Octave in Computer Vision

Etienne Grossmann

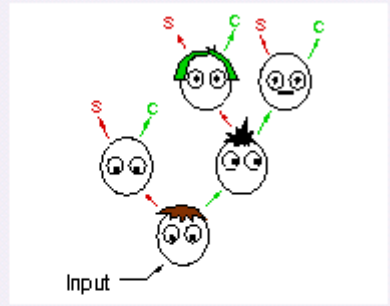
visiting at U. of Montreal

etienne@isr.ist.utl.pt

www.isr.ist.utl.pt/~etienne

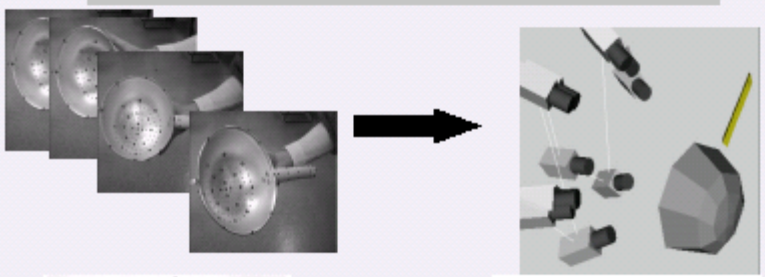
updated Octave Meeting

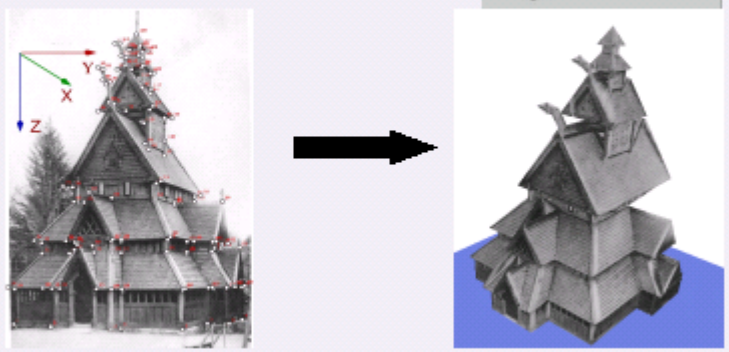
presentation, 2006 / 04 / 23



Decision trees

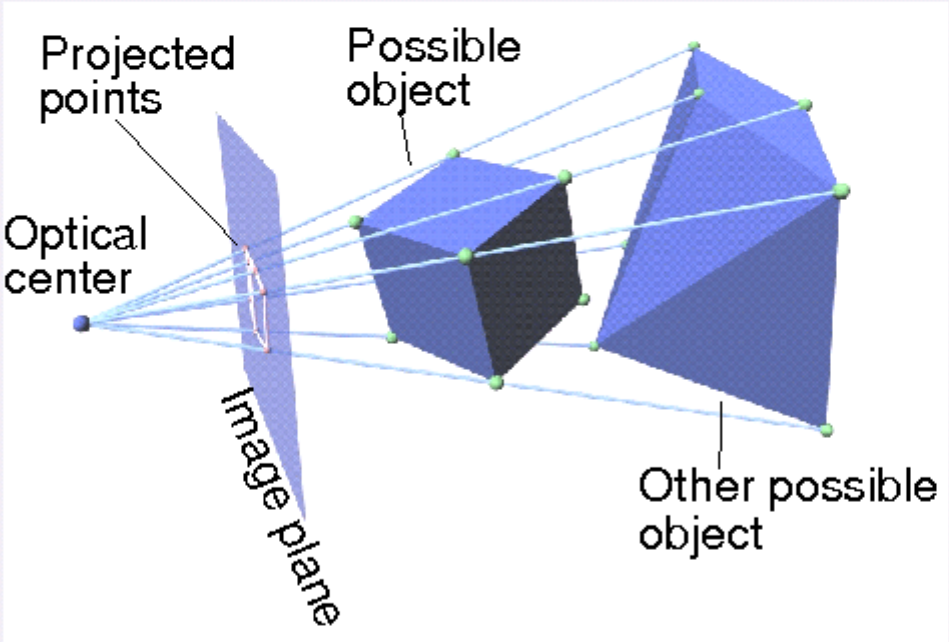
3D reconstruction





Typical usage

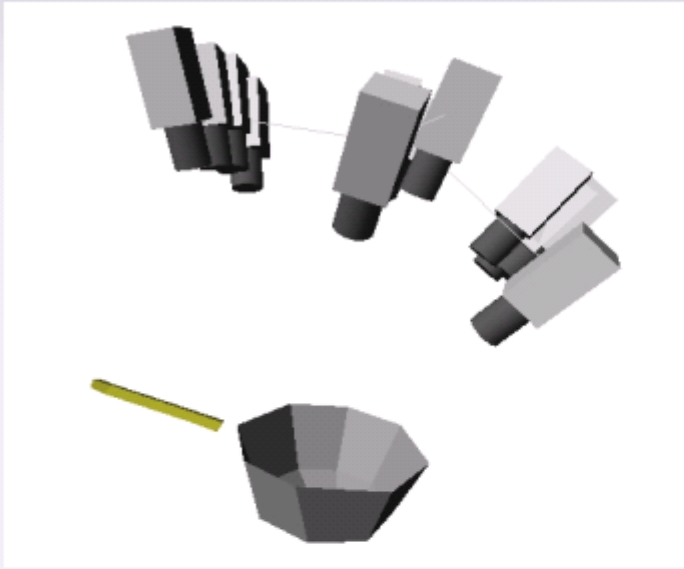
Perspective projection
linear operations



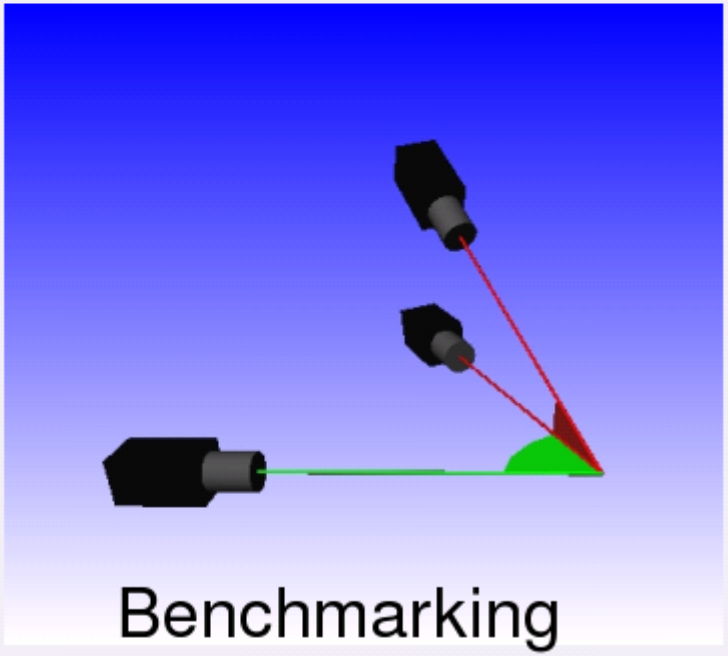
Paraperspective projection : SVD-based 3D reconstruction.

Typical usage

Nonlinear optimization,
sometimes lots of it,
sometimes constrained



Least-squares optimization

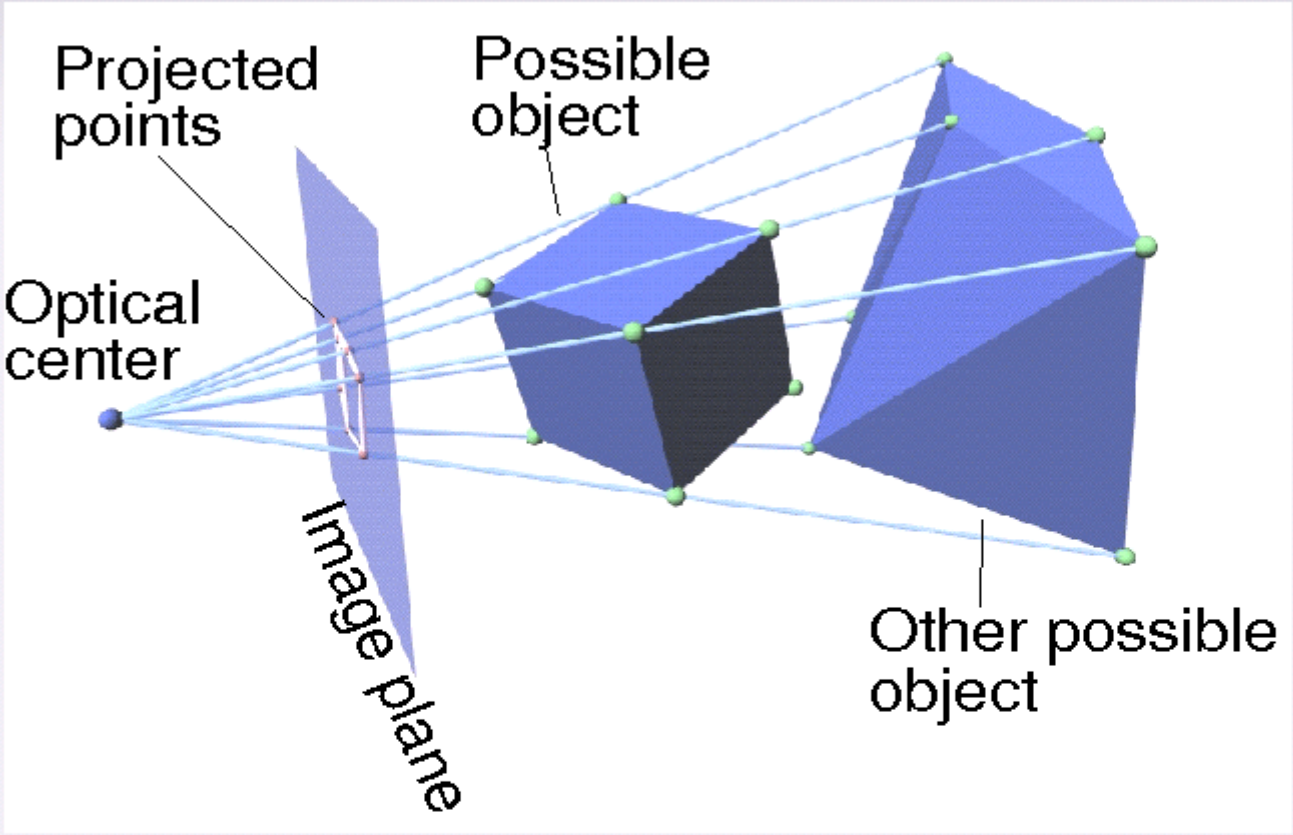


Benchmarking

Geometrically constrained scene

More in detail

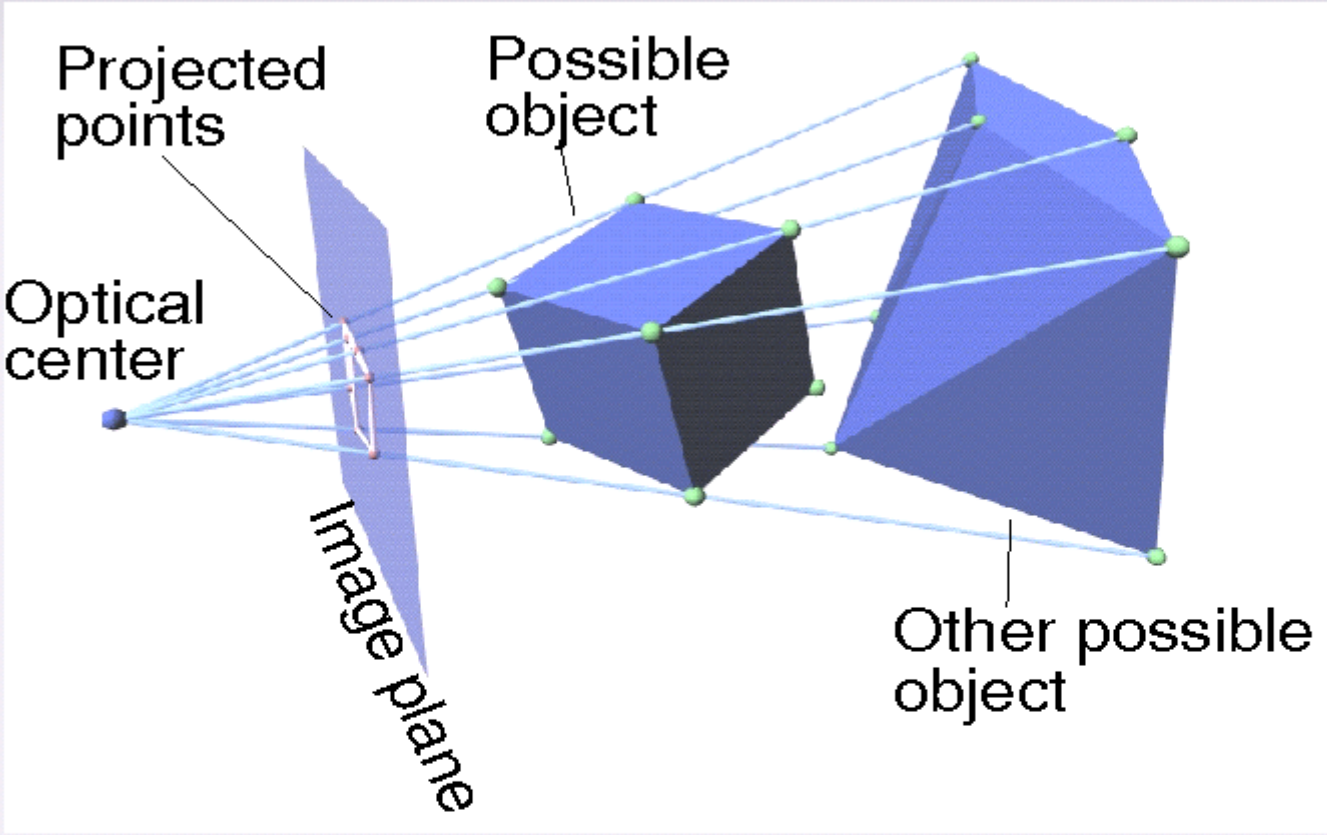
Perhaps the most basic operation in 3D vision is "projecting" a 3D point (on objects, at right) to an image plane.



Perspective projection is computed by 3x3 matrix multiplications and additions of 3x1 vectors. Octave/Matlab syntax is ideal for this.

More in detail

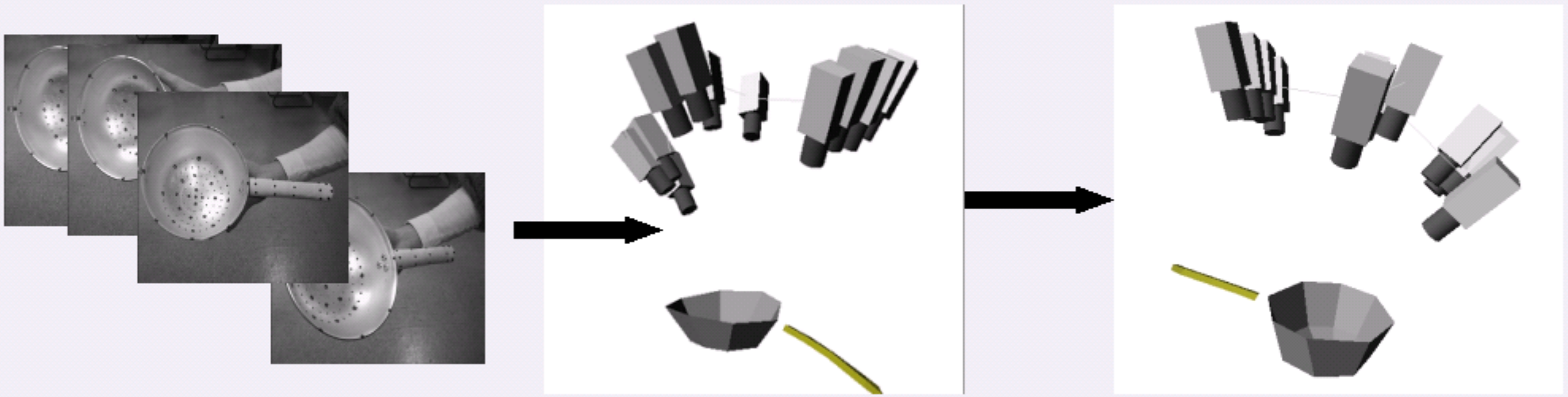
Technically, this image is a snapshot produced by the VRML browser FreeWRL and later annotated with XFig.



The VRML file was produced by Octave, using octave-forge's vrml_lines, vrml_faces and vrml_points functions.

More in detail

In 3D reconstruction, one estimates the position of 3D points (on the noodle drain), camra positions and some internal ("intrinsic") camera parameters.



Typical computation steps include

(A) identifying and tracking interest points in many views, left. Here, this was done "manually" with the help of a homemade GTK click-zoom-scroll tool.

(B) obtaining a first, crude reconstruction, middle. Here, the simple "paraperspective" camera model is used in my "reconst_para()" function [PK93,GSV00ICPR].

(C) A least squares reconstruction is obtained iteratively, middle, using Octave-Forge's Levenberg-Marquardt implementation.

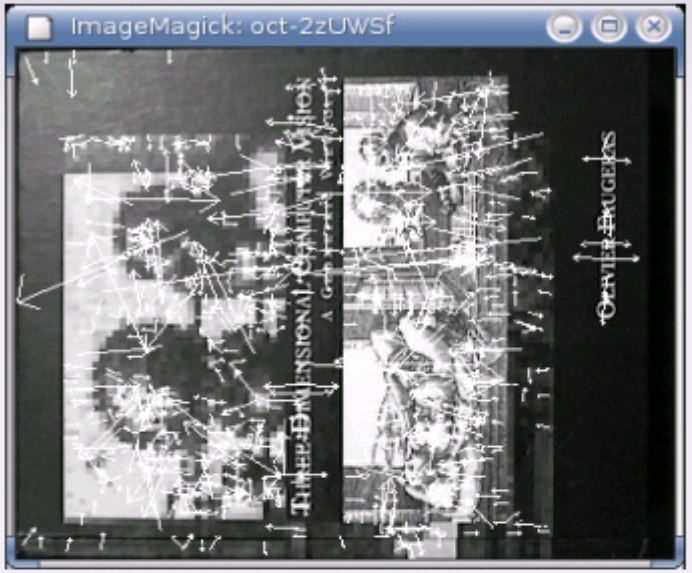
More in detail

When geometric properties of the scene -some planarities, parallelisms, orthogonalities, symmetries- are known, they can be used to obtain a 3D reconstruction (sometimes) from a single view. The reconstruction is also more accurate.

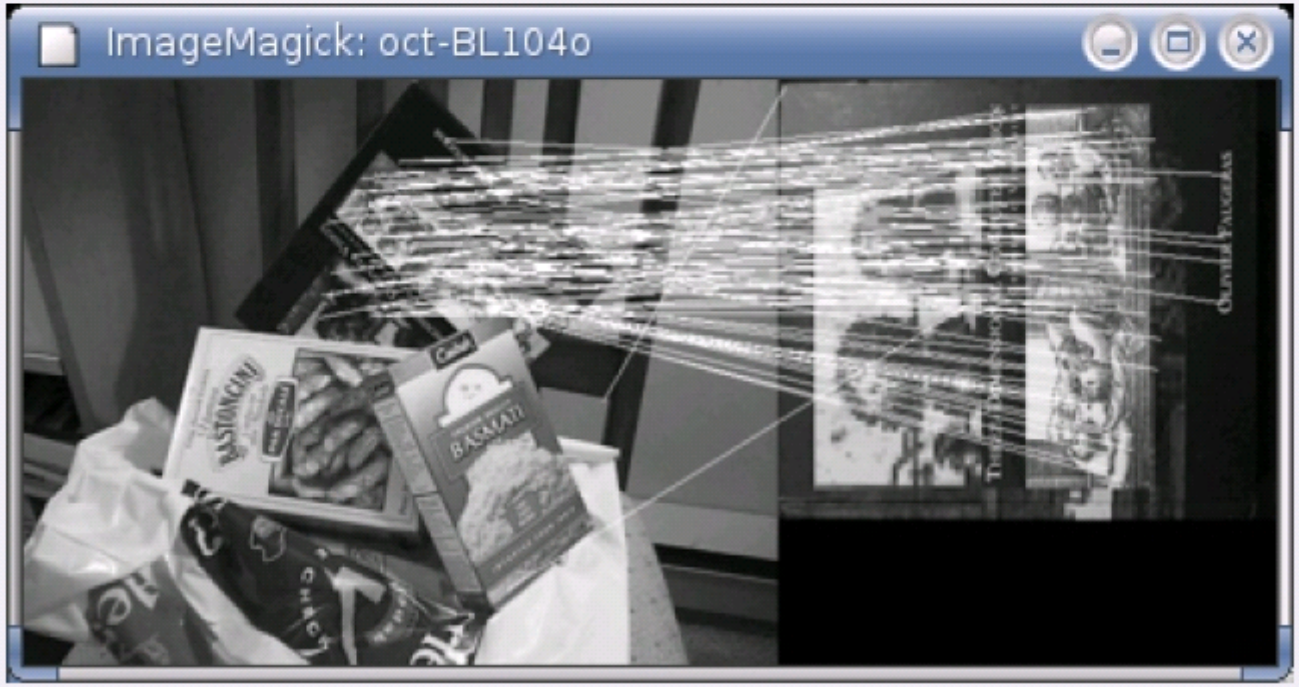
This reconstruction was obtained first by a non-iterative (but non-optimal) direct method and then refined by a solving with Levenberg-Marquardt a constrained least-squares problem. For better appeal, the texture from the original image is added with the vrml_surf() function.

Typical usage

Rather than identifying (right) and tracking (below) image features by hand, one can use David Lowe's [L99ICCV] functions sift() and match().

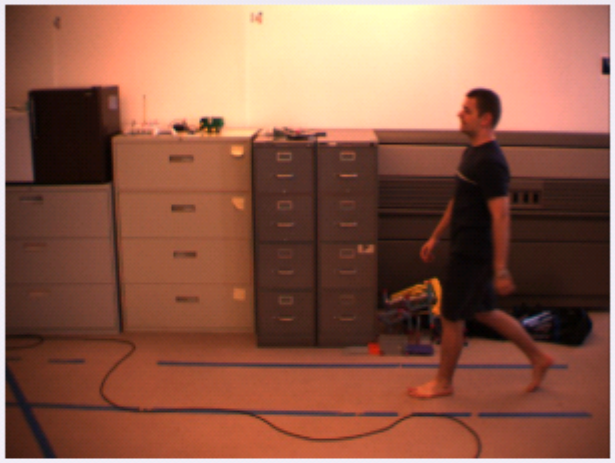


These functions are written in C and Matlab, but I did a patch for Octave.



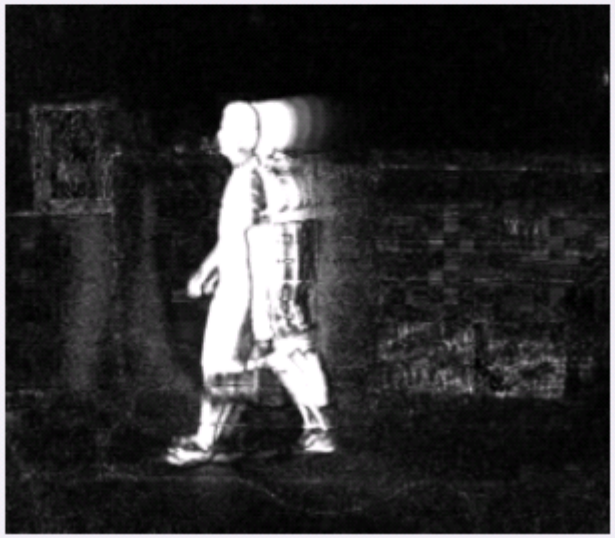
Typical usage

Image and video processing



Sequence of image

Region segmentati [CM02]



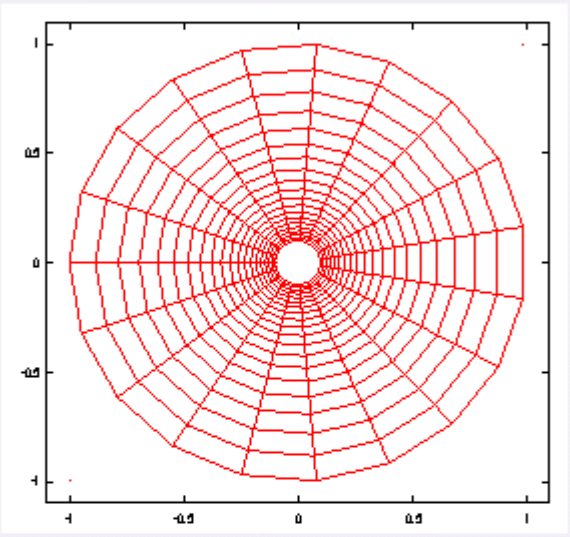
Temporal derivative

Foreground segmentation

Typical usage

Image and video processing: remapping and distorting images.

Original image



Log-polar pixel layout

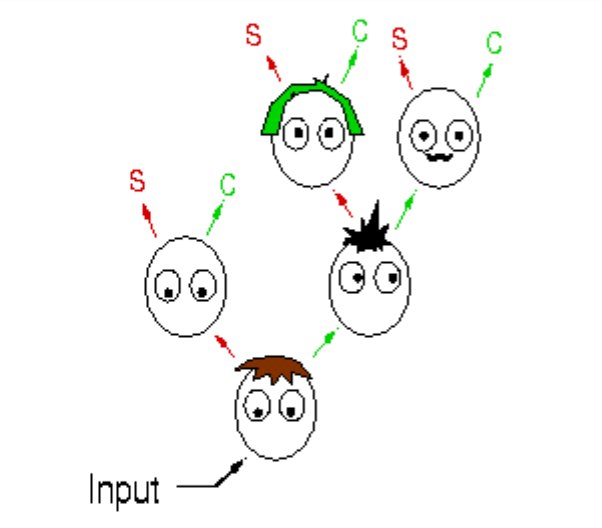
Log-polar image

Bird's eye view

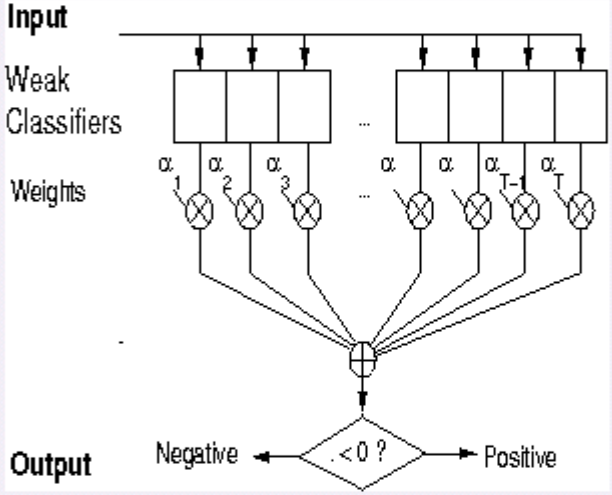
Typical usage

Machine learning

Decision trees



Adaboost/cascades



More in detail

Machine learning

I implemented Adaboost [SS99ML] and a decision tree algorithm [G04]. Below are results of these algorithms applied to face detection.

Faces detected by Adaboost are in green boxes, those by Adatree [G04] in red. Yellow boxes contain faces detected by both. Adaboost and Adatree leverage weak classifiers defined from Haare wavelets [VJ01]. They were trained using 1500 faces and as many non-face images.

What a language should provide

- Matrix calculator / shell
- Fast prototyping / unobtrusive language.
- Reliability (when there's a bug, it's in my code, not in Octave)
- Scripted and interactive image manipulation
- 3D visualization
- Number crunching

Options: Maple, C/C++, Splus, Octave/Matlab
Perl Data Language, Numpy, Scilab

Why I chose Octave ...

... when our Matlab license expired. My choice boiled down to Scilab and Octave. I chose Octave because it had, in 1998:

- variable argument / return list
- keyboard
- eval(), system() functions
- Good licence
- Active development / mailing lists

Appreciated improvements

I greatly appreciated these:

- ND arrays (e.g. images)
- Optimization tools
- Ints (for images)
- Sparse matrices.

Hacked myself:

- 3D visualization
- get mouse coordinates over image, w/ zoom & scroll (coming w/ gnuplot 4.1?)

References

[CM] D. Comaniciu and P. Meer Mean Shift: A robust approach towards feature space analysis, IEEE Pattern Analysis and Machine Intelligence (PAMI), v.24, n.5, pp.603-619, 2002.

[G04] E. Grossmann. AdaTree : boosting a weak classifier into a decision tree. Proc. CVPR workshop on learning in computer vision and pattern recognition, 2004.

[GSV00ICPR] E. Grossmann and J. Santos-Victor, A closed-form Solution for Paraperspective Reconstructio, proc. Intl. Conference on Pattern Recognition, v.1, pp.864-867, 2000.

[L99ICCV] D. G. Lowe, "Object recognition from local scale-invariant features," Intl. Conf. on Computer Vision, pp. 1150-1157, 1999. http://www.cs.ubc.ca/~lowe/keypoints/. Octave patch at http://www.cs.uky.edu/~etienne/code/code.html

[PK97] C. J. Poelman and T. Kanade, A paraperspective factorization method for shape and motion recovery,IEEE PAMI, v.19, n.3,pp.206-218, 1997.

[SS] Schapire & Singer Improved boosting algorithms using confidence-rated predictions, Machine Learning, 1999.

[VJ01] P. Viola and M. Jones, Robust Real-time Object Detection, proc. ICCV workshop on statistical and computational theories of vision.