



**HAL**  
open science

# Reanimating aerial images in real time for geospatial web applications

Alexandre Devaux, Mathieu Brédif

► **To cite this version:**

Alexandre Devaux, Mathieu Brédif. Reanimating aerial images in real time for geospatial web applications. Journées Françaises d'Informatique Graphique, Oct 2017, Rennes, France. hal-01893899

**HAL Id: hal-01893899**

**<https://hal.science/hal-01893899>**

Submitted on 11 Oct 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Reanimating aerial images in real time for geospatial web applications

Alexandre Devaux, Mathieu Bredif

Univ. Paris-Est, LASTIG MATIS, IGN, ENSG, 73 avenue de Paris, F-94160 Saint-Mande, France  
alexandre.devaux@ign.fr, mathieu.bredif@ign.fr

---

## Abstract

*Geovisualization has concentrated most work in displaying static data for a long time, progress have brought incredibly detailed visualizations offline but also in web browsers thanks to GPU access. Heavy textured meshes mixed with vectorial data are common representations for users now. Animation, on the other hand, is just at its beginning. As far as we know there are no current applications that integrate imaged-based dynamic visualization of many distinct objects at different scales. We propose a method generating dynamic representations for objects present in the aerial image that change appearance or position over time such as cars, water, boats, vegetation without the need for a user intervention. It offers a compelling experience, emphasizing object class differentiation. This research involves two main tasks. First to be able to extract object layers and direction using aerial images and land covers information in real time, and secondly to animate these objects realistically, combining computer vision and computer graphics domains.*

---

**keywords :** animation, image-based animation, image-based rendering, geovisualization, aerial imagery, gis, web mapping

## 1. Introduction

The last decade has seen the explosion of geographic applications such as the popular Google Maps<sup>®</sup> with more than a billion users as of 2017. In all those years the quality of the data and the rendering never stopped improving with very high resolution aerial imagery creating high resolution ortho-photos (geometrically corrected image such that the scale is uniform) and textured 3D models. Users can now see the world with precision, mix raw data with vector data and so on.

However, if the quality of the rendering is improving, it is still a fixed representation. Trees, cars, rivers don't move as they would in reality, they are stuck at the time of the photography shot. Cesium [ces] gives the possibility to manipulate time, you can for example follow the trajectory of a satellite in real time or at a specific time. Projects such as MapStyle [CDT\*16] offer to control the style of the representation, sea level for example can be tuned to evolve dynamically following its parameterized cycle. The subject we tackle here is to give the impression of animation hence life in a web mapping application. We could create illusory motion using repeated asymmetric patterns [CLQW08] which uses fixed patterns but it would modify too much the aerial image and we want to keep its full richness for mapping applica-

tions. Some research has been done on picture animation such as Chuang et al. [CGZ\*05] which offers compelling results on photos after an important part of manual settings and they limit their domain to scenes containing passive elements that respond to natural forces (wind) and work on a single image. In this paper we don't restrict on periodic motion caused by wind as one the most sensible aspect of motion from aerial images will be given by moving cars. Plus, we don't take a single image as an input but a moving view from a 3D scene. The point of view can change, the distance to the same region, etc. Also a difference from [CGZ\*05] is that we have more than one motion field per layer. The purpose of this paper is to enhance web mapping visualization, it doesn't address simulation rendering such as real time traffic visualization or realistic climate rendering but can be parameterized in order to give an idea on those conditions. Other than only enhancing aesthetic, it worth noting that adding movements to mobile objects also adds readability to the maps. To give the impression of a living image in a mapping context, it should work at different scale level, from the whole city to a district. We want to achieve this by animating existing objects in the image.

### 1.1. Related Work

One could think that it is easier to add external animated objects templates over the original image. It is partly true as it will remove the need for real time inpainting and for object detection. However, for this purpose the original data should have been previously cleaned (cars should be remo-

ved from the streets and boats from rivers) implying an off-line process to detect objects to have an appealing rendering where cars don't run over other cars. And a real concern as a mapping application is the necessity to conserve styles and conditions of the original aerial image, hence we cannot use external objects. Working only with original image features also brings the possibility to pause the animation rendering or cancel it to see the motion going backward to its original position. The objects that will be affected by motion effects are in order of priority : cars/bus, boats, water, clouds, trees. This is mainly due to their apparent speed, creating motions that attract the human eye.

We investigated different approaches to classify objects in the aerial image such as recent literature [VT17] has shown the superior performance of convolutional neural networks (CNN) for many classification tasks including the automatic analysis of Very High Spatial Resolution (VHR) geospatial images but it is impossible to consider it for real time computing in a web browser. The only possibility would be to deploy the heavy CNN on a powerful server and send him images and get the result back to the client but again, it would slow down the navigation plus we don't need high precision nor high recall. We then tried faster classification methods such as Viola-Jones [VJ04] Haar classifier with boosting with trained classifiers for cars. The solution is fast enough to run in near real time in a web browser but the results were not convincing mainly since the classifier works for specific orientations of cars. Another solution [XYW\*16] integrates the Viola-Jones (V-J) and linear SVM classifier with HOG feature (HOG + SVM) methods for vehicle detection from low-altitude unmanned aerial vehicle (UAV) images but the implementation is heavy for real time web browser computation plus it doesn't work when features are too small which is necessarily the case when looking a whole city from top.

Although it might appear logical to have a detection pass of all objects in the scene to animate them separately we discovered that combining objects per class layers works fine especially for cars. This is a particularly important point, the method, to create the illusion of movement, doesn't need to clearly identify object separately which offers another possibility : using land covers. Land covers describe the physical material at the earth surface (building, road, water, vegetation) and they are easily available with high precision around cities which is where most moving features are found making it the most interesting area for reanimation. It is also possible to use widely available rasterized topological tiles when land covers can't be found at a sufficient precision.

## 2. Method overview

Our method uses ortho-photos and land covers as inputs (Fig1) which are served on a world scale basis using a pyramid of multi resolution tiles. Having discarded the need for object-based segmentation we still need to know in which direction the motion should occur through the different layers. Different passes are computed in real time to compute the direction of the object layer at each pixel and also on a more global scale (the whole frame). The second part of our me-

thod concerns the animation rendering for which we create specific animations based on the object characteristics.

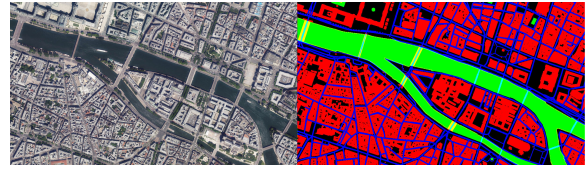


Figure 1: *Input Data : ortho-photos and land covers*

## 3. Axis Detection

For non-periodic motions, determining a coherent direction of the motion is critical. Cars, boats (and water), key objects in reanimating aerial images have a specific direction. As explained before, we don't separately move each cars and each boats but the whole roads and water, support to these features. If rivers usually have a more global motion ie the same motion direction for large areas, roads are much more geometrically diverse. To answer efficiently those two contexts we use a pixel-based axis computation and a window-based. For the first one we use ray casting around the current pixel rotating the rays until it intersects far enough (intersect anything else than the road) This gives us the direction of the road for each road pixel. It's quite fast as we don't need a precise estimation locally and it doesn't mix with surrounding roads.

For the boats motion we detect the river direction using a principal component analysis (PCA) on the frame land cover texture using water class pixel. This solution works usually fine but in some rare cases if many small rivers with different directions are present in the same texture the results won't have any sense. Using the previous solution is not adapted because rivers are much larger objects, a per-pixel direction offers too much variation locally inducing deformation of the boats. Fig2 shows computed vector directions in a color representation for roads and rivers

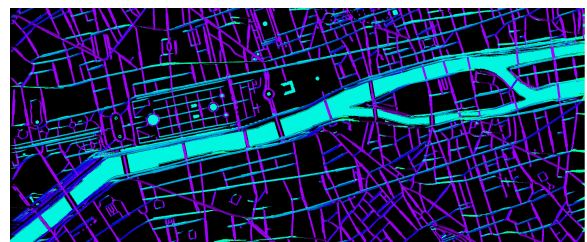
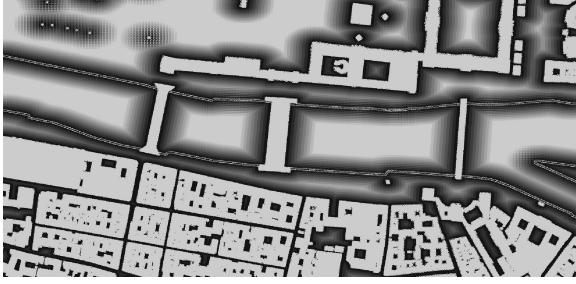


Figure 2: *Computed motion field*

As the input land cover texture is not very precise, to avoid objects layers to be misclassified on the borders we compute a distance texture for water layer. The texture gets a value depending on the distance to any other object class (Fig3). This will later allow to have smooth blending between layers segmentation using the distance to mix textures.



**Figure 3:** Distance Map. The grey level value represent the distances between object classes

The vector field computation step is not free of artifacts. It's worth mentioning that we are not looking for a perfect motion of all objects in the frame but a coherent motion through the whole rendered frame. Avoiding incoherences is critical because it is much more noticeably unpleasant as the human eye gets more attracted to moving objects. One of them concerns trees that overlap roads, they get moved along with cars. As we have no information on sparse trees from the land cover we added a simplistic detection pass based on color information in the ortho-photo so we remove those pixels from the road. Another visible artifact concern roads overlapping other roads creating strange cars crossing bridges perpendicularly. We have no information in the land cover about which one is over the other so it would need some more analysis directly on the ortho-photo every time a road crosses another one.

#### 4. Animation

The second part of our method concerns the animation itself (Fig4). Since it takes place in a 3D geospatial application, the user can change its viewing position at any frames inducing the constraint to work with multiple scales with consistency and being able to handle separate tiles. Also to keep a high framerate, animation rendering should be light and asynchronous with the vector field computation. In this paper we concentrate on non-periodic motion which is feature dependent. We describe in this section our approach to render the different motions.



**Figure 4:** Comparing original image (up) and captured motion rendering (down).

#### 4.1. Cars and Roads

This is the most sensible feature to animate, the first the user will perceive. As explained earlier we don't move each cars separately but we move each pixels of the road. It allows cars to follow the road path and to avoid overlapping other cars. As we know the size of a pixel in meters it is possible to make cars move at a specific realistic speed. Current texture coordinates are computed as :

$$road(u, v, t) = (u, v) + \frac{T}{2} g\left(\frac{t}{T}\right) f(u, v) s(u, v) \quad (1)$$

where  $T$  is the loop duration,  $f$  is the vector field for this feature and  $s$  is the motion speed defined, using the triangle wave  $g(x) = |x \bmod 4 - 2| - 1$ . Notice that the displacement is evaluated as an absolute displacement of the input image rather than a relative displacement of the previous image. In this way, repeated resampling and numerical error accumulation are avoided. The vector field coming from the ray casting is voluntarily not normalized so cars moves naturally faster in straight lines than in curves. An improvement to this technic would be to analyze if two roads are parallel then we could deduce the sense of motion.

#### 4.2. Boats and Water

Large cities of the world often get crossed by rivers. Their motion are very slow but still perceptible, Water reflection changes are even more visible. Boats also usually moves very gently in cities. This pushed us to use a simple unified approach handling all those effects.

The idea is pretty straightforward, similar to the road animation but we move sequentially each pixels in its vector direction depending on its coordinate on the screen. This asynchronous motion through the frame gives a specular effect animation on the water simulating ondulation, moves the water and also the boats with their coherent trails and wave. If the axis of motion is nearly always well computed, a significant drawback concerns the direction which is random. It clearly shows here that not having detected the direction of each boats has some limits. To cope with that we tried to automatically separate the river into two corridors along its axis to have two directions and follow the most representative left-right direction habits in the world but we realized that boats doesn't follow that simple rule used for cars.

$$water(u, v, t) = road(u, v, t) + noise(u, v, t) \quad (2)$$

$$noise(u, v, t) = water(u, v, t) \bmod \epsilon \quad (3)$$

where  $\epsilon$  is a very small value for the phase shift

#### 4.3. Vegetation

Plants and trees motions are caused by the wind. Having an aerial point of view gives the possibility to avoid modeling them as physical systems with mass and stiffness properties due to branches and simplify it with a set of leaves subject to a stochastic motion. We create the motion using Perlin noise on the leaves.





Figure 5: Captured frames showing boats and cars in motion

#### 4.4. Clouds

Clouds are different than all previously described objects as they don't belong to the original image (as an acquisition specification), they are not either in land covers obviously. But they can optionally be added as they give an interesting feeling of activity with a very easy processing. As the user can change the camera position and orientation in 3D and the fact that clouds have a different depth than the image features we cannot work directly in the original image texture. Instead, using Perlin noise we create turbulence textures that we map on different planes at different altitude each with a specific motion speed. Clouds move randomly around the viewing position very slowly and when the user moves to a distant destination clouds positions is then translated according to the viewing target. Clouds could also be added in the region where incoherent motions can occur such as crossing for cars to make them more inconspicuous.

$$cloud_i(x,y,z,t) += Random(x,y,z,t) \quad (4)$$

#### 5. Conclusion

We presented an innovative method to bring back to life aerial still images at multiple different scales in a web mapping application context without the need for any user interventions. Using only images and land covers as input we can animate differently objects as they move or change appearance through time. For each motion type, the user can also tune the motion parameters. Having a solution that runs in real time on the whole world is an encouraging result and a group of testers enjoyed navigating through mapping data that lives (Fig5). It still remains room for improvement. We have applied our system to several places on earth and the realism of the automatic motion is not always great as for example boats on the Paris river should go much slower than boats in Miami. When getting close to the ground, the segmentation/animation should be more precise. Toward image segmentation also, a real challenge would be to obtain the same results without land covers using only aerial images as input.

#### References

- [CDT\*16] CHRISTOPHE S., DUMÉNIU B., TURBET J., HOARAU C., MELLADO N., ORY J., LOI H., MASSE A., ARBELOT B., VERGNE R., BRÉDIF M., HURTUT T., THOLLOT J., VANDERHAEGHE D. : Map Style Formalization : Rendering Techniques Extension for Cartography. Bénard P., Winnemöller H., (Eds.), Non-Photorealistic Animation and Rendering, The Eurographics Association.
- [ces] Cesiumjs.
- [CGZ\*05] CHUANG Y.-Y., GOLDMAN D. B., ZHENG K. C., CURLESS B., SALESIN D. H., SZELISKI R. : Animating pictures with stochastic motion textures. In *ACM SIGGRAPH 2005 Papers* (New York, NY, USA, 2005), SIGGRAPH '05, ACM, pp. 853–860.
- [CLQW08] CHI M.-T., LEE T.-Y., QU Y., WONG T.-T. : Self-animating images : Illusory motion using repeated asymmetric patterns. *ACM Transactions on Graphics*. Vol. 27, Num. 3 (2008).
- [VJ04] VIOLA P., JONES M. J. : Robust real-time face detection. *Int. J. Comput. Vision*. Vol. 57, Num. 2 (mai 2004), 137–154.
- [VT17] VOLPI M., TUIA D. : Dense semantic labeling of subdecimeter resolution images with convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*. Vol. 55, Num. 2 (Feb 2017), 881–893.
- [XYW\*16] XU Y., YU G., WANG Y., WU X., MA Y. : A hybrid vehicle detection method based on viola-jones and hog + svm from uav images. *Sensors*. Vol. 16, Num. 8 (2016).