Corisco: Robust edgel-based orientation estimation for generic camera models

Nicolau L. Werneck

http://nic.hpavc.net

Supervisor: Profa. Dra. Anna Helena Reali Costa
Intelligent Techniques Laboratory, LTI — PCS — Poli

Universidade de São Paulo

Augsburg, Deutschland — 8/10/2015

Meta-introduction

Nicolau Werneck, Sc.D.

- E.Eng. graduated from UFMG, Unicamp and USP.
- Specialized in signal processing and pattern recognition, especially in computer vision and parameter estimation problems.
- Ex-Google and ex-Geekie.
- Wants to help robots help us.

Presentation adapted from my doctorate defense. Results were published in Werneck and Costa [2013].

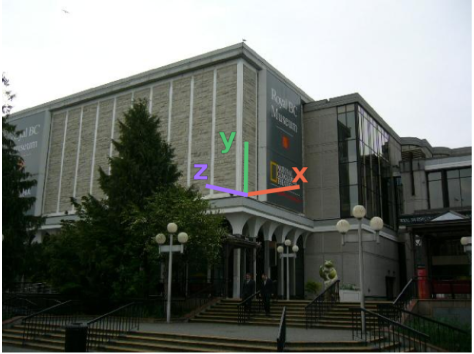
Presentation Schedule

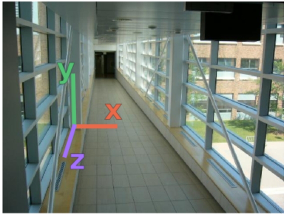
1. Problem introduction
2. Method proposal
3. Experiments and conclusion

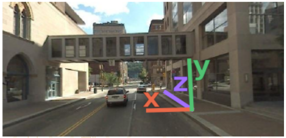
Anthropic Environments

An *anthropic environment* is composed by straight lines, parallel to the directions of a *natural reference frame*.

The orientation we find is a three-dimensional rotation between the natural and the camera reference frame.

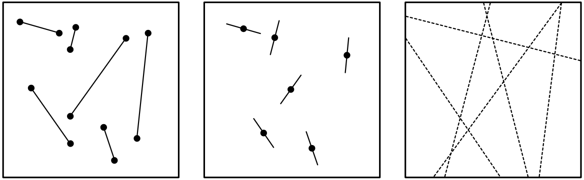






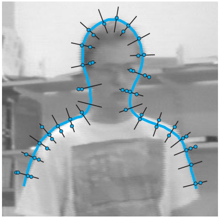
Edgels and straight lines

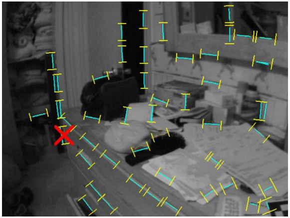
Edgels are points samples over curves or (straight) lines.



Segmentos Edgels Retas

Edgels have many applications.





Proposal

We propose a monocular vision method, denominated *Corisco*, that can estimate the orientation of a camera relative to an anthropic environment.

Evolution of existing edgel based methods [Coughlan and Yuille, 2003].

Aplication examples:

- Guiding a mobile robot.
- Initial estimates for multi-view reconstruction.
- Object orientation estimation.

Proposal

We propose a monocular vision method, denominated *Corisco*, that can estimate the orientation of a camera relative to an anthropic environment.

Evolution of existing edgel based methods [Coughlan and Yuille, 2003].

Aplication examples:

- Guiding a mobile robot.
- Initial estimates for multi-view reconstruction.
- Object orientation estimation.

Proposal

We propose a monocular vision method, denominated *Corisco*, that can estimate the orientation of a camera relative to an anthropic environment.

Evolution of existing edgel based methods [Coughlan and Yuille, 2003].

Aplication examples:

- Guiding a mobile robot.
- Initial estimates for multi-view reconstruction.
- Object orientation estimation.

Contributions

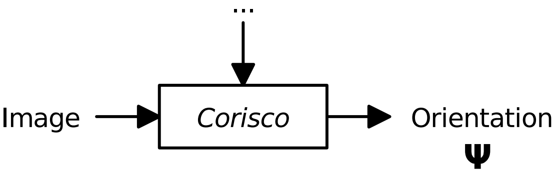
Corisco has the following peculiarities:

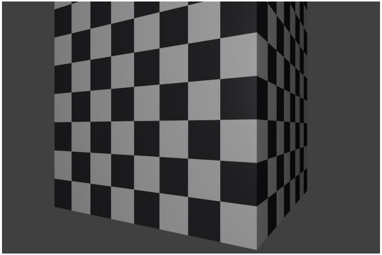
- Supports any possible camera model.
- Compromise between speed and precision.
- Dismisses the use of very costly operations (`sin`, `atan`, `exp`, `log`). Uses the function $x^{-1/2}$.

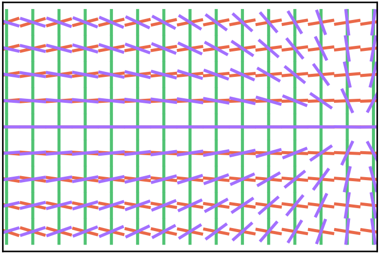
There are also no assumptions about the solution.

High-level view

Inputs: Image, intrinsic and control parameters.
Output: Orientation $\boldsymbol{\Psi}$ (three-dimensional rotation).



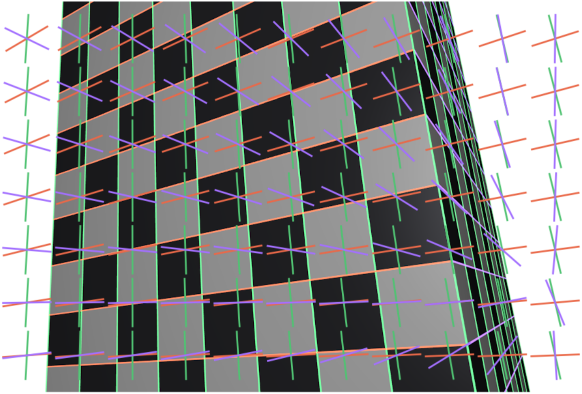




Geometry

The direction $\mathbf{v}$ of a line over the point $\mathbf{p}$ depends on $\boldsymbol{\Psi}$.

$$(\boldsymbol{\Psi}, \mathbf{p}) \rightarrow \mathbf{v}$$



http://youtu.be/PbUyvdnBIzs

Objective

Our question

What could be the best way to estimate the orientation of a camera in real time in an anthropic environment based on a single distorted image?

Objective

Our question

What could be the best way to estimate the orientation of a camera in real time in an anthropic environment based on a single distorted image?

Edgels *versus* lines

Orientation can be found from lines (Caprile and Torre [1990], Cipolla et al. [1999], Rother [2002]).

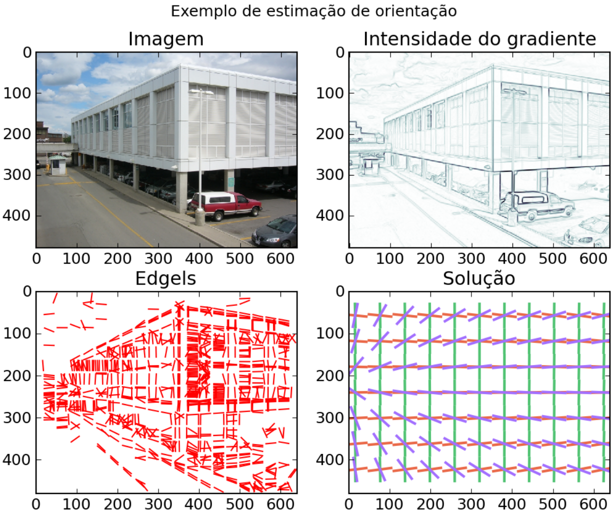
- More intuitive.
- Attain a good precision.

On the other hand:

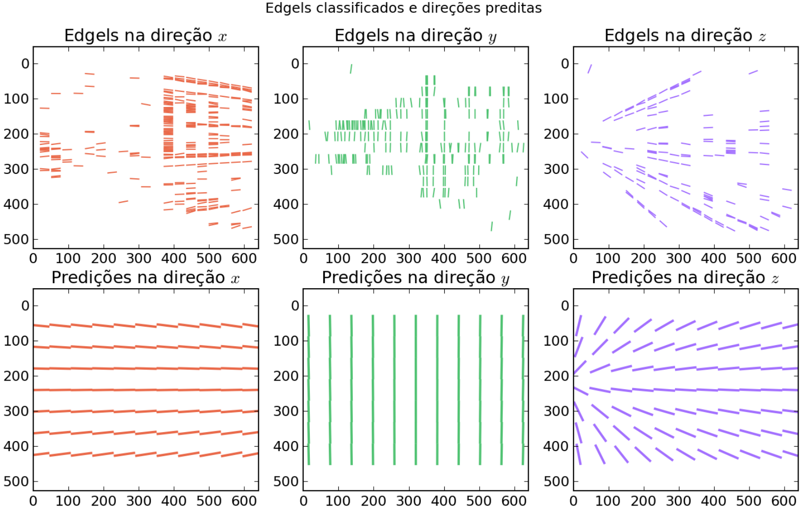
- Restricted to perspective projection.
- Rely on complicated line extraction.

Edgels allow us to handle distortions and are easier to extract and sub-sample.

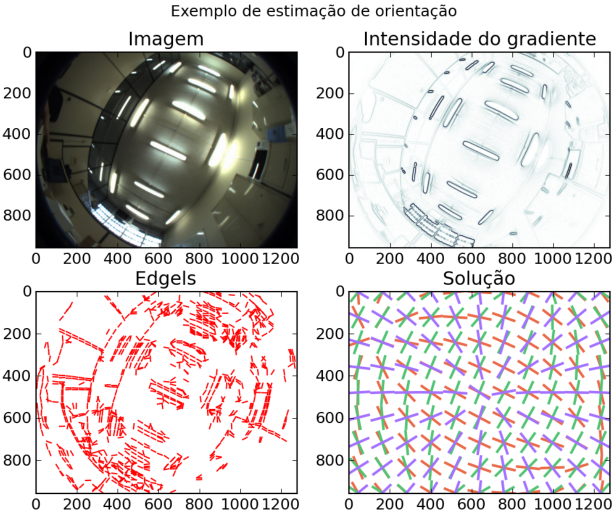
Corisco in detail



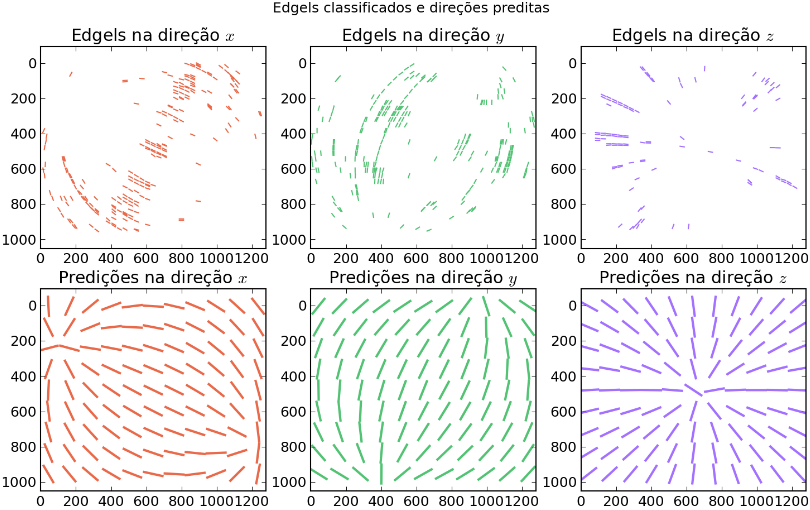
Corisco in detail



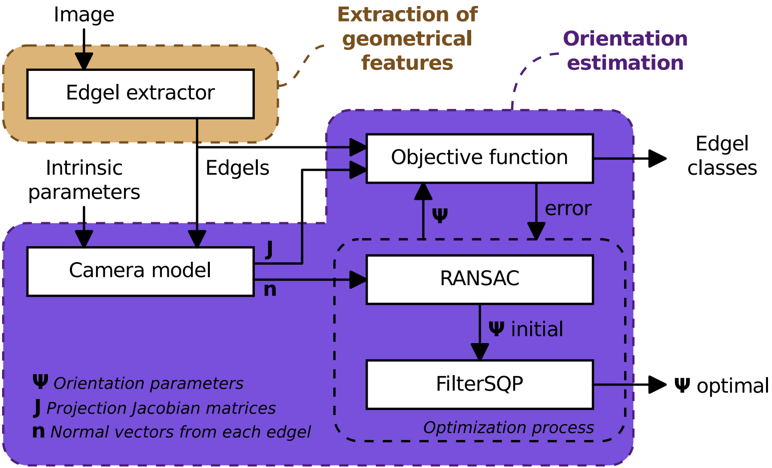
Corisco in detail



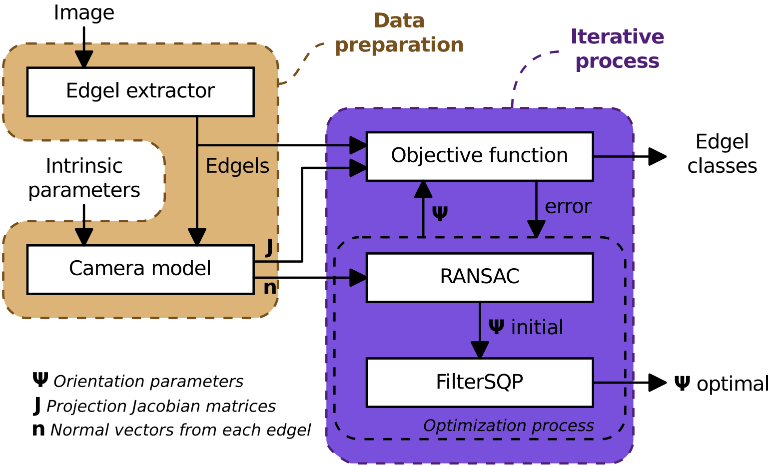
Corisco in detail



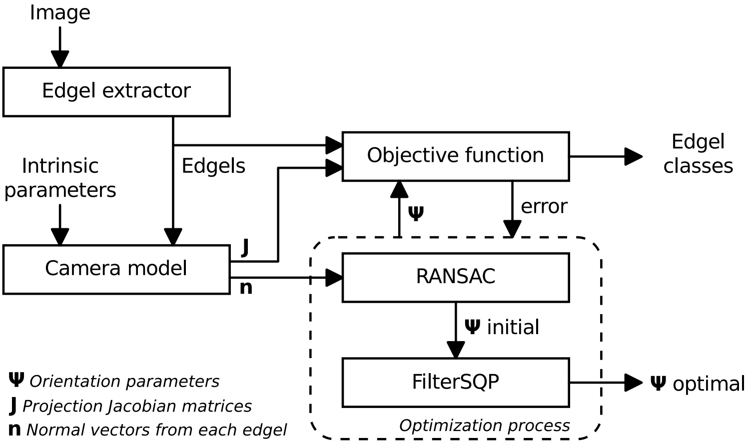
Block diagram



Block diagram

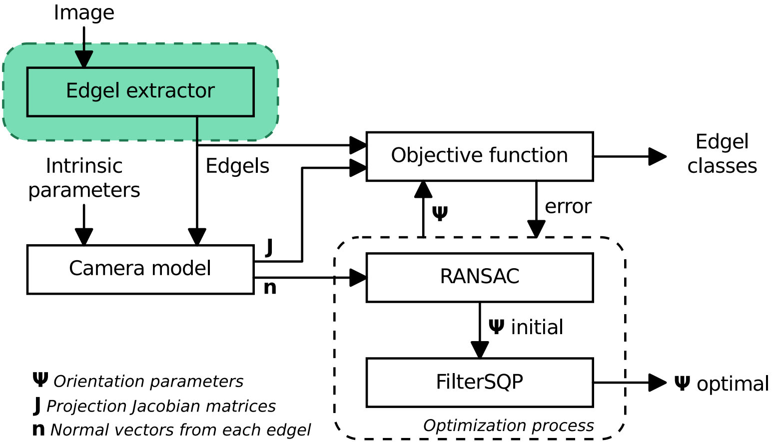


Block diagram

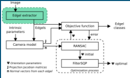


Ψ *Orientation parameters*
J *Projection Jacobian matrices*
n *Normal vectors from each edgel*

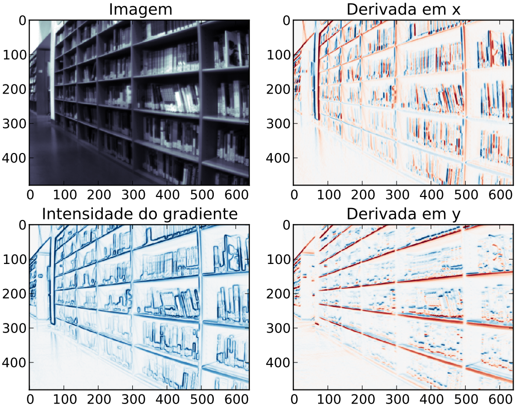
Edgel extraction



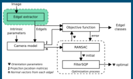
Edgel extraction



- Similar to the Canny border detection.
- Borders are local maxima in the gradient direction.

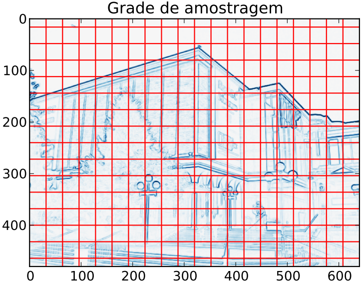


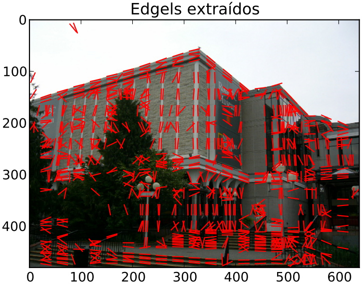
Edgel extraction



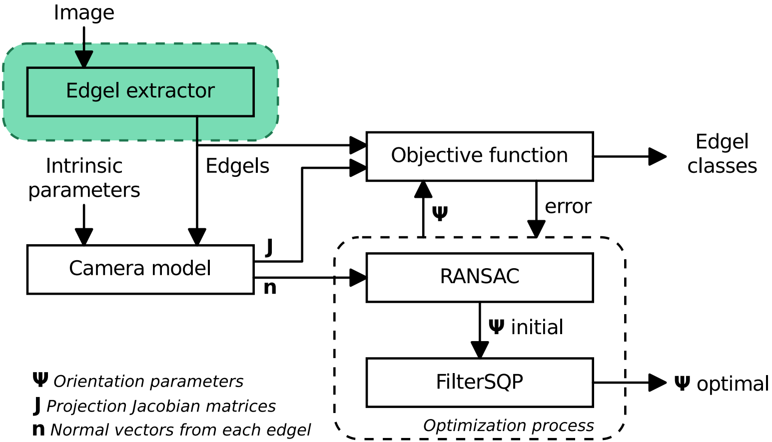
Sweep the image over a set of lines and columns that constitute a *grid mask*.

Each border found produces an edgel. Its direction should be approximately orthogonal to the line swept.

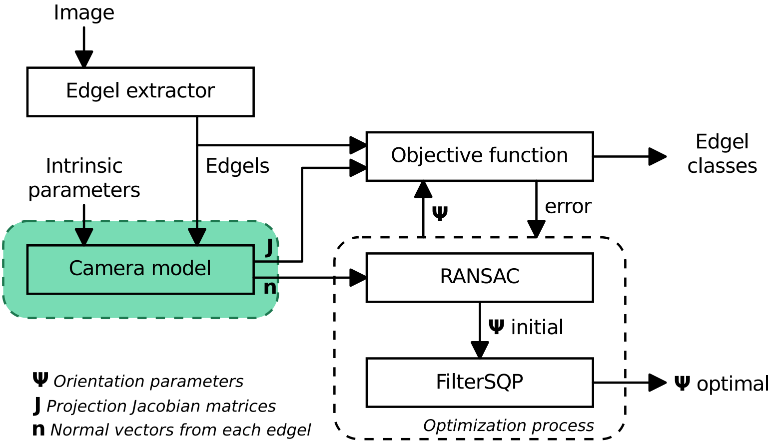


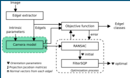


Camera models



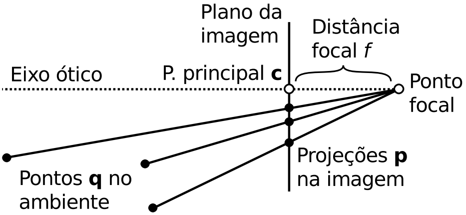
Camera models

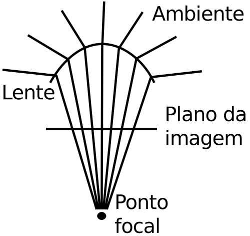




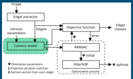
Bijective map between the image points and directions around the camera focal point.

$$\mathbf{q} \rightleftharpoons \mathbf{p}$$

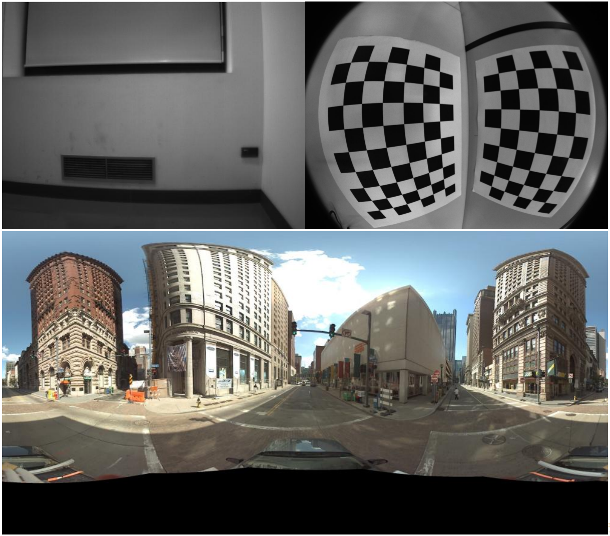




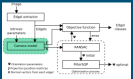
Camera model examples



Wide-angle, fisheye, equirectangular.



Equações de modelos de câmera



Perspective

$$\mathbf{p}^x = (\mathbf{q}^x/\mathbf{q}^z)f + \mathbf{c}^x$$
$$\mathbf{p}^y = (\mathbf{q}^y/\mathbf{q}^z)f + \mathbf{c}^y$$

Equiretangular (lat-lon)

$$\mathbf{p}^x = f\tan^{-1}(\mathbf{q}^z, \mathbf{q}^x)$$
$$\mathbf{p}^y = f\sin^{-1}(\mathbf{q}^y/|\mathbf{q}|)$$

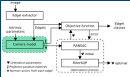
Harris (radial distortion)

$$g(x) = \frac{1}{\sqrt{1-2\kappa x^2}}$$
$$\mathbf{p}^x = \mathbf{p}'^x g(|\mathbf{p}'|) + \mathbf{c}^x$$
$$\mathbf{p}^y = \mathbf{p}'^y g(|\mathbf{p}'|) + \mathbf{c}^y$$

Polar Equidistant (fisheye)

$$\varphi = \cos^{-1}(\mathbf{q}^z/|\mathbf{q}|)$$
$$\mathbf{p}^x = \varphi\,\frac{\mathbf{q}^x}{\sqrt{\mathbf{q}^{x2}+\mathbf{q}^{y2}}}f + \mathbf{c}^x$$
$$\mathbf{p}^y = \varphi\,\frac{\mathbf{q}^y}{\sqrt{\mathbf{q}^{x2}+\mathbf{q}^{y2}}}f + \mathbf{c}^y$$

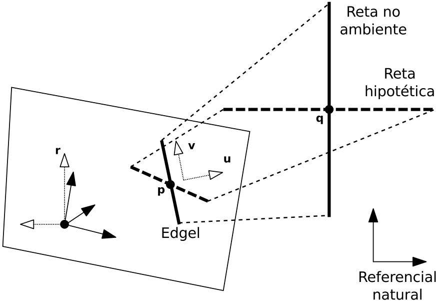
Edgel projection



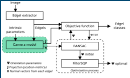
A point $\mathbf{q}_n$ from a line in the direction $\mathbf{r}_k$ is projected on $\mathbf{p}_n$, producuing an edgel com with direction

$$\mathbf{v}_{nk} \propto \mathbf{J}_n \mathbf{r}_k \qquad (\mathbf{v} \leftarrow (\boldsymbol{\Psi}, \mathbf{p}))$$

The projection Jacobian matrix $\mathbf{J}_n$ depends on $\mathbf{p}_n$. The direction orthogonal to $\mathbf{v}_n$ is denominated $\mathbf{u}_n$.



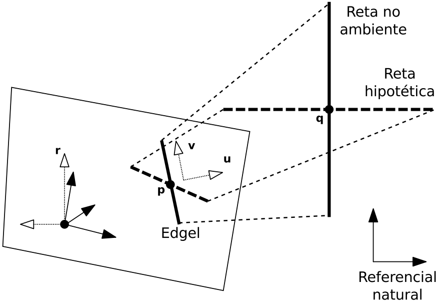
Edgel projection



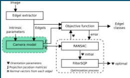
A point $\mathbf{q}_n$ from a line in the direction $\mathbf{r}_k$ is projected on $\mathbf{p}_n$, producuing an edgel com with direction

$$\mathbf{v}_{nk} \propto \mathbf{J}_n \mathbf{r}_k \qquad\qquad (\mathbf{v} \leftarrow (\boldsymbol{\Psi}, \mathbf{p}))$$

The projection Jacobian matrix $\mathbf{J}_n$ depends on $\mathbf{p}_n$. The direction orthogonal to $\mathbf{v}_n$ is denominated $\mathbf{u}_n$.



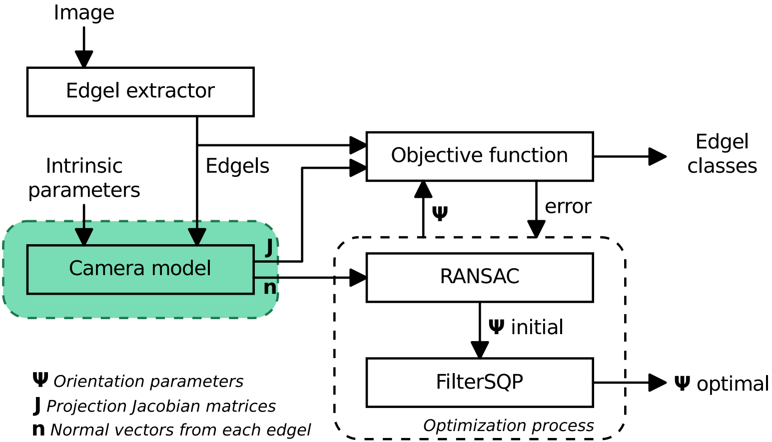
Edgel normal vector



Direction of a plane defined by the focal point and by an edgel, or a corresponding environment line.

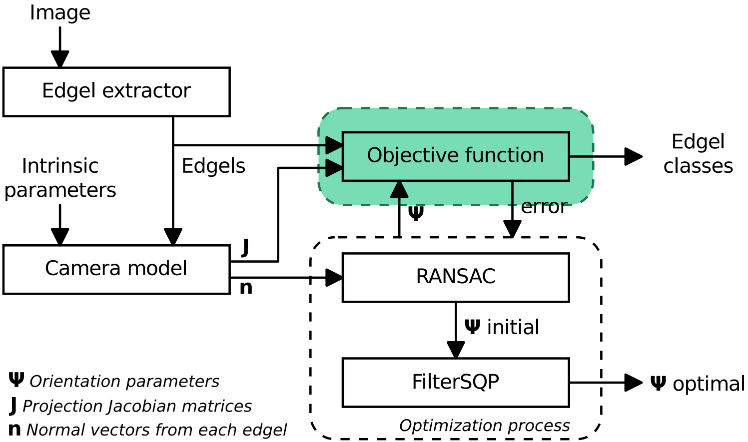
$$\mathbf{n}_k = \mathbf{u}_k^x \mathbf{J}_k^x + \mathbf{u}_k^y \mathbf{J}_k^y$$

Objective function

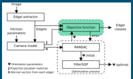


Ψ *Orientation parameters*
J *Projection Jacobian matrices*
n *Normal vectors from each edgel*

Objective function



Objective function



The objective function is the heart of the method.

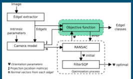
- Does not depend on the extraction technique.
- Determines the result.
- Conducts the choice of the optimization algorithm.

The expression has the shape of a summation of errors obtained from each observation.

$$y_n = x_n - \hat{x}_n(\mathbf{\Psi})$$

$$F(\mathbf{\Psi}) = \sum_n (y_n)^2$$

Rotation matrix



An edgel direction prediction starts by finding the directions of the natural frame $\mathbf{r}_k$ as a function of $\boldsymbol{\Psi}$.

Corisco works with quaternions.

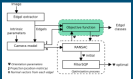
$$\boldsymbol{\Psi} = (\Psi^a, \Psi^b, \Psi^c, \Psi^d) \qquad |\boldsymbol{\Psi}| = 1$$

For a general orientation $\boldsymbol{\Psi}$ we have

$$\mathbf{R}(\boldsymbol{\Psi}) = \left[\ \mathbf{r}_x\ \mathbf{r}_y\ \mathbf{r}_z\ \right]^T =$$

$$\begin{bmatrix} {\Psi^a}^2 + {\Psi^b}^2 - {\Psi^c}^2 - {\Psi^d}^2 & 2\Psi^b\Psi^c + 2\Psi^a\Psi^d & 2\Psi^b\Psi^d - 2\Psi^a\Psi^c \\ 2\Psi^b\Psi^c - 2\Psi^a\Psi^d & {\Psi^a}^2 - {\Psi^b}^2 + {\Psi^c}^2 - {\Psi^d}^2 & 2\Psi^c\Psi^d + 2\Psi^a\Psi^b \\ 2\Psi^b\Psi^d + 2\Psi^a\Psi^c & 2\Psi^c\Psi^d - 2\Psi^a\Psi^b & {\Psi^a}^2 - {\Psi^b}^2 - {\Psi^c}^2 + {\Psi^d}^2 \end{bmatrix}$$

Edgel residue



Given the $\mathbf{r}_k$, we calculate the predicted $\mathbf{v}_{nk}$ using $\mathbf{J}_k$.

$$\boldsymbol{\Psi} \to \mathbf{r}_k \to \mathbf{v}_{nk}$$

$$\mathbf{v}_{nk} \propto \mathbf{J}_n \mathbf{r}_k$$

The residue compares:

- $\mathbf{v}_n$ (ou $\mathbf{u}_n$) taken from the imagem.
- $\mathbf{v}_{nk}$ predicted from $\boldsymbol{\Psi}$ e $\mathbf{p}_n$.

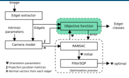
Previous methods:

$$\angle \mathbf{v} = \arctan(\mathbf{v}^x, \mathbf{v}^y)$$

$$\angle \mathbf{v}_n - \angle \mathbf{v}_{nk}$$

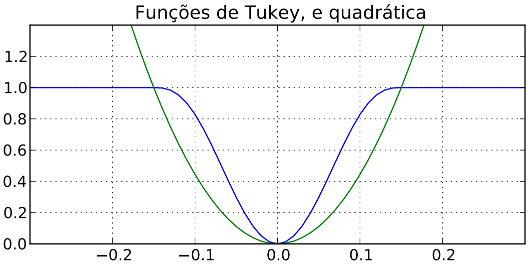
Corisco:

$$\mathbf{u}_n \mathbf{v}_{nk}$$

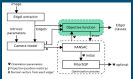


Comparison with existing edgel based techiques:

	MAP	EM	M-estimation
Models	Probabilistic	Probabilistic	$\rho(x)$
Classes	4	4	3
Classification	$\times$	Iterative	Direct



Objective function expression



MAP estimation (Coughlan and Yuille [2003]):

$$F(\boldsymbol{\Psi}) = -\sum_n \log\left(\sum_k p(c_n = k)p(\angle \mathbf{v}_n | \boldsymbol{\Psi}, \mathbf{p}_n, c_n = k)\right)$$

EM estimation (Schindler and Dellaert [2004]):

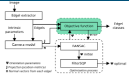
$$F(\boldsymbol{\Psi}) = -\sum_n \sum_k p(c_n = k) \log\left(p(\angle \mathbf{v}_n | \boldsymbol{\Psi}, \mathbf{p}_n, c_n = k)\right)$$

$$F(\boldsymbol{\Psi}) = \sum_n \sum_k p(c_n = k)\left(\angle \mathbf{v}_n - \angle \mathbf{v}_{nk}\right)^2$$

Corisco:

$$F(\boldsymbol{\Psi}) = \sum_n \min_k \rho\left(\mathbf{u}_n \mathbf{v}_{nk}\right)$$

Objective function expression



MAP estimation (Coughlan and Yuille [2003]):

$$F(\boldsymbol{\Psi}) = -\sum_n \log\left(\sum_k p(c_n = k)\, p(\angle \mathbf{v}_n | \boldsymbol{\Psi}, \mathbf{p}_n, c_n = k)\right)$$

EM estimation (Schindler and Dellaert [2004]):

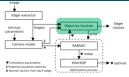
$$F(\boldsymbol{\Psi}) = -\sum_n \sum_k p(c_n = k) \log\left(p(\angle \mathbf{v}_n | \boldsymbol{\Psi}, \mathbf{p}_n, c_n = k)\right)$$

$$F(\boldsymbol{\Psi}) = \sum_n \sum_k p(c_n = k)\left(\angle \mathbf{v}_n - \angle \mathbf{v}_{nk}\right)^2$$

Corisco:

$$F(\boldsymbol{\Psi}) = \sum_n \min_k \rho(\mathbf{u}_n \mathbf{v}_{nk})$$

Objective function expression



MAP estimation (Coughlan and Yuille [2003]):

$$F(\boldsymbol{\Psi}) = -\sum_n \log\left(\sum_k p(c_n = k)p(\angle\mathbf{v}_n|\boldsymbol{\Psi}, \mathbf{p}_n, c_n = k)\right)$$

EM estimation (Schindler and Dellaert [2004]):

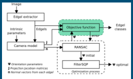
$$F(\boldsymbol{\Psi}) = -\sum_n\sum_k p(c_n = k)\log\left(p(\angle\mathbf{v}_n|\boldsymbol{\Psi}, \mathbf{p}_n, c_n = k)\right)$$

$$\boxed{F(\boldsymbol{\Psi}) = \sum_n\sum_k p(c_n = k)\left(\angle\mathbf{v}_n - \angle\mathbf{v}_{nk}\right)^2}$$

Corisco:

$$F(\boldsymbol{\Psi}) = \sum_n \min_k \rho(\mathbf{u}_n\mathbf{v}_{nk})$$

Objective function expression



MAP estimation (Coughlan and Yuille [2003]):

$$F(\mathbf{\Psi}) = -\sum_n \log\left(\sum_k p(c_n = k)\, p(\angle\mathbf{v}_n|\mathbf{\Psi}, \mathbf{p}_n, c_n = k)\right)$$

EM estimation (Schindler and Dellaert [2004]):

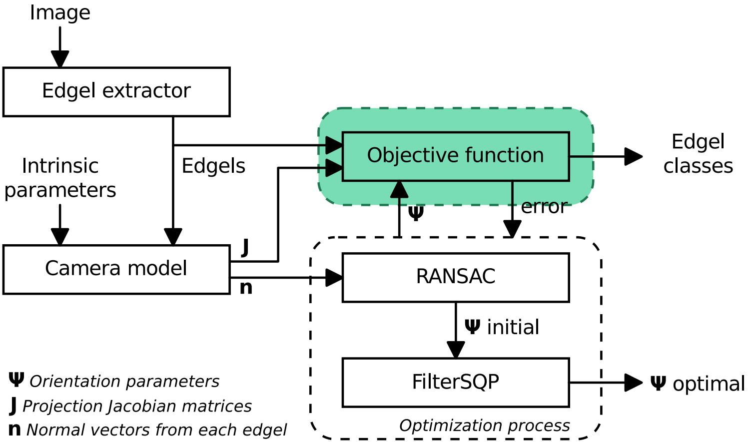
$$F(\mathbf{\Psi}) = -\sum_n \sum_k p(c_n = k) \log\left(p(\angle\mathbf{v}_n|\mathbf{\Psi}, \mathbf{p}_n, c_n = k)\right)$$

$$F(\mathbf{\Psi}) = \sum_n \sum_k p(c_n = k)\left(\angle\mathbf{v}_n - \angle\mathbf{v}_{nk}\right)^2$$

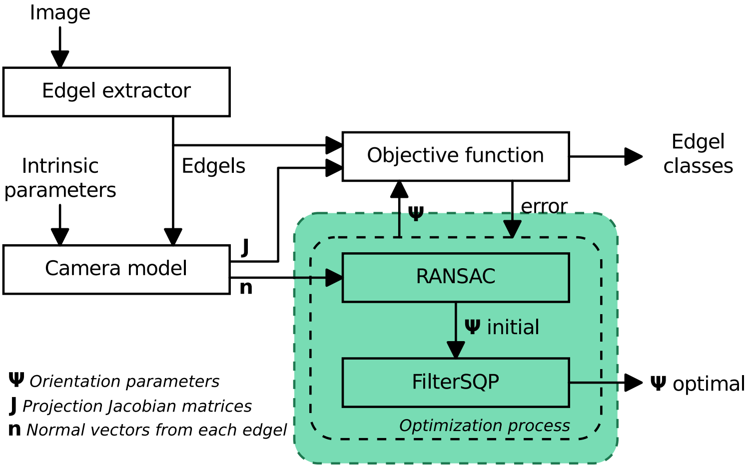
Corisco:

$$F(\mathbf{\Psi}) = \sum_n \min_k \rho(\mathbf{u}_n \mathbf{v}_{nk})$$

Optimization



Optimization



Optimization

First step: RANSAC, stochastic search guided by data. Inherently inefficient and imprecise.

$\rightarrow \boldsymbol{\Psi}$ inicial

Second step: FilterSQP, continuous optimization, more efficient and precise than RANSAC.

$\rightarrow \boldsymbol{\Psi}$ final

Optimization

First step: RANSAC, stochastic search guided by data. Inherently inefficient and imprecise.

$\rightarrow \boldsymbol{\Psi}$ inicial

Second step: FilterSQP, continuous optimization, more efficient and precise than RANSAC.

$\rightarrow \boldsymbol{\Psi}$ final

Optimization

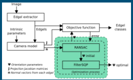
First step: RANSAC, stochastic search guided by data. Inherently inefficient and imprecise.

$\rightarrow \boldsymbol{\Psi}$ inicial

Second step: FilterSQP, continuous optimization, more efficient and precise than RANSAC.

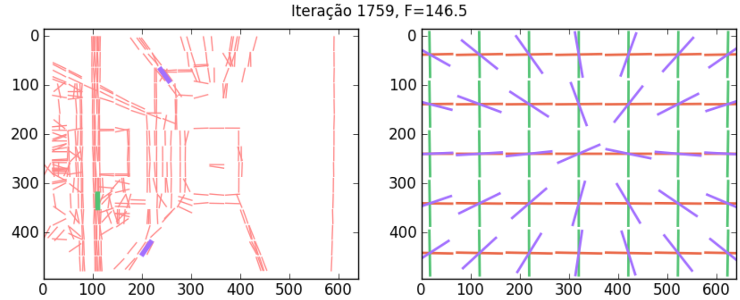
$\rightarrow \boldsymbol{\Psi}$ final

RANSAC



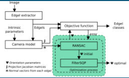
Observation triplets are randomly selected. From each one we calculate a hypothetical $\boldsymbol{\Psi}$, using $\mathbf{n}_k$.

The $\boldsymbol{\Psi}$ with the smallest $F(\boldsymbol{\Psi})$ is the initial estimate.



http://i.imgur.com/O9jP8tz.gifv

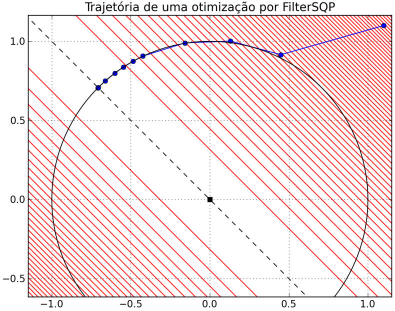
FilterSQP



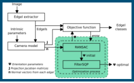
Minimize $F(\mathbf{\Psi})$ over the 4D and constrained to $|\mathbf{\Psi}| = 1$.

Non-linear program $\rightarrow$ **SQP:** Sequential Quadratic Programming

FilterSQP (Fletcher and Leyffer [2002]) dismisses penalty functions.



Derivative



Corisco calculates the $\boldsymbol{\Psi}$ derivatives by closed formulas.

$$\frac{\partial F}{\partial \Psi^a}(\boldsymbol{\Psi}) = \sum_{nk} K_{nk}\, \rho'(\mathbf{u}_n \mathbf{v}_{nk}) \left(u_n^x \frac{\partial \mathbf{v}_{nk}^x}{\partial \Psi^a} + u_n^y \frac{\partial \mathbf{v}_{nk}^y}{\partial \Psi^a} \right)$$

The derivatives of the directions $\mathbf{r}_k$ are trivial:

$$\begin{aligned} \frac{\partial \mathbf{r}^x}{\partial \Psi^a} &= 2(\Psi^a, \Psi^d, -\Psi^c) \\ \frac{\partial \mathbf{r}^x}{\partial \Psi^b} &= 2(\Psi^b, \Psi^c, \Psi^d) \\ &\cdots \end{aligned}$$

Experiments

We performed 3 experiments to assess the performance of *Corisco*.

Each experiment used a different image set, and method to obtain the reference orientations.

The observed error is the displacement in degrees of the "rotação residual" entre cada estimativa e referência.

Corisco was executed changing the settings:

- grid size C_g,
- number of RANSAC iterations C_r.

YorkUrbanDB

Images: 101 imagens from anthropic environments.

Model: Perspective.

Reference: Semi-automatic method based on lines.

Comparison: Methods studied by Denis et al. [2008].

Camera parameters, reference orientations and performance statistics provided by the authors.

YorkUrbanDB



YorkUrbanDB

Method	Time [s]	Error				
		Mean	σ	1/4	Median	3/4
EM Newton	27+?	4.00°	1.00°	1.15°	2.61°	4.10°
MAP Quasi-Newton	6+?	4.00°	1.00°	1.32°	2.39°	4.07°
EM Quasi-Newton	1+?	9.00°	1.00°	4.04°	6.21°	10.33°
J-linkage	1.13	8.23°	13.76°	1.14°	2.36°	4.44°
Corisco $C_r = 10^4$ $C_g = 1$	47.20	1.51°	3.26°	0.69°	1.09°	1.51°
Corisco $C_r = 10^4$ $C_g = 4$	16.68	1.71°	3.35°	0.72°	1.14°	1.64°
Corisco $C_r = 10^4$ $C_g = 32$	7.57	2.43°	4.03°	0.97°	1.54°	2.42°
Corisco $C_r = 10^3$ $C_g = 1$	8.12	1.70°	3.22°	0.70°	1.11°	1.77°
Corisco $C_r = 10^3$ $C_g = 4$	2.50	2.02°	3.86°	0.81°	1.24°	1.80°
Corisco $C_r = 10^3$ $C_g = 32$	0.99	2.44°	3.54°	1.00°	1.68°	2.64°
Corisco $C_r = 200$ $C_g = 1$	5.34	2.08°	3.38°	0.72°	1.22°	1.78°
Corisco $C_r = 200$ $C_g = 4$	1.89	3.27°	6.38°	0.85°	1.34°	2.35°
Corisco $C_r = 200$ $C_g = 32$	0.45	3.29°	4.99°	0.99°	1.72°	3.46°

YorkUrbanDB

Method	Time [s]	Error Mean	Error σ	Error 1/4	Error Median	Error 3/4
EM Newton	27+?	4.00°	1.00°	1.15°	2.61°	4.10°
MAP Quasi-Newton	6+?	4.00°	1.00°	1.32°	2.39°	4.07°
EM Quasi-Newton	1+?	9.00°	1.00°	4.04°	6.21°	10.33°
J-linkage	1.13	8.23°	13.76°	1.14°	2.36°	4.44°
Corisco $C_r = 10^4$ $C_g = 1$	47.20	1.51°	3.26°	0.69°	1.09°	1.51°
Corisco $C_r = 10^4$ $C_g = 4$	16.68	1.71°	3.35°	0.72°	1.14°	1.64°
Corisco $C_r = 10^4$ $C_g = 32$	7.57	2.43°	4.03°	0.97°	1.54°	2.42°
Corisco $C_r = 10^3$ $C_g = 1$	8.12	1.70°	3.22°	0.70°	1.11°	1.77°
Corisco $C_r = 10^3$ $C_g = 4$	2.50	2.02°	3.86°	0.81°	1.24°	1.80°
Corisco $C_r = 10^3$ $C_g = 32$	0.99	2.44°	3.54°	1.00°	1.68°	2.64°
Corisco $C_r = 200$ $C_g = 1$	5.34	2.08°	3.38°	0.72°	1.22°	1.78°
Corisco $C_r = 200$ $C_g = 4$	1.89	3.27°	6.38°	0.85°	1.34°	2.35°
Corisco $C_r = 200$ $C_g = 32$	0.45	3.29°	4.99°	0.99°	1.72°	3.46°

YorkUrbanDB

Method	Time [s]	Error Mean	Error σ	Error 1/4	Error Median	Error 3/4
EM Newton	27+?	4.00°	1.00°	1.15°	2.61°	4.10°
MAP Quasi-Newton	6+?	4.00°	1.00°	1.32°	2.39°	4.07°
EM Quasi-Newton	1+?	9.00°	1.00°	4.04°	6.21°	10.33°
J-linkage	1.13	8.23°	13.76°	1.14°	2.36°	4.44°
Corisco $C_r = 10^4$ $C_g = 1$	47.20	1.51°	3.26°	0.69°	1.09°	1.51°
Corisco $C_r = 10^4$ $C_g = 4$	16.68	1.71°	3.35°	0.72°	1.14°	1.64°
Corisco $C_r = 10^4$ $C_g = 32$	7.57	2.43°	4.03°	0.97°	1.54°	2.42°
Corisco $C_r = 10^3$ $C_g = 1$	8.12	1.70°	3.22°	0.70°	1.11°	1.77°
Corisco $C_r = 10^3$ $C_g = 4$	2.50	2.02°	3.86°	0.81°	1.24°	1.80°
Corisco $C_r = 10^3$ $C_g = 32$	0.99	2.44°	3.54°	1.00°	1.68°	2.64°
Corisco $C_r = 200$ $C_g = 1$	5.34	2.08°	3.38°	0.72°	1.22°	1.78°
Corisco $C_r = 200$ $C_g = 4$	1.89	3.27°	6.38°	0.85°	1.34°	2.35°
Corisco $C_r = 200$ $C_g = 32$	0.45	3.29°	4.99°	0.99°	1.72°	3.46°

YorkUrbanDB

Method	Time [s]	Error Mean	σ	1/4	Median	3/4
EM Newton	27+?	4.00°	1.00°	1.15°	2.61°	4.10°
MAP Quasi-Newton	6+?	4.00°	1.00°	1.32°	2.39°	4.07°
EM Quasi-Newton	1+?	9.00°	1.00°	4.04°	6.21°	10.33°
J-linkage	1.13	8.23°	13.76°	1.14°	2.36°	4.44°
Corisco $C_r = 10^4$ $C_g = 1$	47.20	1.51°	3.26°	0.69°	1.09°	1.51°
Corisco $C_r = 10^4$ $C_g = 4$	16.68	1.71°	3.35°	0.72°	1.14°	1.64°
Corisco $C_r = 10^4$ $C_g = 32$	7.57	2.43°	4.03°	0.97°	1.54°	2.42°
Corisco $C_r = 10^3$ $C_g = 1$	8.12	1.70°	3.22°	0.70°	1.11°	1.77°
Corisco $C_r = 10^3$ $C_g = 4$	2.50	2.02°	3.86°	0.81°	1.24°	1.80°
Corisco $C_r = 10^3$ $C_g = 32$	0.99	2.44°	3.54°	1.00°	1.68°	2.64°
Corisco $C_r = 200$ $C_g = 1$	5.34	2.08°	3.38°	0.72°	1.22°	1.78°
Corisco $C_r = 200$ $C_g = 4$	1.89	3.27°	6.38°	0.85°	1.34°	2.35°
Corisco $C_r = 200$ $C_g = 32$	0.45	3.29°	4.99°	0.99°	1.72°	3.46°

ApaSt

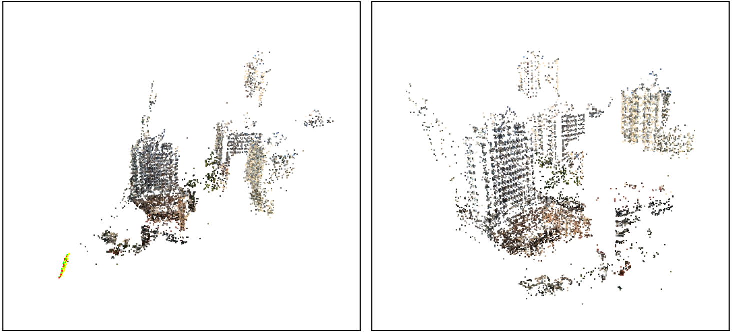
Images: 24+24 building images.

Model: Perspective with radial distortion (Harris).

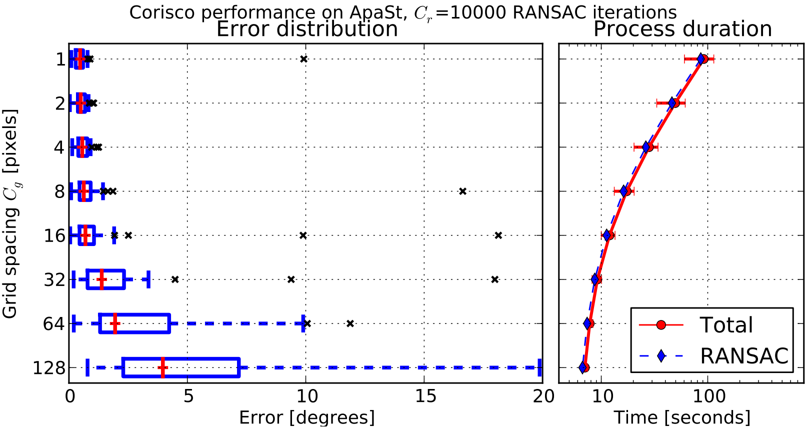
Reference: *Bundler*.

(Snavely et al. [2006]) foi used to obtain the reference orientations and intrinsic parameters.

Point-based multi-view method.

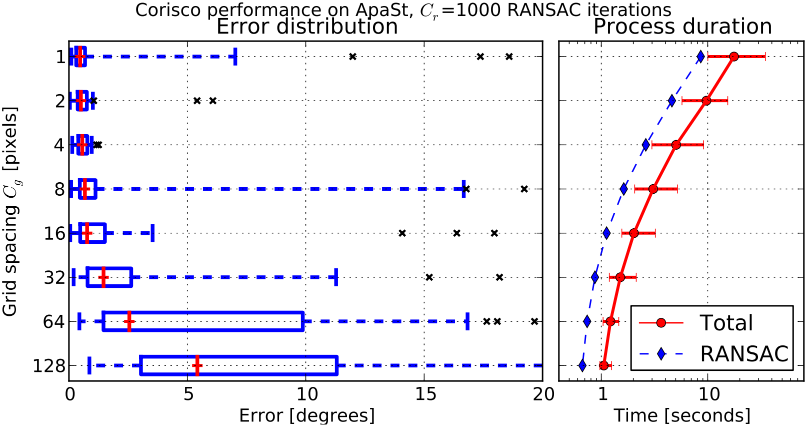


ApaSt



Corisco performance on ApaSt, C_r =10000 RANSAC iterations
Error distribution
Process duration
Grid spacing C_g [pixels]
1
2
4
8
16
32
64
128
0
5
10
15
20
Error [degrees]
10
100
Time [seconds]
Total
RANSAC

ApaSt



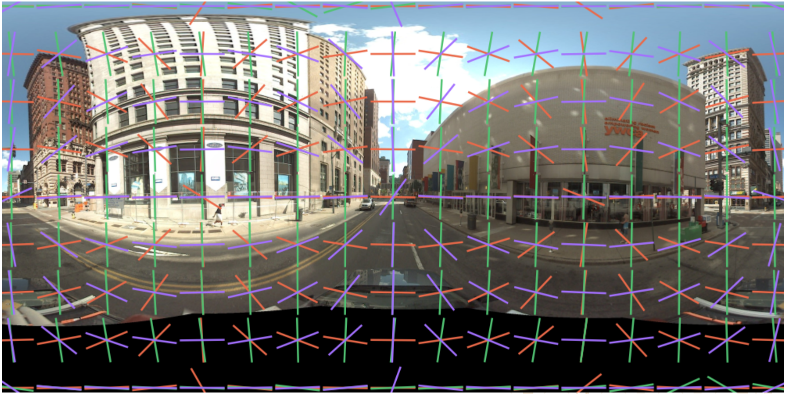
Corisco performance on ApaSt, C_r =1000 RANSAC iterations
Error distribution
Process duration
Grid spacing C_g [pixels]
1
2
4
8
16
32
64
128
0
5
10
15
20
Error [degrees]
1
10
Time [seconds]
Total
RANSAC

StreetView

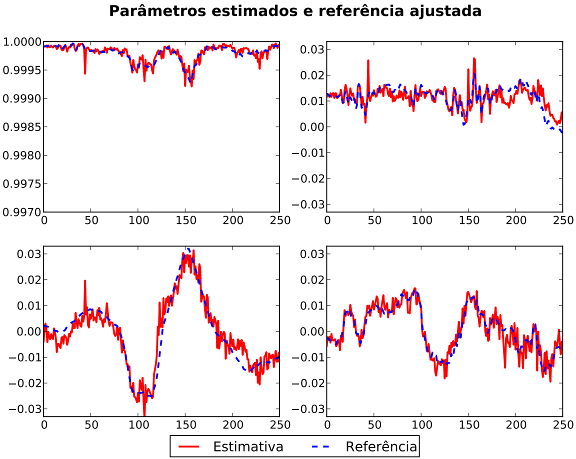
Images: 250 images from an urban environment.

Model: Equiretangular projection.

Reference: Physical sensors (IMU).



Erro $\simeq 1^\circ$, time $\simeq 17$s.



Conclusion

Corisco is a method with great potential for immediate application.

- Relatively simple implementation.
- Great robustness (distortions, RANSAC).
- Good performance.

We demonstrated the advantages of applying the grid mask and M-estimation to the problem.

We also demonstrated how to use FilterSQP in order to work with quaternions in a convenient way.

Future Work

- Automatic parameter control. Replace RANSAC.
- Apply to multi-view reconstruction, monocular SLAM, object tracking...
- Use directional filters to measure the angular error.
- Use the grid mask in other problems.
- Estimate orientation, camera parameters, and extract curves all in a single unified process. (MRF?)

Fim

Obrigado!

Referências Bibliográficas

B. Caprile and V. Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4(2):127–139, March 1990. ISSN 0920-5691. doi: 10.1007/BF00127813. URL http://www.springerlink.com/content/k75077108473tm15/.

R Cipolla, T Drummond, and D Robertson. Camera calibration from vanishing points in images of architectural scenes. In *British Machine Vision Conference*, volume 2, pages 382–391, Nottingham, England, 1999. BMVA.

James M. Coughlan and A. L. Yuille. Manhattan World: Orientation and outlier detection by Bayesian inference. *Neural Computation*, 15(5): 1063—-1088, March 2003. URL http://www.mitpressjournals.org/doi/abs/10.1162/089976603765202668.

Patrick Denis, James H Elder, and Francisco J Estrada. Efficient edge-based methods for estimating Manhattan frames in urban imagery. In *European Conference on Computer Vision*, pages 197–210, Marselha, França, 2008. Springer. doi: 10.1007/978-3-540-88688-4_15.

Roger Fletcher and Sven Leyffer. Nonlinear programming without a penalty function. *Mathematical Programming*, 91(2):239–269, January 2002. ISSN 0025-5610. doi: 10.1007/s101070100244. URL http://www.springerlink.com/content/qqj37x00y79ygdl8/.

Carsten Rother. A new approach to vanishing point detection in architectural environments. *Image and Vision Computing*, 20(9-10): 647–655, 2002. doi: 10.1016/S0262-8856(02)00054-9. URL http://dx.doi.org/10.1016/S0262-8856(02)00054-9.

G. Schindler and F. Dellaert. Atlanta World: An expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments. In *Conference on Computer Vision and Pattern Recognition*, pages 203–209, Washington, DC, USA, 2004. IEEE. ISBN 0-7695-2158-4. doi: 10.1109/CVPR.2004.1315033. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1315033.

Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo Tourism: Exploring photo collections in 3D. In *SIGGRAPH*, pages 835–846, Boston, MA, USA, 2006. ACM. URL http://dl.acm.org/citation.cfm?id=1141964.

Nicolau Leal Werneck and Anna Helena Reali Costa. Corisco: Robust edgel-based orientation estimation for generic camera models. *Image and Vision Computing*, 12(31):969—-981, 2013. URL http://nic.hpavc.net/almoxarifado/imavis2013-final.pdf.

Calibration test

Estimating focal distance with the same $F(\boldsymbol{\Psi})$.