

API de navegação no Google Street View e análise de imagens da paisagem urbana

Rodrigo Alves Lima

Wallace Faveron de Almeida

Orientador: Roberto Hirata Junior

Trabalho de Conclusão de Curso - Novembro de 2013

Motivação

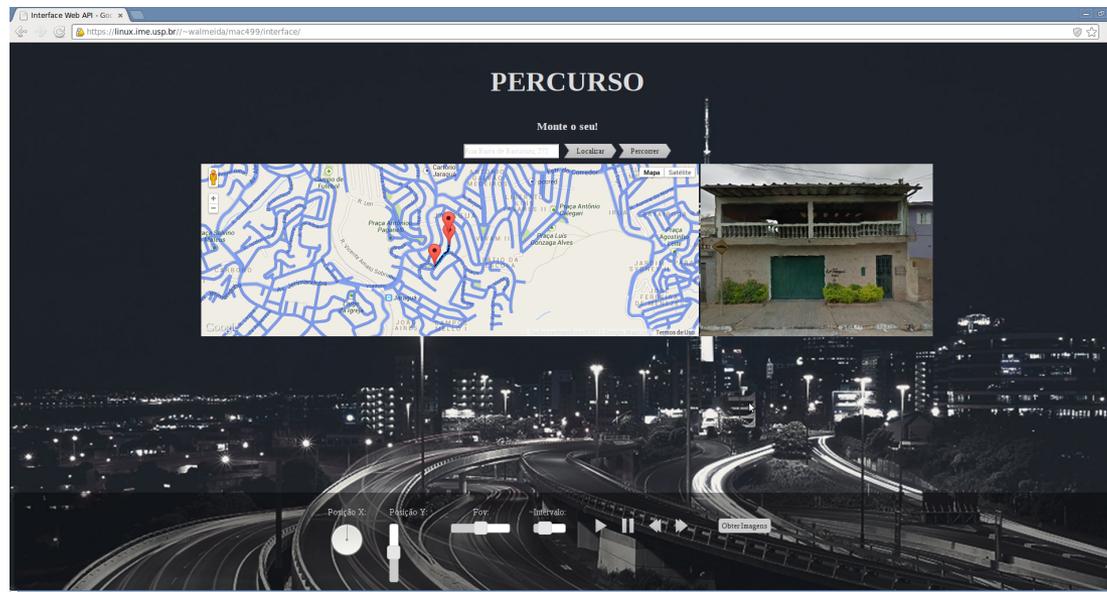
- Arquigrafia: sistema web de compartilhamento de imagens de arquitetura
- Teoria das Janelas Quebradas: influência da paisagem urbana no comportamento social
- Artigo publicado no *American Journal of Preventive Medicine*: Using Google Street View to audit neighborhood environments

APIs do Google

- Maps: visualização de mapa
- Directions: cálculo de rota
- Geocoding: conversão de endereço em coordenadas
- Street View Image: panorama estático do serviço Street View

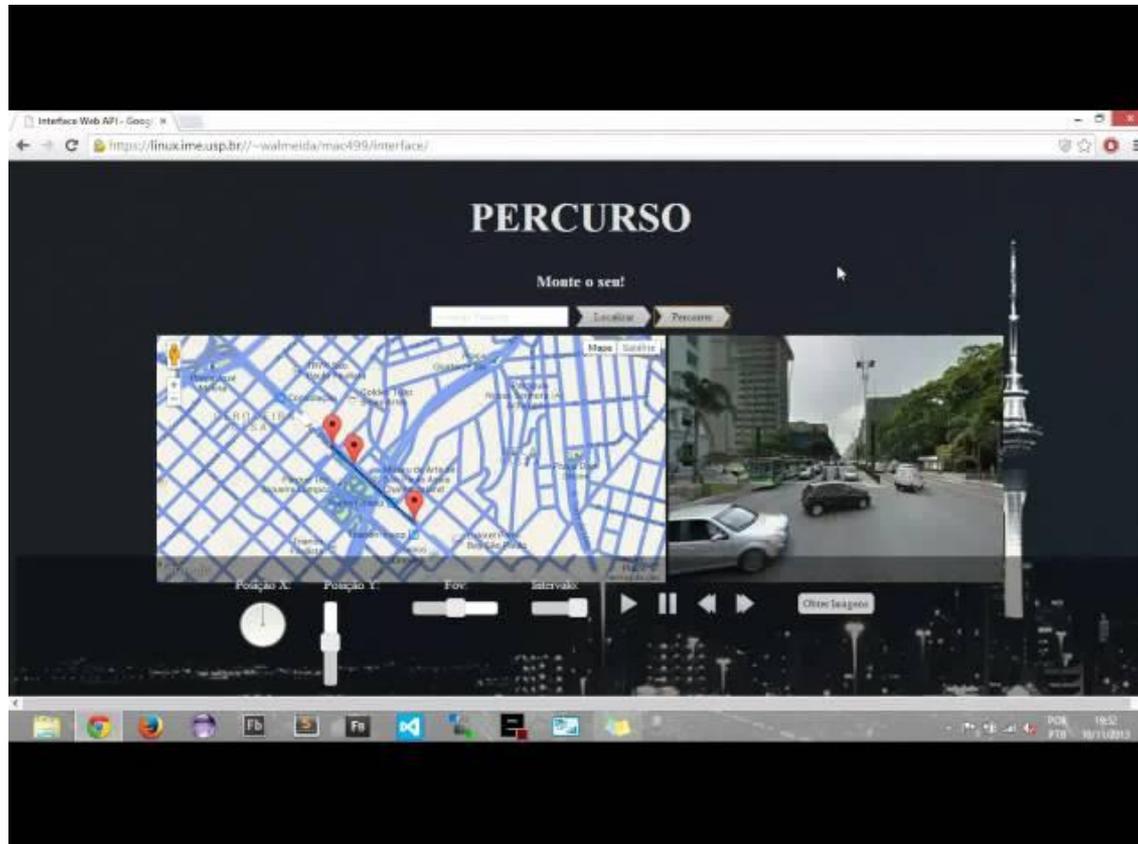
Interface web

- Navegação em uma rota
- HTML5, CSS, JavaScript
- Hyperlapse.js: time-lapse em uma rota



Interface web

<http://linux.ime.usp.br/~walmeida/mac499/interface>

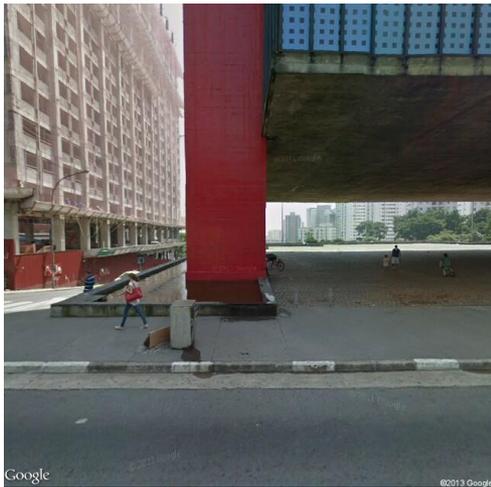


API de navegação

- Linguagem de programação Python
- Encapsula e estende funções das APIs do Google
- Associada a técnicas de processamento de imagens, permite a extração automática de informações da paisagem urbana

API de navegação

```
origin_point = Point.for_address(origin)
destination_point = Point.for_address(destination)
route = Route.between(origin_point, destination_point,
                      maximum_distance_between_points)
for image in route.google_street_view_images(heading, pitch):
    # image.url(API_KEY)
```



Imagens de rota na Avenida Paulista

Análise das cores dominantes

- Algoritmo de clusterização k-means agrupa os pixels pela proximidade de suas cores
- Os centros dos cluster são as cores dominantes



Cores dominantes de margem da Avenida Paulista, em São Paulo



Cores dominantes de margem da Avenida Doutor Arnaldo, em São Paulo



Cores dominantes de margem da Quinta Avenida, em Nova Iorque

Análise das cores dominantes

Exemplo de uso: obtenção das cores dominantes de imagens em uma rota entre dois endereços

```
origin_point = Point.for_address(origin)
destination_point = Point.for_address(destination)
route = Route.between(origin_point, destination_point,
                      maximum_distance_between_points)
for image in route.google_street_view_images(heading, pitch):
    # colors = image.dominant_colors(API_KEY, num_of_colors)
```

Detecção de objetos

- Implementação do algoritmo SURF da biblioteca OpenCV detecta objeto de interesse em imagens de uma rota



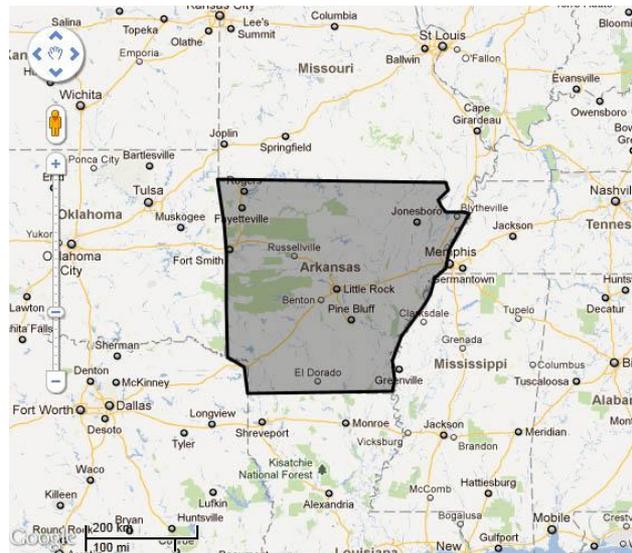
Detecção de objetos

Exemplo de uso: detecção de pontos característicos de um objeto em uma rota entre dois endereços

```
origin_point = Point.for_address(origin)
destination_point = Point.for_address(destination)
route = Route.between(origin_point, destination_point,
                      maximum_distance_between_points)
for image in route.google_street_view_images(heading, pitch):
    # image_surf(API_KEY, object_filename)
```

Continuidade...

- Detectar objetos por um classificador treinado com imagens do usuário
- Explorar todas as vias dentro de um polígono



Fonte: http://www.imapbuilder.com/images/gmap/polygon_area.jpg

Referências

Arquigrafia. Disponível em: <<http://www.arquigrafia.org.br>>. Acesso em: 1 novembro 2013.

Hyperlapse. Disponível em: <<http://hyperlapse.tllabs.io>>. Acesso em: 1 novembro 2013.

OpenCV. Disponível em: <<http://www.opencv.org>>. Acesso em: 1 novembro 2013.

Google Maps API. Disponível em: <<https://developers.google.com/maps/documentation>>. Acesso em: 1 novembro 2013.

Using python and k-means to find the dominant colors in images. Disponível em: <<http://charlesleifer.com/blog/using-python-and-k-means-to-find-the-dominant-colors-in-images>>. Acesso em: 1 novembro 2013.

James Q. Wilson, George L. Kelling. BROKEN WINDOWS: The police and neighborhood safety. Disponível em: <http://www.manhattan-institute.org/pdf/_atlantic_monthly-broken_windows.pdf>. Acesso em: 1 novembro 2013.

RUNDLE, Andrew G. et al. Using Google Street View to Audit Neighborhood Environments. *American Journal of Preventive Medicine*, janeiro 2011.