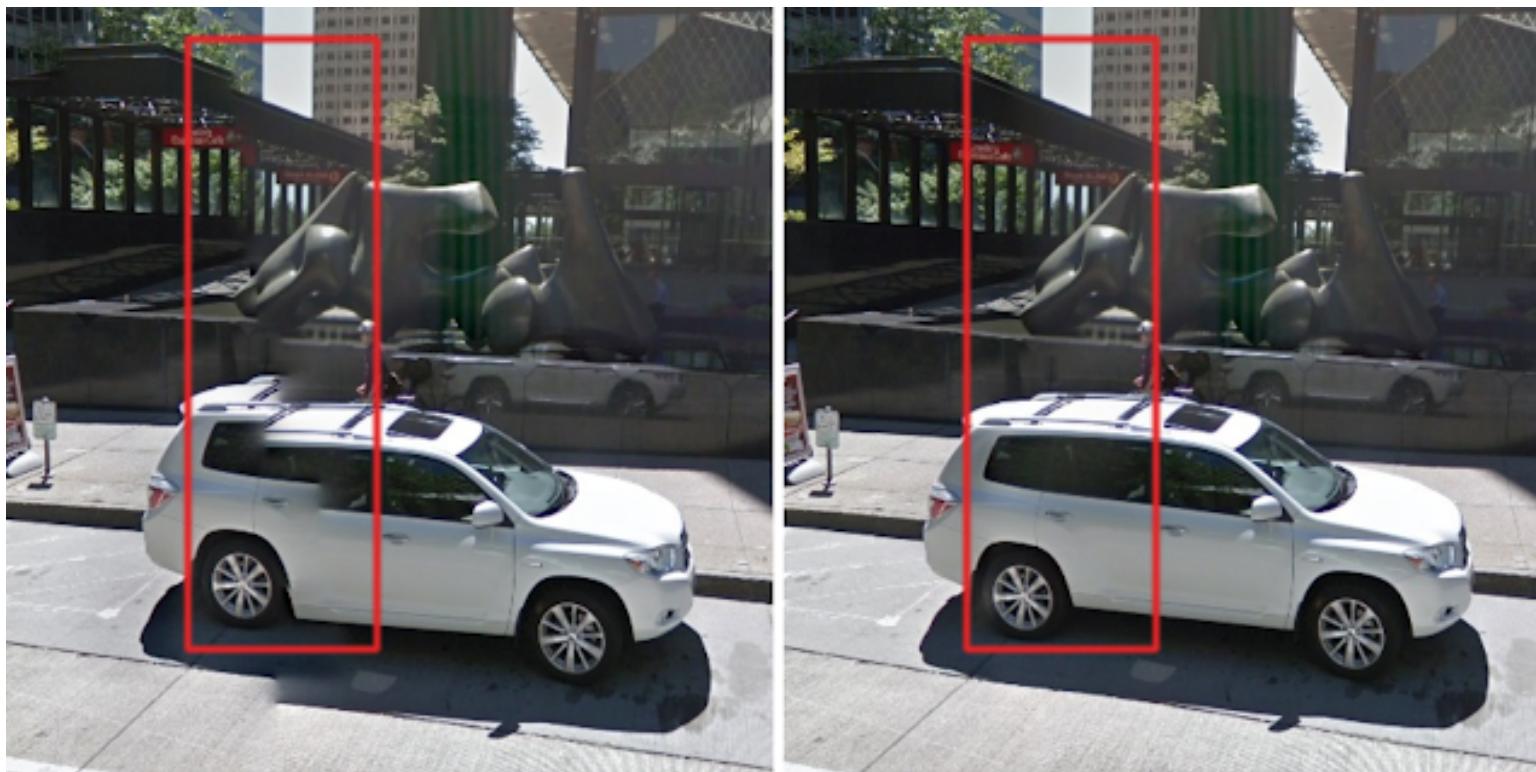


Tower Bridge, London





Christ the Redeemer, Rio de Janeiro



An SUV on the streets of Seattle

This new algorithm was recently added to the Street View stitching pipeline. It is now being used to restitch existing panoramas on an ongoing basis. Keep an eye out for improved Street View near you!

Seamless Google Street View Panoramas



Acknowledgements

Special thanks to Bryan Klingner for helping to integrate this feature with the Street View infrastructure.





The classic Google logo, consisting of the word "Google" in its signature blue, red, yellow, green, and red font.

[Google](#) · [Privacy](#) · [Terms](#)