Digital Image Processing And Surface and Volumetric Data Rendering and Visualization

Dot. Andrea F. Bocchese

Dot. Sergio Leoni

YASVVM

Yet Another Street View Video Maker

Dot. Andrea F. Bocchese

Dot. Sergio Leoni

What is YASVVM?

Digital Image Processing And Surface and Volumetric Data Rendering and Visualization

Dot. Andrea F. Bocchese

Dot. Sergio Leoni

YASVVM is a WEB tool that permits to generate video of a route using Google Street View images.

- Define your Route
- Start the video generation
- Download & Watch the virtual trip

How YASVVM Works

Dot. Andrea F. Bocchese

Dot. Sergio Leoni

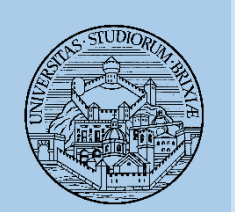
The YASVVM web frontend permits to the user to define the start and the end points of the route.
This is done using the **Google maps and street view API**.
In this phase all the coordinates of the street view pictures are computed.

The bunch of points computed are submitted to the YASVVM server that retrieves the images using the **Street view API**, and builds the video using an application developed with **OpenCV**.

In the case of the Motion Interpolation is also used an external application called **Butterflow[1].**

At the end of the video computation the video is available for the download.

1. https://github.com/dthpham/butterflow



YASVVM's Architecture

Digital Image Processing And Surface and Volumetric Data Rendering and Visualization

Dot. Andrea F. Bocchese

Dot. Sergio Leoni

- **Maps**:
  - Selection Map
  - Route Computing
- **Street View**:
  - Retrieve Street View points

Google Maps/Street View

Retrieve Street View Images

AJAX

Google Maps/StreetView API

Google StreetView API

YASVVM WEB Interface

AJAX

YASVVM API

YASVVM Server

- Create / Delete Jobs
- Start Video Elaboration
- Video downloads
- Percentage
- …

- OpenCV
- Butterflow
- FFmpeg

Web Frontend

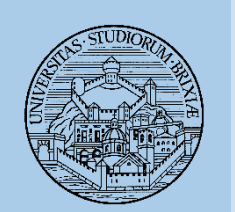
Digital Image Processing And Surface and Volumetric Data Rendering and Visualization

Dot. Andrea F. Bocchese

Dot. Sergio Leoni

YASVVM

The Web Frontend

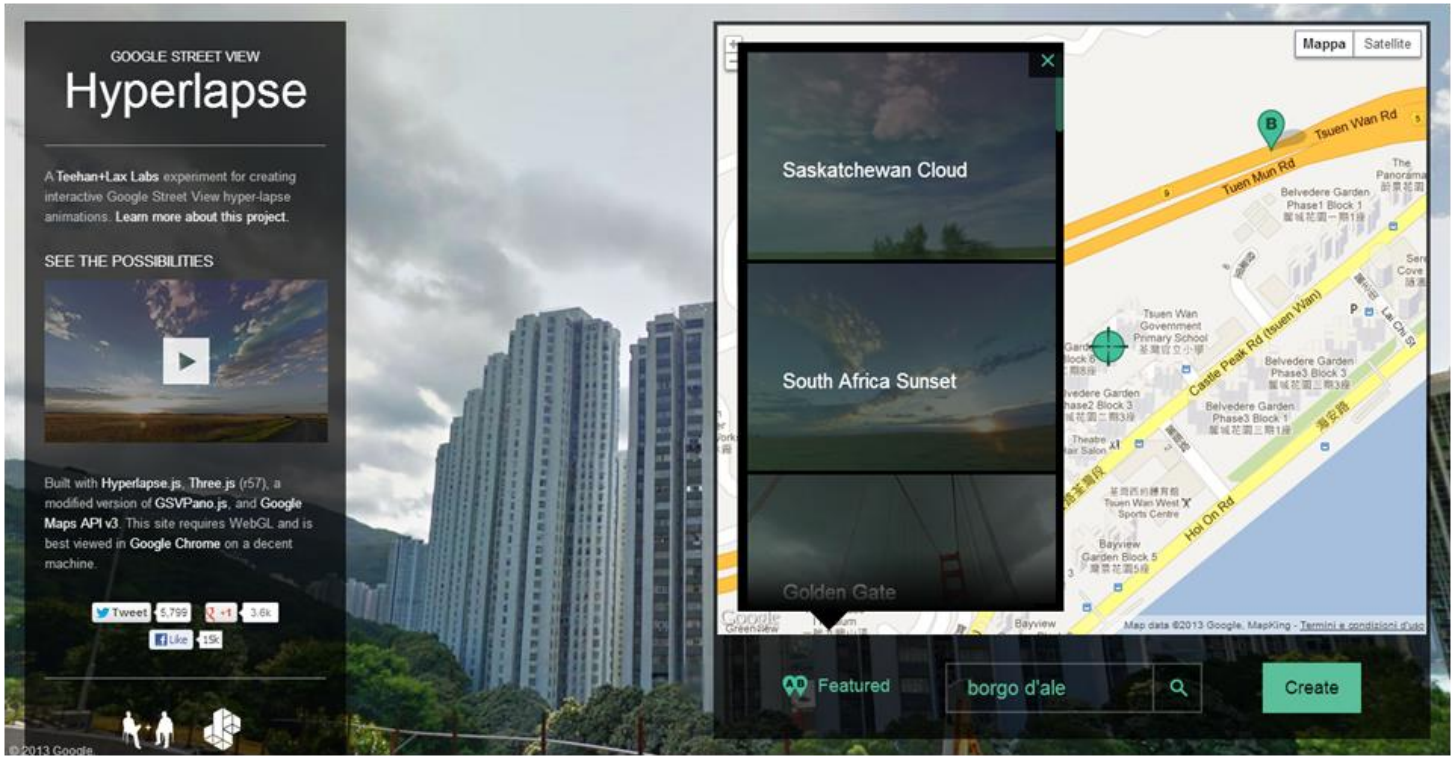


Getting inspired

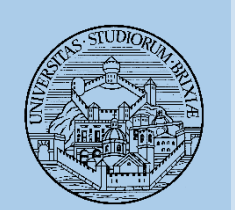
Digital Image Processing And Surface and Volumetric Data Rendering and Visualization

Dot. Andrea F. Bocchese

Dot. Sergio Leoni



https://github.com/TeehanLax/Hyperlapse.js/



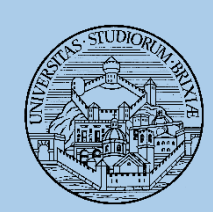
Getting inspired

Digital Image Processing And Surface and Volumetric Data Rendering and Visualization

Dot. Andrea F. Bocchese

Dot. Sergio Leoni

https://github.com/spite/GSVPano.js/



Getting inspired

Digital Image Processing And Surface and Volumetric Data Rendering and Visualization

Dot. Andrea F. Bocchese

Dot. Sergio Leoni

Google Street View Panorama Util

14 commits | 2 branches | 0 releases | 1 contributor

branch: master ▾ GSVPano.js / +

Bye :'(1 comment

spite authored on Sep 25, 2013 latest commit 50d6add859

README.md Bye :'(2 years ago

README.md

GSVPano.js - Google Street View Panorama lib

Library to help requesting and stitching Google Street View panoramas.

Where is it then?

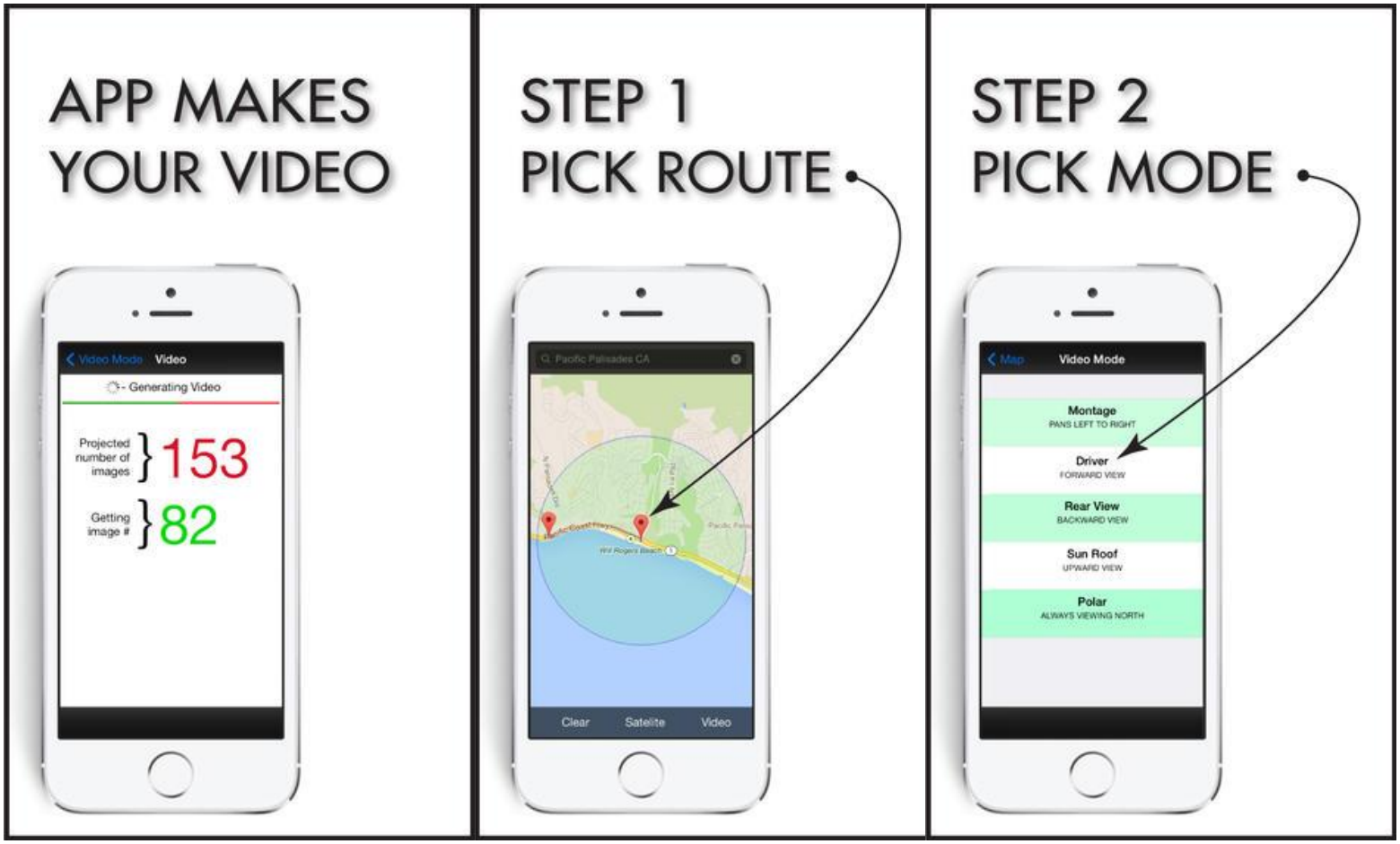
Sadly, I've been requested by the Maps team to remove the library, since it was enabling people to break Google's Terms of Service. Which is true, but also an intrinsic problem with the only purpose of the library. Let's hope in a very near future the Google Maps API will provide a very similar feature.

Getting inspired

Digital Image Processing And Surface and Volumetric Data Rendering and Visualization

Dot. Andrea F. Bocchese

Dot. Sergio Leoni

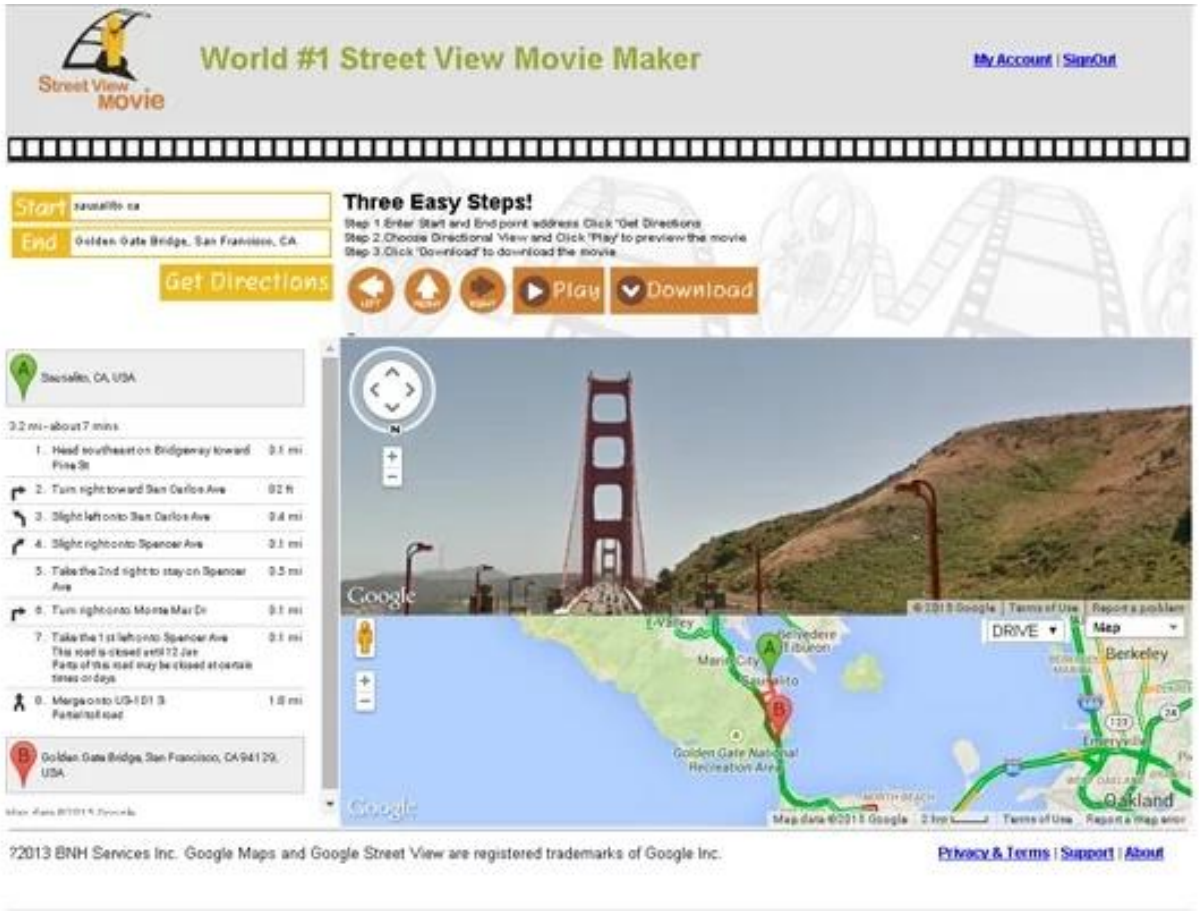


Getting inspired

Digital Image Processing And Surface and Volumetric Data Rendering and Visualization

Dot. Andrea F. Bocchese

Dot. Sergio Leoni



Getting inspired

Dot. Andrea F. Bocchese

Dot. Sergio Leoni

What we understand

- Street View API interaction has to be done locally (browser or app)
- Must use a set of Google API
- We probably violate Google API TOS

What we decide

- Exploring Google API from scratch
- Split points retrieving (browser) and video making (server)

Points retrieving

Digital Image Processing And Surface and Volumetric Data Rendering and Visualization

Dot. Andrea F. Bocchese

Dot. Sergio Leoni

Two distinct operations:

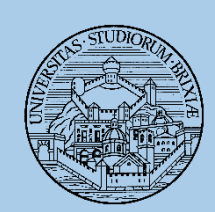
- Define a route between a starting point and an ending point
- Find, on that route, the street view images

One API, several APIs:

- Google Maps JavaScript API v3: contains different modules (services)
- Directions Service: allows user to calculate a route between a starting and ending point
- Street View Service: allows user to retrieve Street View information

Common limitations:

- Interaction limited to API
- Limited number of requests (Quota)
  - Google Maps JS API v3: 25,000 requests/day (max 1 request/second/user)
  - Directions API: 2,500 requests/day
  - Street View API: 25,000 requests/day
- API change



Directions Service Issues

Digital Image Processing And Surface and Volumetric Data Rendering and Visualization

Dot. Andrea F. Bocchese

Dot. Sergio Leoni

- Route request is async (ajax): when the request is completed a callback is executed, data are sent in json format

  requestData = { origin: …, destination: …, param: … };

  directionsService.route(**requestData** , function(data, status){…});

- Direction result is a polyline
  - Very dense where turn, no points on straigth line

  - Add intermediate points
  - Verify EACH point, and find Street View Image (PanoramaID) associated

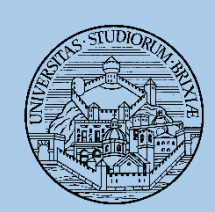
Street View Service Issues

Digital Image Processing And Surface and Volumetric Data Rendering and Visualization

Dot. Andrea F. Bocchese

Dot. Sergio Leoni

- Limited interaction:
  - getPanoramaById()
  - getPanoramaByLocation()
- getPanoramaByLocation()
  - ”Searches for panorama data over a given area, given a passed LatLng and the radius (in meters) over which to search. If the radius is 50 meters or less, the panorama returned will be the nearest panorama to the given location.”
  - Async = Call a callback after execution = a problem when we want to test ALL the point in a prestabilited order.
  - Street View Layer sometimes is not aligned to Direction Layer (solved in our case study)
  - 50 meters diameter is too much for our application
  - <50 meters diameter = nearest panorama produce strange results (on different execution, on same execution results are cached).

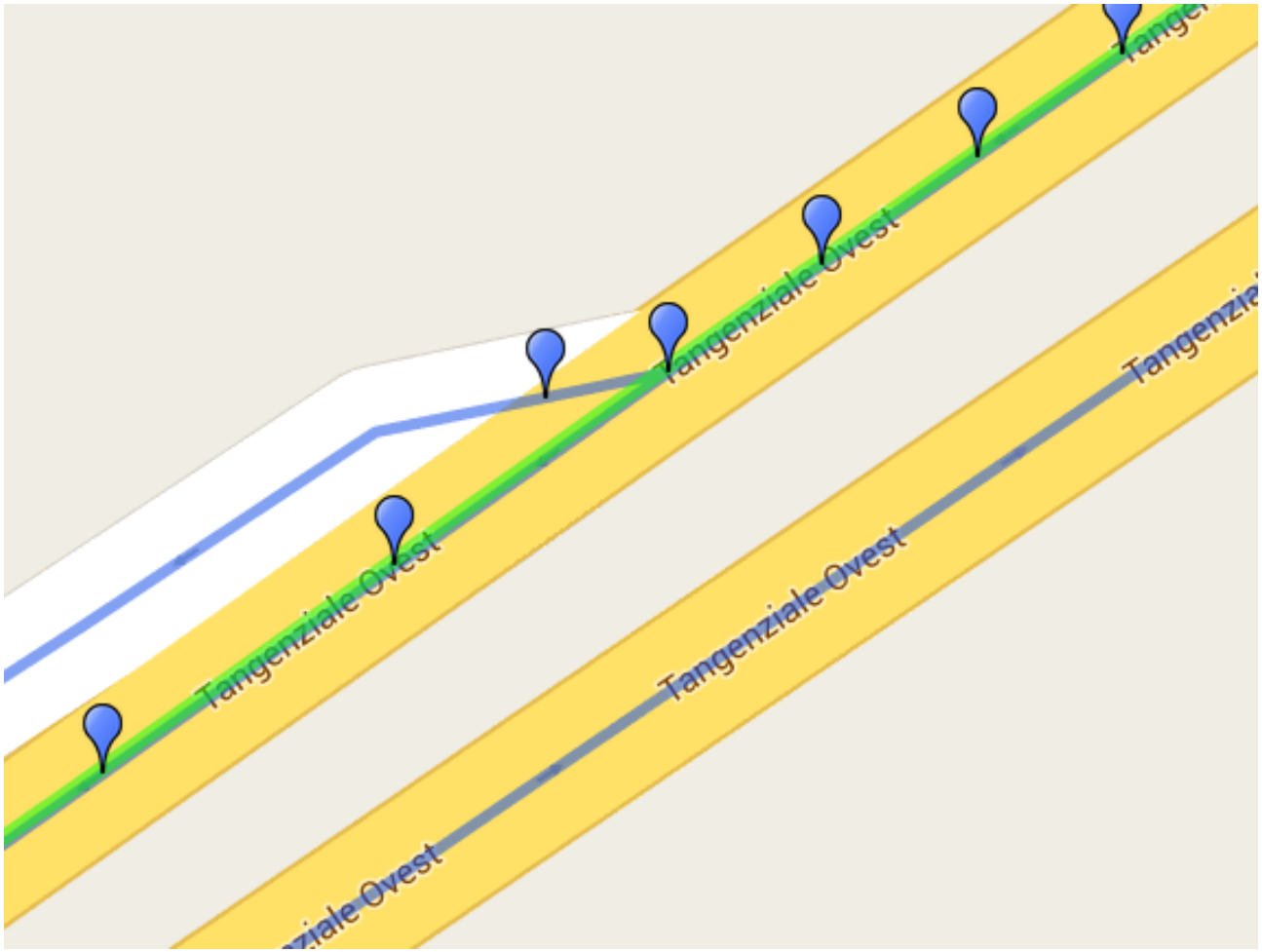


Street View Service Issues

Digital Image Processing And Surface and Volumetric Data Rendering and Visualization

Dot. Andrea F. Bocchese

Dot. Sergio Leoni



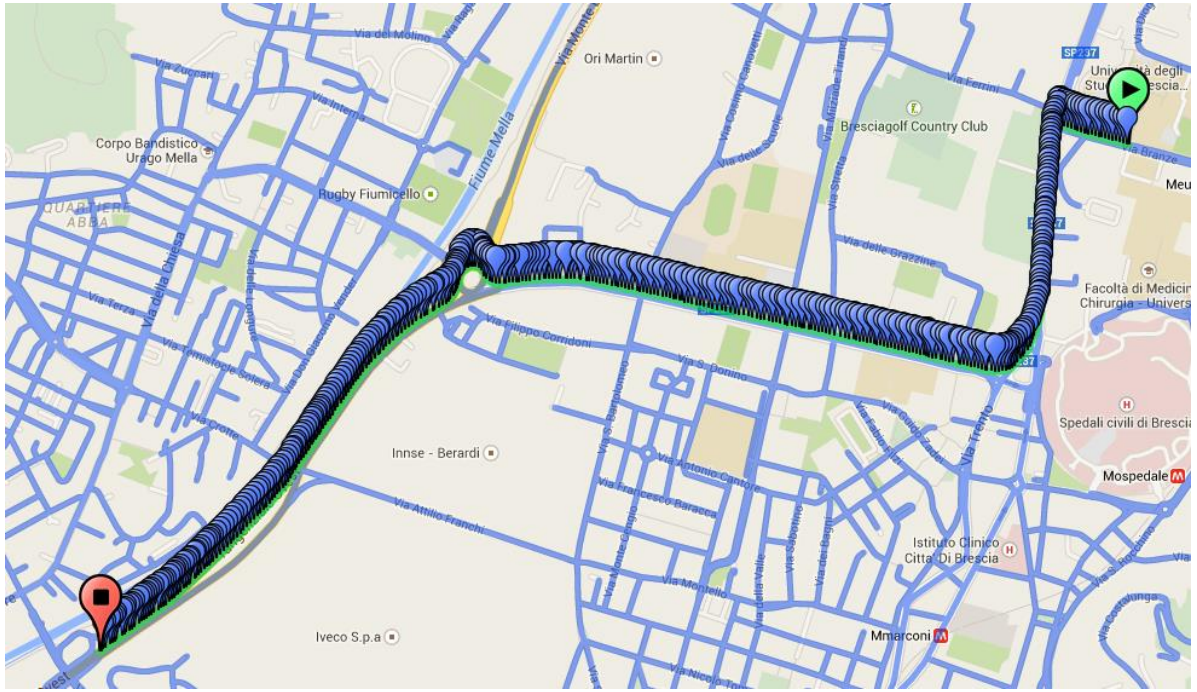
Street View Service Issues

Digital Image Processing And Surface and Volumetric Data Rendering and Visualization

Dot. Andrea F. Bocchese

Dot. Sergio Leoni

- Each point represents an unique PanoramaID
- A panorama is a 360° photo, taken in a certain position.
- You can retrieve only a single piece of panorama (each request)
  - Heading (pointing direction) discriminates Images with same PanoramaID



Street View Service Issues

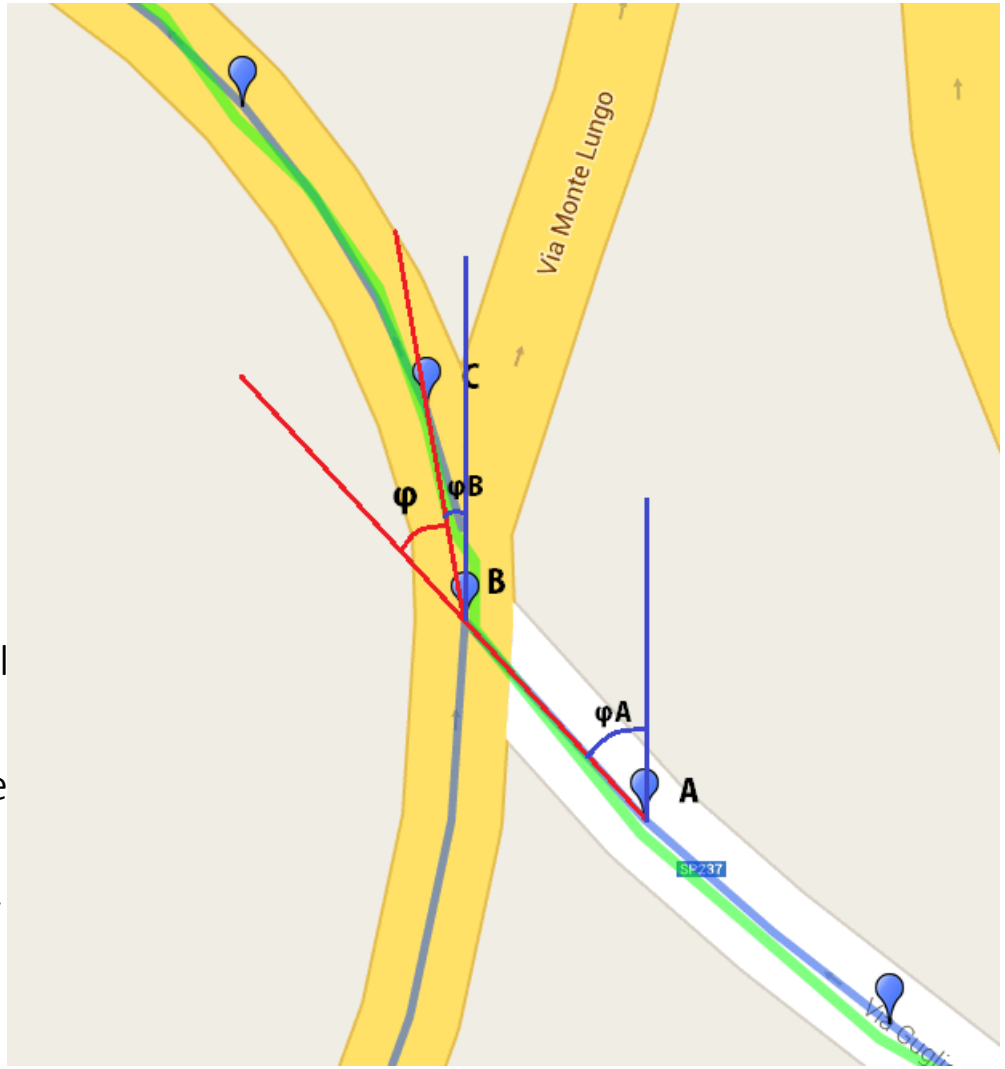
Digital Image Processing And Surface and Volumetric Data Rendering and Visualization

Dot. Andrea F. Bocchese

Dot. Sergio Leoni

Computate heading

- A, B, C points
- ϕA = heading from A to B, pointing direction of image in point A
- ϕB = heading from B to C, pointing direction of image in point B
- ϕA, ϕB are computed by Google geometry API
- ϕ = ϕB-ϕA = heading difference. Means there is a turn.
- If ϕ is high, creates new points with different heading value



Data Format

Dot. Andrea F. Bocchese

Dot. Sergio Leoni

Job = {JobId: id, Points: {Point, Point, ...}}

Point =

{

l: latitude,

b: longitude,

h: heading,

dtp: distance to previous point,

dtn: distance to next point,

pano: panorama ID,

t: turn (left or right),

};

Image Retrive Google Street View Image API **(Server Side)**:

https://maps.googleapis.com/maps/api/streetview?size=640x640&location=46.414382,10.013988&heading=151.78&pitch=0&key=879865746352445&fov=90

Digital Image Processing And Surface and Volumetric Data Rendering and Visualization

Dot. Andrea F. Bocchese

Dot. Sergio Leoni

YASVVM

The Server-Side Application

Server

Digital Image Processing And Surface and Volumetric Data Rendering and Visualization

Dot. Andrea F. Bocchese

Dot. Sergio Leoni

This Web application is meant to be as a single user application.
There's no session and all the videos created are public.
This choice is only due to the development easiness.

The server is composed by two components:

- Some PHP scripts that provide the API and implement the image download phase.
  The available APIs permit to:
  - Create new Job (sent the data point to the server).
  - Start video elaboration (download and start the video creator).
  - Retrieve the list of the jobs created.
  - Retrieve the percentage of completeness of the jobs.

- An application written in CPP (video creator).
  The main purposes of this application is:
  - Create the video.
  - Apply extra information (turn sign)
  - Apply the interpolation.

Video Creator

Digital Image Processing And Surface and Volumetric Data Rendering and Visualization

Dot. Andrea F. Bocchese

Dot. Sergio Leoni

As we said before, the video creator is an application written in CPP that use the OpenCV library.

The application called yasvvm accepts the following parameters[1]:

1. Directory of the frames
2. Filename of the output file
3. Frame number[2]
4. Working mode[3]

The yasvvm application during the phase of video creation does also other things:

- Add an arrow in the direction of the turn.
- Try to interpolate the frames in order to create smoother videos.

---

1. There isn't a CLI parser so the order must be strictly respect
2. If the working mode is set to 'l' this number is the sub frame added between the original frames; if the working mode is set to m this number is the fps of the videos.
3. 'l' for linear interpolation 'm' for motion flow interpolation.

Frame Doubling

Dot. Andrea F. Bocchese

Dot. Sergio Leoni

The street view images are shot several meters from each others.

Due to this if we only use this images like frame the resulting video is not smooth enough.

To try to make the video more fluid we used different techniques:

- Doubling by Zooming
- Linear Interpolation
- Motion Interpolation

The Zooming Technique

Digital Image Processing And Surface and Volumetric Data Rendering and Visualization

Dot. Andrea F. Bocchese

Dot. Sergio Leoni

Description:

A first naive approach used for multiplying frames, consists to retrieve from Google more pictures with different zoom levels to create intermediate frames.

This is implemented outside the video creator program, directly in PHP scripts.

Drawbacks:

When the different zoom levels are many (5-6) this technique tends to create a very smooth sequence interrupted by a jump. This is due to the camera FOV that is reduced in order to zoom the images.

The time to create the video and his size tend to explode, because the server must download several images.

Another drawback is that the objects that are stationary with the camera (other cars with same speed) tend to become near (bigger) and suddenly return to the normal distance (size).

Linear Interpolation

Digital Image Processing And Surface and Volumetric Data Rendering and Visualization

Dot. Andrea F. Bocchese

Dot. Sergio Leoni

Description:

This technique consists in the creation of additional frames blending the existing frames linearly. Every pixel of the additional (A) frame is the mean between the previous (P) and the consecutive (C) frame.

$$A_{x,y} = \frac{P_{x,y} + C_{x,y}}{2}$$

In order to generate more than one additional frame between two street view images we repeat this approach using a weighted mean.

We call $P^o{}_{x,y}$ the pixels of the first previous image directly downloaded (original) from street view.

We call $C^o{}_{x,y}$ the pixels of the first consecutive image directly downloaded (original) from street view.

We call N the number of sub frames between P^o and C^o.

$$A_{i\,x,y} = \left(1 - \frac{i}{N+1}\right) * P^o{}_{x,y} + \left(\frac{i}{N+1}\right) * C^o{}_{x,y}$$

An example with N = 3:

$$A_{1\,x,y} = 0.75 * P^o{}_{x,y} + 0.25 * C^o{}_{x,y}, \quad A_{2\,x,y} = 0.5 * P^o{}_{x,y} + 0.5 * C^o{}_{x,y}, \quad A_{3\,x,y} = 0.25 * P^o{}_{x,y} + 0.75 * C^o{}_{x,y}$$

Linear Interpolation 2

Dot. Andrea F. Bocchese

Dot. Sergio Leoni

Drawbacks:

If an object change its position respect the street view camera too quickly, the effect that appears at the end of this operation is a fading of the object from the start to the ending position.

This happens for example for the car that changes lane or the object at the side of the roadway.

Another drawback is that the size of the video tends to explode.

Motion Interpolation

Digital Image Processing And Surface and Volumetric Data Rendering and Visualization

Dot. Andrea F. Bocchese

Dot. Sergio Leoni

Description:

Motion Interpolation increases frame rate and can give the perception of smoother motion and more fluid animation. This effect is also know as “soap opera effect”

In our project Motion interpolation is achieved using an external open source application called **Butterflow**[1].

Drawbacks:

This technique is perfect when the frames are similar enough. But this isn’t our case.

The resulting effect is a weird deformation at the border of the image, where the scene changes most.

Another drawback is greater CPU time taken by the elaboration of the sub frames.

---

1. https://github.com/dthpham/butterflow

Summary

Dot. Andrea F. Bocchese

Dot. Sergio Leoni

The gap between the original street view pictures is too high, so every techniques implemented can't produce an acceptable smooth video.

- The zooming technique is slow and produce the worst video.
- The linear interpolation produce a fading effect of the object in the frame.
- The motion interpolation produce a weird effect mainly at the side of the images, where the scene most differ from each frame.

In conclusion no one of the techniques used could be considered the winner, the drawbacks are more than the pros, since the shots are so distant from each other "no technique" is the right choice.

This conclusion could be confirmed by the fact that no one of the existing "street view to video" systems try to interpolate the frames.

Video Sample

Digital Image Processing And Surface and Volumetric Data Rendering and Visualization

Dot. Andrea F. Bocchese

Dot. Sergio Leoni

YASVVM

UI demo

Video sample