Time-lapse Mining from Internet Photos

Ricardo Martin-Brualla[1*] David Gallup[2] Steven M. Seitz[1,2]
[1]University of Washington [2]Google Inc.

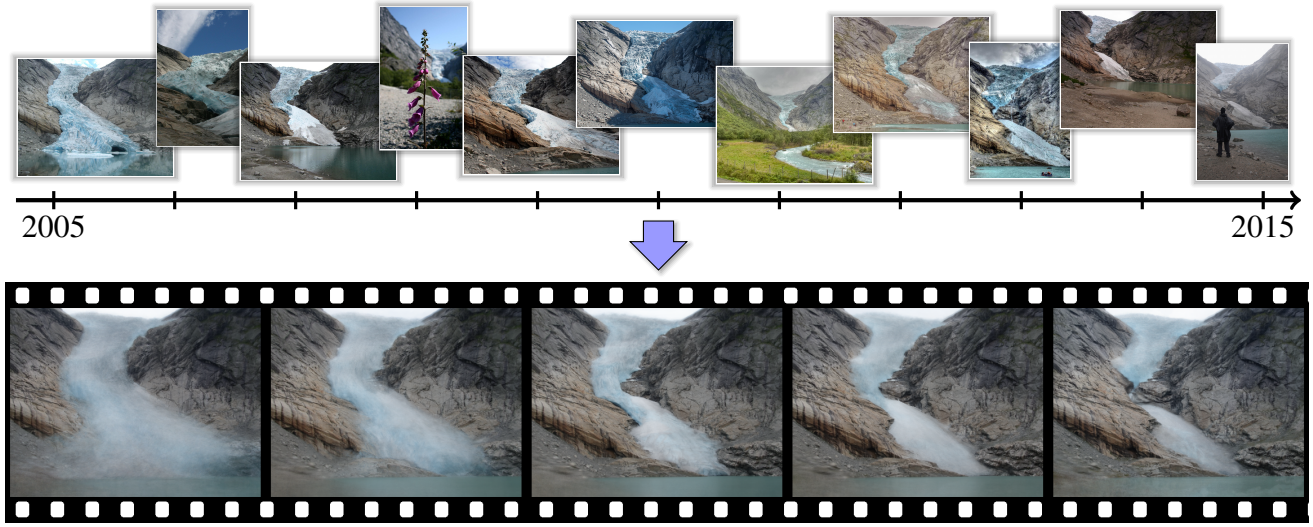


Figure 1: *We mine Internet photo collections to generate time-lapse videos of locations all over the world. Our time-lapses visualize a multitude of changes, like the retreat of the Briksdalsbreen Glacier in Norway shown above. The continuous time-lapse (bottom) is computed from hundreds of Internet photos (samples on top).* *Photo credits:* `Aliento Más Allá`, `jirihnidek`, `mcxurxo`, `elka_cz`, *Juan Jesús Orío, Klaus Wißkirchen,* `Daikrieg`, `Free the image`, `dration` *and Nadav Tobias.*

Abstract

We introduce an approach for synthesizing time-lapse videos of popular landmarks from large community photo collections. The approach is completely automated and leverages the vast quantity of photos available online. First, we cluster 86 million photos into landmarks and popular viewpoints. Then, we sort the photos by date and warp each photo onto a common viewpoint. Finally, we stabilize the appearance of the sequence to compensate for lighting effects and minimize flicker. Our resulting time-lapses show diverse changes in the world's most popular sites, like glaciers shrinking, skyscrapers being constructed, and waterfalls changing course.

CR Categories: I.2.10 [Artificial Intelligence]: Vision and Scene Understanding—Modeling and recovery of physical attributes; I.4.3 [Image Processing and Computer Vision]: Enhancement—Filtering and Geometric Correction; I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Stereo and Time-varying imagery

Keywords: time-lapse, computational photography, image-based rendering

*This work was partially done while the first author was an intern at Google.

1 Introduction

We see the world at a fixed temporal scale, in which life advances one second at a time. Instead, suppose that you could observe an entire year in a few seconds—a 10 million times speed-up. At this scale, you could see cities expand, glaciers shrink, seasons change, and children grow continuously. Time-lapse photography provides a fascinating glimpse into these timescales. And while limited time-lapse capabilities are available on consumer cameras [Apple ; Instagram], observing these ultra-slow effects requires a camera that is locked down and focused on a single target over a period of months or years [Extreme Ice Survey].

Yet, these ultra-slow changes are documented by the billions of photos that people take over time. Indeed, an Internet image search for any popular site yields several years worth of photos. In this paper, we describe how to transform these photo collections into high quality, stabilized time-lapse videos. Figure 1 shows a few frames from one result video of a glacier receding over a decade. This capability is transformative; whereas before it took months or years to create one such time-lapse, we can now almost instantly create thousands of time-lapses covering the most popular places on earth. The challenge now is to find the interesting ones, from all of the public photos in the world. We call this problem *time-lapse mining*.

Creating high quality time-lapses from Internet photo sharing sites is challenging, due to the vast viewpoint and appearance variation in such collections. The main technical contribution of this paper is an approach for producing extremely stable videos, in which viewpoint and transient appearance changes are almost imperceptible, allowing the viewer to focus on the more salient, longer time scale scene changes. We employ structure-from-motion and stereo algorithms to compensate for viewpoint variations, and a simple but effective new temporal filtering approach to stabilize appearance. Our second significant contribution is a world-scale deployment,

where we process over 80 million public Internet photos, yielding several thousand mined time-lapses spanning the worlds most photographed sites.

2 Related work

Unstructured time-lapses Very related to our work, [Matzen and Snavely 2014] discover changing elements in 3D scenes by clustering reconstructed 3D patches into space-time cuboids. This approach is limited to reconstructing planar structures like billboards or graffiti in urban scenes. The authors estimate the period of time an element was visible in the scene and propose a 3D visualization where the user can move through time and space, seeing only the discovered elements that existed at the given time.

The 4D Cities project [Schindler et al. 2007; Schindler and Dellaert 2010] models changes in a city using historical imagery over several decades. By reasoning about the visibility of features points, their system infers missing or inaccurate timestamps and builds a sparse 4D reconstruction of the buildings in a city.

Our approach substantially differs from both projects in key aspects. Our focus is to generate complete time-lapse videos (with no holes), instead of building sparse 4D representations of the world. Our approach also works on a global scale, discovering thousands of time-lapses all over the world, in contrast to the handful of sites analyzed by previous work. Finally, our system is not limited in scope to urban scenes and we generate time-lapse videos for diverse natural phenomena.

Lastly, Picasa FaceMovies [Kemelmacher-Shlizerman et al. 2011] use a personal photo collection to generate a movie of how a person ages through time. Besides the obvious difference of exclusively targeting faces, FaceMovies stopped short of creating a continuous time-lapse video which is a key focus of our work.

Time-lapse with human intervention: The *ConstructAide* system [Karsch et al. 2014] aids in analyzing and visualizing construction progress by having the user guide the registration of unstructured photos to a 3D model of a building.

Another common approach to visualizing how a scene changes over time is rephotography, where one compares two photos taken from the same viewpoint. [Bae et al. 2010] presents a user interface to guide a photographer to lock in the exact same viewpoint of a previous photograph of the scene.

Static time-lapses: The synthesis of time-lapse videos from static video cameras, like webcams, has been explored in the literature. In [Bennett and McMillan 2007], ordinary videos are condensed into time-lapses, using sampling and filtering strategies to convey different visual objectives. [Rubinstein et al. 2011] propose a method to denoise small motions in a time-lapse, optimizing the resulting video by borrowing pixel values in a spatio-temporal neighborhood.

Static time-lapse videos also provide extensive information about how the scene interacts with different lighting conditions. This has been exploited to compute factored lighting models [Sunkavalli et al. 2007] and perform photometric stereo to obtain scene BRDFs [Ackermann et al. 2012]. Scene geometry can also be inferred from the shadows cast by clouds [Jacobs et al. 2010] or by finding correspondences along the shadow edges [Abrams et al. 2013]. By using a database of time-lapse videos, [Shih et al. 2013; Laffont et al. 2014] learn appearance transfer models that can change the time of day or time of year of a photograph.

Appearance modeling: The works of [Amirshahi et al. 2008; Whyte et al. 2009] perform image inpainting of occluders in an input photo using Internet photos of the same scene. In contrast, [Hays and Efros 2007] find similar scenes in a large Internet photo collection to inpaint regions of an input image. [Laffont et al. 2012] compute coherent intrinsic images for a photo collection by reasoning about reflectance of pairs of 3D points across the photo collection.

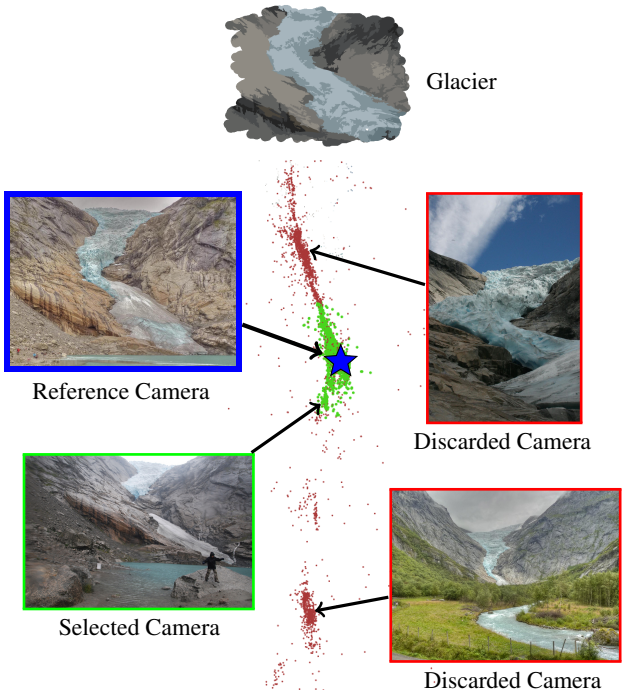


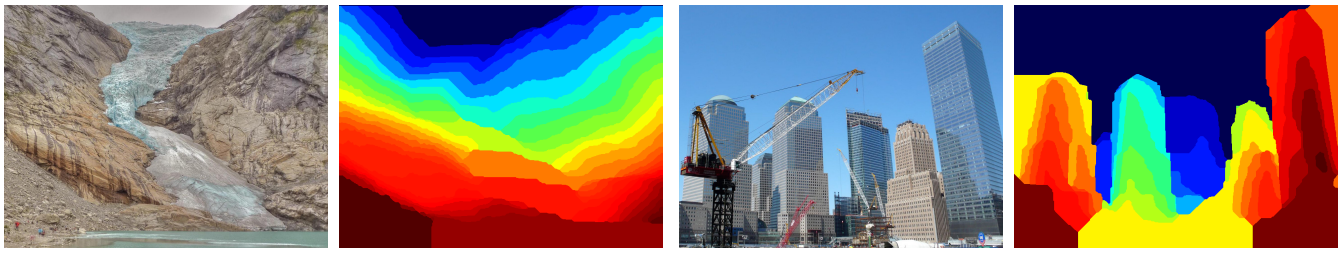
Figure 2: *Top-down view of the Briksdalsbreen Glacier reconstruction. Red and green points correspond to the 9411 camera centers in the SfM reconstruction. The reference image for the time-lapse in Figure 1 is shown in top left and the blue star represents its camera center. Selected cameras for the time-lapse are shown in green and discarded cameras in red. The two images on the right correspond to other clusters in the distribution of photos of the scene. Photo credits:* `Daikrieg`, `jirihnidek` *and Nadav Tobias.*

3D change detection: To detect geometric changes in a city, [Taneja et al. 2011; Taneja et al. 2013] warp different images onto each other by using a previously captured 3D model. In [Ulusoy and Mundy 2014] probabilistic 3D change detection guides the 4D spatio-temporal reconstruction of laboratory scenes.

3 System overview

The input to our system is a collection of 86 million timestamped and geotagged photos around the world. The system automatically discovers all locations in the world with enough imagery and generates a time-lapse video for each.

Section 4 describes how candidate time-lapse video locations are mined from unstructured photo collections. Each candidate time-lapse video consists of a reference camera viewpoint and a set of nearby images. Next, the images of each candidate time-lapse are ordered chronologically and warped into the reference camera to compensate for viewpoint differences, as explained in Section 5. Section 6 describes our approach to stabilize the appearance of the video to compensate for varying lighting conditions and occlusions from transient objects like people.



(a) Briksdalsbreen Glacier, Norway

(b) Goldman Sachs Tower, New York

Figure 3: *Reference image and computed depthmap for Briskdalsbreen Glacier and Goldman Sachs Tower scenes. Warmer colors represent pixels closer to the camera. Note that in the Goldman Sachs Tower scene, the building under construction is reconstructed even though it is absent for part of the time-lapse.* *Photo credits:* `Daikrieg` *and* `Cebete`.

4 Locating time-lapses at planet scale

In this section we present a method to discover locations for mined time-lapse videos. These locations correspond to camera viewpoints that, due to the prominence of the depicted scenes, have been photograped from a similar viewpoint repeatedly over time by many different tourists.

We pose the problem of discovering time-lapse viewpoints as finding clusters of images that feature the same subject from similar viewpoints. We first cluster the photos based on their geolocations into *landmarks* and for each landmark we compute 3D reconstructions using Structure-from-Motion techniques [Agarwal et al. 2011]. Note that a landmark may have several disjoint reconstructions, e.g., inside vs. outside.

To find popular viewpoints within a 3D reconstruction, we use the canonical view approach of [Simon et al. 2007]. Their approach works by analyzing SIFT feature co-occurrences to partition the set of images into groups of photos with similar content and viewpoint. Representative images are then chosen for each group, by finding images that share the most features within the group. We compute the 20 highest ranked reference images (canonical views) for each 3D model.

For each reference image I_R, we find "nearby" images $\{I_i\}$ with similar viewpoints and directions, satisfying the following criteria:

- the optical axis is within α degrees of the reference viewpoint direction and,
- the camera center is located within a radius $R = tan(\alpha) \cdot \overline{d}$ of the reference image camera center, where $\overline{d}$ is the average distance from the reference camera center to 3D locations of image features visible in the reference image,

where α is a camera inclusion threshold.

Finally, we filter all candidate time-lapses that contain fewer than 300 timestamped images. Note, that two different candidate time-lapses from the same landmark might overlap in the photos they include.

Figure 2 shows the discovered reference image of the Briskdalsbreen Glacier time-lapse and the camera centers of the nearby images as green points. Note the tongue of the glacier being occluded by the landscape in the bottom right image, which is discarded by our proximity constraint.

5 Geometric stabilization

In this section we describe how to correct the photos for different viewpoints with respect to the reference image. If the scene is nearly planar, a homography could be sufficient to warp image I_i into reference image I_R. We can compute such homographies by using RANSAC on projections of the 3D tracks in the SfM model from camera C_i to camera C_R. This baseline method works well for scenes without parallax, but as expected, is not able to stabilize scenes with larger depth variations.

To account for parallax, we compute a depthmap D from the viewpoint of the reference image I_R. Although changes in the scene geometry over time are not modeled, we found that computing one global depthmap for the whole sequence provides adequate alignment for most scenes.

We use a temporal version of the classical plane sweep multi-view stereo algorithm [Kang and Szeliski 2004], modified to account for changing scene geometry and occluders like people. The main idea is to compute matching costs between images that were taken close in time. As with classical plane sweep, we generate a set of fronto-parallel depth planes with respect to the reference image I_R to compute matching costs. We discard the nearest and farthest 1% of 3D SfM points as well as 3D points with triangulation angles of less than 2 degrees, and evenly distribute enough depth planes (with a maximum of 200) over the depth range of the remaining 3D points to cover all disparity values.

We now define our temporal matching cost. Traditionally, stereo methods choose a reference image and only compute matching costs against that image. This does not work for our scenes as the scene geometry is changing over time. Instead, we compute a matching cost for each image as reference, using only images with nearby timestamps for matching, and then compute the overall cost as the median of costs over time, as described next.

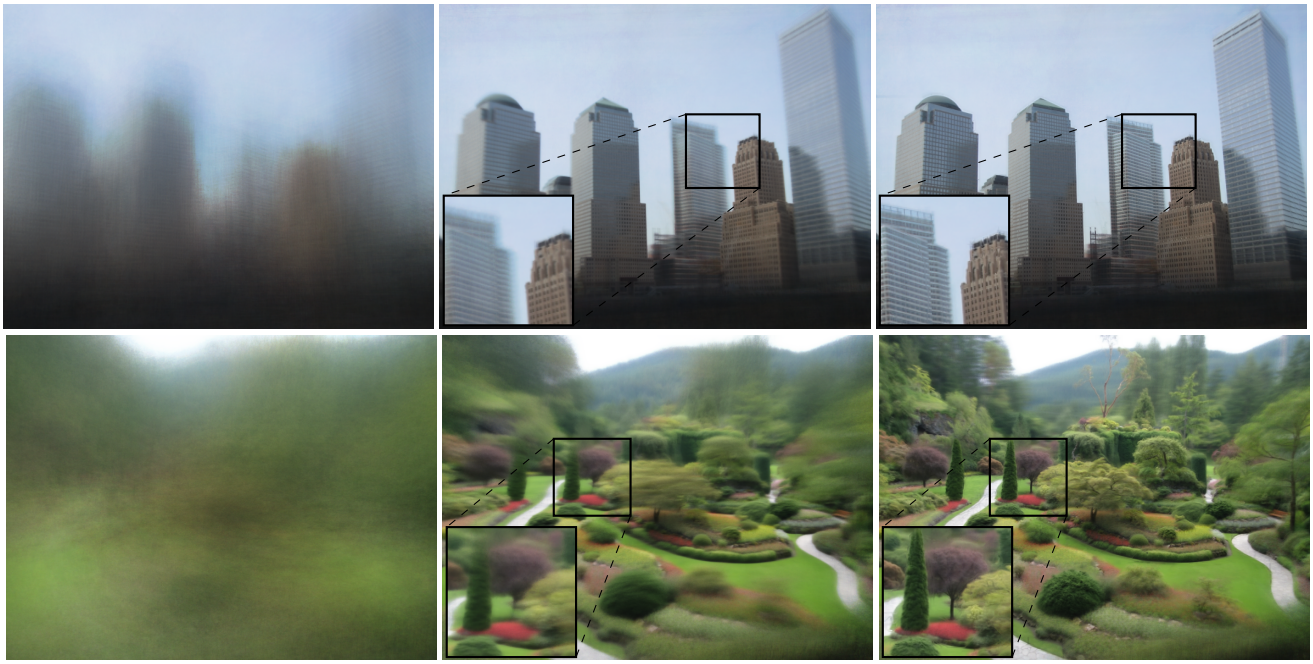
Given the sequence of input images $(I_1, \ldots, I_n)$ ordered by timestamp, the per-image cost $C_d^i(p)$ for pixel p at depth d at timestamp i is defined as

$$C_d^i(p) = \mathbf{median}_{j \in [i-T, i+T]} NCC_d(i, j, p) \qquad (1)$$

where $NCC_d(i, j, p)$ is the normalized cross correlation of a patch of size $K = 7$ around p of the projections of images i and j to the depth plane d, and $T = 20$ is the temporal window size. The overall cost is then

$$C_d(p) = \mathbf{median}_{i \in [1,n]} C_d^i(p). \qquad (2)$$

We compute a smooth depthmap D by using a standard MRF formulation where the data term for each plane is the matching cost C_d described above and the spatial term is a truncated L_1 distance [Boykov et al. 2001]. We used a spatial term weight of 0.2 and a truncation parameter of 4 disparity values. Figure 3 shows the resulting depthmaps for two sites.



(a) Median of all images (b) Median with homography alignment (c) Median with stereo alignment

Figure 4: *Stabilization results for two different scenes, Goldman Sachs Tower (top) and Butchart Gardens (bottom). Aligning the images with depth (c), produces a sharper composite compared to homography (b), or no alignment (a).*

Finally, we compute the warped images I_i^w by projecting each image into the reference camera C_R. For each pixel in I_R, we find its correspondence in I_i by using the depthmap to infer its 3D position and projection into I_i, using z-buffering to account for occlusions. We inpaint occluded pixels whose projection falls inside the image boundary of I_i using [Telea 2004].

Figure 4 compares stabilization techniques. We test two methods, stabilization with homographies and the proposed stabilization with stereo, and compute for each the median image of the stabilized sequence. We also show the median of all input images (without stabilization) for comparison. The stereo method produces significantly sharper results.

6 Appearance stabilization

In this section we describe how to stabilize the appearance of the warped images to correct for different lighting conditions and occluders. We formulate this task as computing an output time-lapse video frame for each warped image $I_i^w(p)$.

One effective approach for removing noise is median filtering. [Bennett and McMillan 2007] apply a temporally moving median filter to the frames of a time-lapse video. We adapt this method to the warped image sequences by computing the median of the valid pixels in the warped images, i.e., the pixels whose projection into the input image camera lies within the image frame. We found that large temporal windows are needed to reduce flicker but also result in oversmoothed transitions.

To address this drawback, we introduce a new temporal regularization approach. For each pixel, the goal is to compute its RGB value over time, by regularizing the pixel values of the warped sequence. Let $x_i = I_i^w(p) \in [0,1]^3$ be the RGB value in the warped image i, and let y_i be the RGB value in the output frame that we wish to compute. We optimize the following:

$$\min_{y_1,\ldots,y_n} \sum_{i|x_i \neq \emptyset} \delta\left(\|y_i - x_i\|\right) + \lambda \sum_i \delta\left(\|y_{i+1} - y_i\|\right) \quad (3)$$

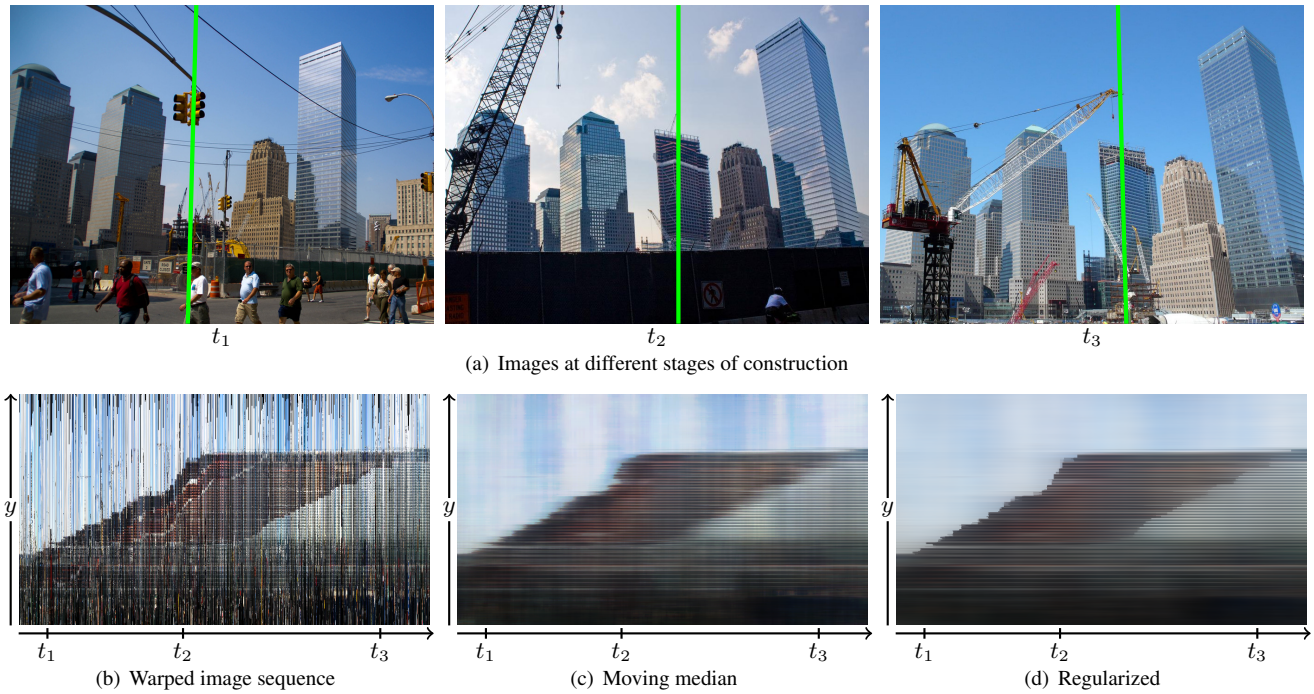
where $\delta(\cdot)$ is a loss function, λ is a temporal smoothing coefficient and $x_i = \emptyset$ when p corresponds to pixel coordinates outside the image boundary of I_i, i.e., has no correspondence in I_i. As for occluded pixels, we found that inpainting them works better than treating them as missing because they appear consistently around depth discontinuities and our temporal regularization operates only on a per-pixel basis, i.e., lacks a spatial regularization term. Figure 6 shows the effects of inpainting in the warped images and the resulting artifacts in the output frames around depth discontinuities if not used.

We experimented with several loss functions, including L_1 and L_2. We found L_2 works best for smooth transitions, whereas L_1 behaves better at discontinuities; we obtained best results with Huber, a robust loss function that is L_2 near 0, and L_1 elsewhere. Figure 5 compares a moving median with our Huber approach and shows the advantage of our method (easier seen in the video).

7 Planet scale time-lapse results

We mined time-lapses from 86M public geolocated photos from Picasa and Panoramio. We clustered 120K different landmarks and computed 755K 3D reconstructions. We then discovered 10,728 time-lapses across 2942 landmarks, that contain more than 300 images, using the camera selection criteria of $\alpha = 10$ degrees. We mined the time-lapses on a cluster with over 1000 nodes.

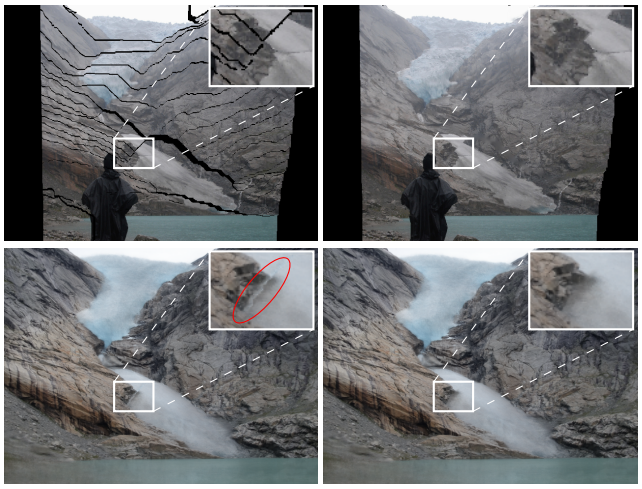
Figure 7 shows that the discovered time-lapses cover the globe and follow a similar distribution as publicly available Internet photos [Hays and Efros 2008]. Figure 8 shows a histogram of the length of the discovered time-lapses. A view of London from Greenwich Park contains several of the longest time-lapse sequences, with more than 10K photos each.



(a) Images at different stages of construction

(b) Warped image sequence (c) Moving median (d) Regularized

Figure 5: *Appearance stabilization for the Goldman Sachs Tower scene. (a): 3 sample images of the sequence showing the building at different stages of construction, with the pixel column of the y-t profiles highlighted. (b): y-t profile of the warped image sequence, showing this pixel column over time (moving to right). (c): result of temporal moving median filter of width* 80 *[Bennett and McMillan 2007]. (d): result of our proposed temporal regularization with smoothing coefficient* $\lambda = 100$*. Moving median blurs the transitions and has more flickering, particularly in the sky pixels. Photo credits: Zack Lee,* `ToastyKen` *and* `Cebete`*.*



(a) Without inpainting (b) With inpainting

Figure 6: *Effects of inpainting in the resulting time-lapses. Top: Warped images without inpainting (left) and with inpainting (right). Occluded pixels and pixels outside the input frustrum are shown in black. Note only occluded pixels are inpainted. Bottom: Corresponding frames of the resulting time-lapses. Note the artifacts around the depth discontinuities without inpainting. Photo credits: Nadav Tobias.*

To compute the final time-lapses, we subsampled time-lapse candidate locations containing more than 1000 photos, by choosing the 1000 closest images under our camera selection criteria. We generated time-lapse videos at a resolution of 1200 pixels in its larger dimension and set the temporal regularization parameter to $\lambda = 100$ for all sequences. We set the scale parameter of the Huber loss in Equation 3 to $4/255$ for the data term, i.e., 4 pixel values, and to $1/255$ for the temporal term. We use Ceres Solver [Agarwal et al.] to solve for the temporal appearance independently per color channel. Although our depthmap parameter choices worked reasonably well for most scenes, we fine-tuned the depthmap estimation parameters for some of the sequences in the video, in particular, the size of the NCC filter and the weight of the spatial term.

For efficiency, we computed depthmaps at lower spatial (800 pixels) and temporal (500 images) resolution. To generate final time-lapse videos, we play back the regularized output frames at a rate of 120 frames per second (subsampled by 4x to achieve 30fps), meaning that time proceeds at a rate proportional to the rate of photos taken.

A typical time-lapse with 1000 input posed photos takes about 6 hours to compute on a single machine, split equally between viewpoint and appearance stabilization. SfM reconstruction of 1000 photos takes 16 hours for matching and 1 hour for reconstruction with VisualSFM [Wu 2011]. While the algorithms can be optimized a lot more for efficiency, we point out that a few hours is negligible compared to the time period of several years it took to capture the photos.

For the special case of Briksdalsbreen Glacier, we expanded the time-lapse with more online photos, as our sequence contained few recent photos. We downloaded images from Flickr using a manually specified query, e.g., "Briksdalsbreen Glacier", and added them to the reconstruction using 2D-to-3D matching techniques to register the images.

Our time-lapses cover a broad range of interesting transformations:

- **Construction**: from individual buildings to whole skylines. The time-lapse of the Goldman Sachs Tower (Figure 9(a)),

Figure 7: *Map of the location of discovered time-lapses. Europe contains the highest density of time-lapses, while few exist in Africa and South America, as there are fewer photos available.*

shows the building rise from the ground, followed by windows coming in.

- **Changing cities**: smaller changes in the appearance of cities, like billboards or changes in urban elements, like sidewalks, etc. (see video).
- **Vegetation**: plants and trees growing, like the trees in the Butchart Gardens (see video).
- **Waterfalls**: we found that waterfalls are constantly changing, as branches dry up and new ones appear (Figure 9(b)).
- **Renovations**: monuments being renovated, like the Basilica of St. Maria of Salute in Venice, in Figure 9(c).
- **Seasons**: seasonal changes, like the blooming cycles of the flowers in Lombard Street (Figure 9(d)).
- **Geological changes**: retreating glaciers, erosion or, like in Figure 9(e), the growth of a hot spring in Yellowstone due to the deposit of minerals.
- **Stationary**: some scenes are interesting because of how little they change. For example, the Swiss Guard is so still, that it becomes part of the time-lapse of an entrance to the Vatican (see video).

We evaluated a random subset of 500 time-lapses for 1) reconstruction quality, rating them as "good" or "bad", and 2) interestingness, either interesting or not interesting. We found about 45% of the discovered time-lapses to be both good and interesting, 14% only good and 25% only interesting.

See the supplemental video for more examples.

7.1 Failure modes

We observed a number of interesting failure modes in our system. As noted by 4D cities, timestamps of online photos are not accurate. When many photos are incorrectly timestamped, our regularization approach can generate spurious halos, like in the second inset of the Goldman Sachs Tower (Figure 9(a)).

The time-lapse of Las Vegas (see video), shows blurring in areas where the geometry changes significantly over time. Generating time-varying depthmaps from unstructured photo collections is an exciting direction for future work.

In other cases, the 3D reconstruction (SfM or stereo) fails. For example, in the Mendelhall Glacier scene (see video), some cameras are registered to features on the moving glacier and our time-lapse video fails to stabilize the background. Such scenes pose a special challenge, as they break the assumptions in Structure-from-Motion systems.

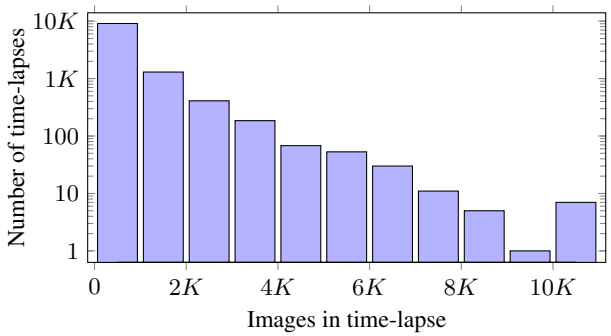


Figure 8: *Histogram of number of cameras in the discovered time-lapses, ranging from 300 to 10953 photos.*

Another limitation of our system are scenes whose recovered 3D models contain both day and night photos. The synthesized time-lapses show an unrealistic "twilight" effect that averages the day and night photos and flickers over time, as seen in the Hong Kong skyline time-lapse (see video).

Our depthmaps are inaccurate in regions that are known to be challenging for stereo algorithms, such as oblique surfaces, like ground planes, clutter or occlusions, like busy squares, or thin structures.

Addressing these limitations is a great topic for future work.

8 Conclusion

We introduced an approach to mine time-lapses from Internet photos. Our system discovered 10,728 time-lapses that show how the world's most popular landmarks are changing over time. Our method stabilizes the time-lapse video sequence so that the underlying changes in the scene become visible. The depicted changes include buildings under construction, glaciers retreating, plants growing, seasonal changes, and many geological processes.

The scale and ubiquity of our mined time-lapses creates a new paradigm for visualizing global changes. As more photos become available online, mined time-lapses will visualize even longer time periods, showing more drastic changes.

Photo credits

We thank Flickr user `dration`, Zack Lee, Nadav Tobias, Juan Jesús Orío and Klaus Wißkirchen for allowing us to reproduce their photographs. We also acknowledge the following Flickr users whose photographs we reproduced under Creative Commons license[1]: `Aliento Más Allá`, `jirihnidek`, `mcxurxo`, `elka_cz`, `Daikrieg`, `Free the image`, `Cebete` and `ToastyKen`.

References

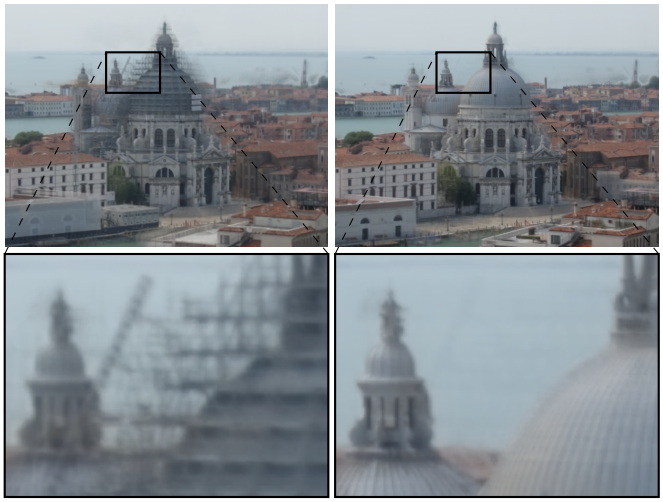
ABRAMS, A., MISKELL, K., AND PLESS, R. 2013. The episolar constraint: Monocular shape from shadow correspondence. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, 1407–1414.

ACKERMANN, J., LANGGUTH, F., FUHRMANN, S., AND GOESELE, M. 2012. Photometric stereo for outdoor webcams. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, IEEE, 262–269.

[1] `https://creativecommons.org/licenses/by/2.0/`

(a) Goldman Sachs Tower, New York City, USA

(b) Galovac Waterfall, Plitvice Lakes, Croatia



(c) St. Maria of Salute, Venice

(d) Lombard Street, San Francisco, USA

(e) Mammoth Hot Springs, Yellowstone, USA

Figure 9: *Selected frames of mined time-lapses, showing different phenomena captured. (a) Several phases of the construction of the Goldman Sachs Tower. (b) New branch appears in Galovac Waterfall. (c) Renovation of the St. Maria of Salute Basilica. (d) Blooming of flowers in Lombard Street. (e) Hot spring terraces in Yellowstone grow and change color due to the deposition of minerals.*

AGARWAL, S., MIERLE, K., AND OTHERS. Ceres Solver. http://ceres-solver.org.

AGARWAL, S., FURUKAWA, Y., SNAVELY, N., SIMON, I., CURLESS, B., SEITZ, S. M., AND SZELISKI, R. 2011. Building rome in a day. *Communications of the ACM 54*, 10, 105–112.

AMIRSHAHI, H., KONDO, S., ITO, K., AND AOKI, T. 2008. An image completion algorithm using occlusion-free images from internet photo sharing sites. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci. E91-A*, 10.

APPLE. Apple iOS8 camera.

BAE, S., AGARWALA, A., AND DURAND, F. 2010. Computational rephotography. *ACM Trans. Graph. 29*, 3 (July), 24:1–24:15.

BENNETT, E. P., AND MCMILLAN, L. 2007. Computational time-lapse video. In *ACM SIGGRAPH 2007 Papers*, ACM, New York, NY, USA, SIGGRAPH '07.

BOYKOV, Y., VEKSLER, O., AND ZABIH, R. 2001. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 23*, 11, 1222–1239.

EXTREME ICE SURVEY. http://extremeicesurvey.org.

HAYS, J., AND EFROS, A. A. 2007. Scene completion using millions of photographs. *ACM Transactions on Graphics (SIGGRAPH 2007) 26*, 3.

HAYS, J., AND EFROS, A. A. 2008. im2gps: estimating geographic information from a single image. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

INSTAGRAM. Hyperlapse app.

JACOBS, N., BIES, B., AND PLESS, R. 2010. Using cloud shadows to infer scene structure and camera calibration. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 1102–1109.

KANG, S. B., AND SZELISKI, R. 2004. Extracting view-dependent depth maps from a collection of images. *International Journal of Computer Vision 58*, 139–163.

KARSCH, K., GOLPARVAR-FARD, M., AND FORSYTH, D. 2014. ConstructAide: Analyzing and visualizing construction sites through photographs and building models. *ACM Trans. Graph. 33*, 6 (November).

KEMELMACHER-SHLIZERMAN, I., SHECHTMAN, E., GARG, R., AND SEITZ, S. M. 2011. Exploring photobios. In *ACM SIGGRAPH 2011 Papers*, SIGGRAPH '11, 61:1–61:10.

LAFFONT, P.-Y., BOUSSEAU, A., PARIS, S., DURAND, F., AND DRETTAKIS, G. 2012. Coherent intrinsic images from photo collections. *ACM Transactions on Graphics (SIGGRAPH Asia Conference Proceedings) 31*.

LAFFONT, P.-Y., REN, Z., TAO, X., QIAN, C., AND HAYS, J. 2014. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Transactions on Graphics (proceedings of SIGGRAPH) 33*, 4.

MATZEN, K., AND SNAVELY, N. 2014. Scene chronology. In *Proc. European Conf. on Computer Vision*.

RUBINSTEIN, M., LIU, C., SAND, P., DURAND, F., AND FREEMAN, W. T. 2011. Motion denoising with application to time-lapse photography. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, 313–320.

SCHINDLER, G., AND DELLAERT, F. 2010. Probabilistic temporal inference on reconstructed 3d scenes. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 1410–1417.

SCHINDLER, G., DELLAERT, F., AND KANG, S. B. 2007. Inferring temporal order of images from 3d structure. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, 1–7.

SHIH, Y., PARIS, S., DURAND, F., AND FREEMAN, W. T. 2013. Data-driven hallucination of different times of day from a single outdoor photo. *ACM Trans. Graph. 32*, 6 (Nov.), 200:1–200:11.

SIMON, I., SNAVELY, N., AND SEITZ, S. M. 2007. Scene summarization for online image collections. *ICCV 7*, 1–8.

SUNKAVALLI, K., MATUSIK, W., PFISTER, H., AND RUSINKIEWICZ, S. 2007. Factored time-lapse video. In *ACM SIGGRAPH 2007 Papers*, ACM, New York, NY, USA, SIGGRAPH '07.

TANEJA, A., BALLAN, L., AND POLLEFEYS, M. 2011. Image based detection of geometric changes in urban environments. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2336–2343.

TANEJA, A., BALLAN, L., AND POLLEFEYS, M. 2013. City-scale change detection in cadastral 3d models using images. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, 113–120.

TELEA, A. 2004. An image inpainting technique based on the fast marching method. *Journal of Graphics Tools 9*, 1, 23–34.

ULUSOY, A. O., AND MUNDY, J. L. 2014. Image-based 4-d reconstruction using 3-d change detection. In *Computer Vision–ECCV 2014*. Springer, 31–45.

WHYTE, O., SIVIC, J., AND ZISSERMAN, A. 2009. Get out of my picture! internet-based inpainting. In *Proceedings of the 20th British Machine Vision Conference*.

WU, C., 2011. VisualSFM: A visual structure from motion system. http://ccwu.me/vsfm.