MegaWave
http://megawave.cmla.ens-cachan.fr

Jacques Froment

LMBA UMR CNRS
Université de Bretagne-Sud, France

June 27th, 2012

Outline

1. Overview
   - Basic concept
   - History
   - People
   - Present and future
   - Content
2. Examples of released modules
   - Fourier and wavelet transforms
   - Curves and level lines representation and applications
   - Image and signal filtering
   - Other feature detection
3. Example of a module's source : fftconvol.c

Outline

1. Overview
   - Basic concept
   - History
   - People
   - Present and future
   - Content
2. Examples of released modules
   - Fourier and wavelet transforms
   - Curves and level lines representation and applications
   - Image and signal filtering
   - Other feature detection
3. Example of a module's source : fftconvol.c

Outline

1. Overview
   - Basic concept
   - History
   - People
   - Present and future
   - Content
2. Examples of released modules
   - Fourier and wavelet transforms
   - Curves and level lines representation and applications
   - Image and signal filtering
   - Other feature detection
3. Example of a module's source : fftconvol.c

MegaWave : why ?

Main goals of MegaWave :

- Help researchers to write codes on signal processing and image analysis;
- Facilitate the sharing of codes between researchers;
- Enable reproducible research.

MegaWave : how ?

Main ideas behind MegaWave :

- Code an algorithm (*module*) in the C language without worry about input/output;
- Interface code is automatically generated depending on the context;
- Including automatic generation of documentation (excluding description of the algorithm).

Outline

1. Overview
   - Basic concept
   - History
   - People
   - Present and future
   - Content
2. Examples of released modules
   - Fourier and wavelet transforms
   - Curves and level lines representation and applications
   - Image and signal filtering
   - Other feature detection
3. Example of a module's source : fftconvol.c

Main released versions

Date	Version	Main changes
MegaWave1, supporting lab : **CEREMADE**		
1988	-	Initial release (C code)
MegaWave2, supporting lab : CEREMADE		
1994	1.00	Initial release, support for HP-UX and Sparc SunOS 4
1995	1.02	SunOS 5 (Solaris)
1997	1.05	SGI IRIX 5
MegaWave2, supporting lab : **CMLA**		
2000	2.00	Open kernel; work on Linux i386
2002	2.10	Lionel Moisan integrates new modules
2007	3.00	Light preprocessor removes lot of incompatibility issues

Latest internal versions

Date	Main changes
2009	Rewriting by Nicolas Limare to use and follow common standards
2010-2012	(personal version) support for 64-bits processors

Why the name MegaWave ?

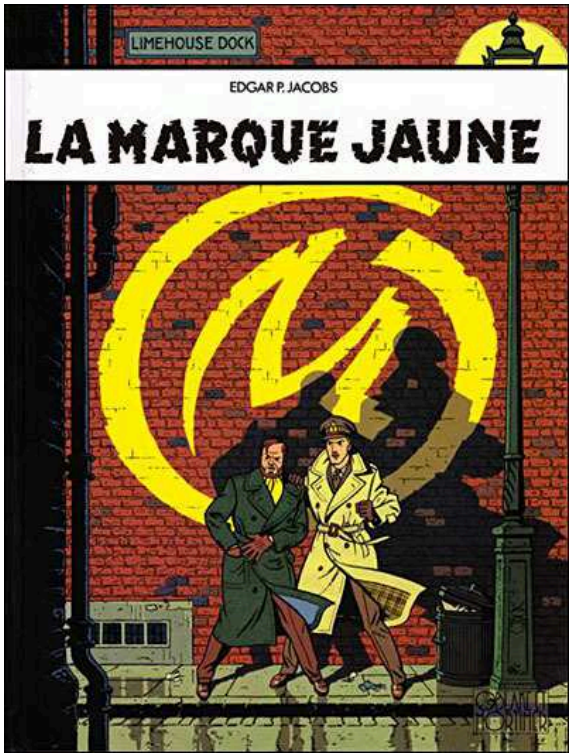
You may consider that

- The first code was about wavelet transforms;
- In the late 80's, the prefix **mega** (10^6) was associated with a large amount of power.

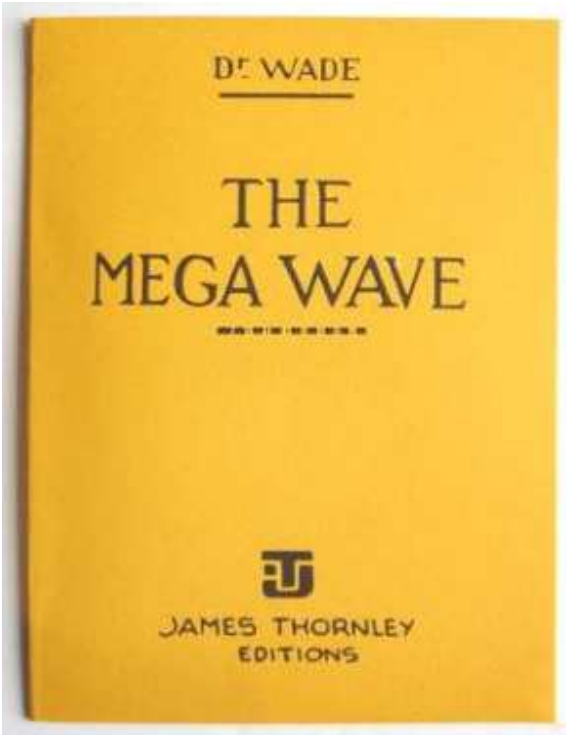
But the truth is out there ...

The true origin of MegaWave I

This name was chosen by Jean-Michel Morel in reference to the comic



The true origin of MegaWave II



As described in this book that is unfortunately sold out, the *Mega Wave* is a wave that can take control of the human brain !

Outline

1. Overview
   - Basic concept
   - History
   - People
   - Present and future
   - Content
2. Examples of released modules
   - Fourier and wavelet transforms
   - Curves and level lines representation and applications
   - Image and signal filtering
   - Other feature detection
3. Example of a module's source : fftconvol.c

Who contributed to MegaWave ?

- Jean-Michel Morel : as in charge of the image team at the CEREMADE then at the CMLA, primarily responsible for all;
- Jacques Froment : he created MegaWave1 then MegaWave2; try to keep the system up to date;
- Lionel Moisan : in charge of collecting and updating modules; main contributor in research modules;
- More than 30 researchers contributed in MegaWave2 by writing modules.

Who used MegaWave ?

- More than one hundred universities and public or private research centers;
- Used in some universities for teaching at Masters level.

Outline

1. Overview
   - Basic concept
   - History
   - People
   - Present and future
   - Content
2. Examples of released modules
   - Fourier and wavelet transforms
   - Curves and level lines representation and applications
   - Image and signal filtering
   - Other feature detection
3. Example of a module's source : fftconvol.c

Current issues

- The age of MegaWave may represent some disadvantages :
  - It does not use current standards and tools of open source softwares;
  - It does not use generic programming.
- IPOL project may decrease the value of using MegaWave
  - However goals of IPOL and MegaWave are different;
  - Being a on line journal, IPOL offers the best answer to reproducible research;
  - MegaWave offers a pleasant environment to generate standard source codes;
  - Such codes may be adapted to follow IPOL guidelines.
- The most critical issue is probably in human resources.

Outline

1. Overview
   - Basic concept
   - History
   - People
   - Present and future
   - Content
2. Examples of released modules
   - Fourier and wavelet transforms
   - Curves and level lines representation and applications
   - Image and signal filtering
   - Other feature detection
3. Example of a module's source : fftconvol.c

What is included in MegaWave ?

MegaWave is composed of three main parts :

- Set of commands to manage the system : see Volume 1, MegaWave2 User's Guide (83 pages);
- Library of functions to facilitate writing of new modules : see Volume 2, MegaWave2 System Library (269 pages);
- Library of 350 modules, to be called by another modules or as a command : see Volume 3, MegaWave2 User's Modules Library (668 pages).

Outline

1. Overview
   - Basic concept
   - History
   - People
   - Present and future
   - Content
2. Examples of released modules
   - Fourier and wavelet transforms
   - Curves and level lines representation and applications
   - Image and signal filtering
   - Other feature detection
3. Example of a module's source : fftconvol.c

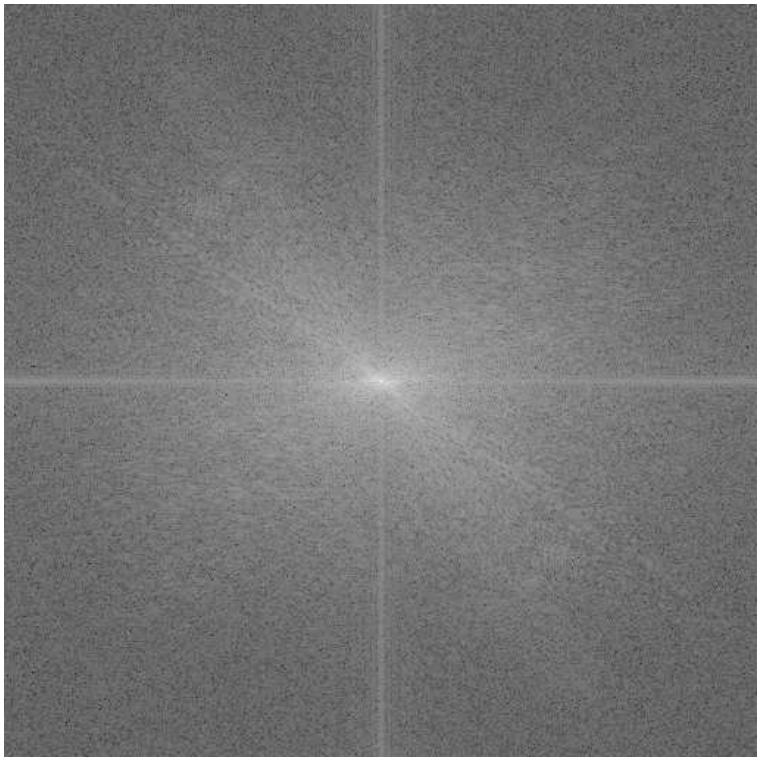
Outline

1. Overview
   - Basic concept
   - History
   - People
   - Present and future
   - Content
2. Examples of released modules
   - Fourier and wavelet transforms
   - Curves and level lines representation and applications
   - Image and signal filtering
   - Other feature detection
3. Example of a module's source : fftconvol.c

FFT in 1D and 2D

fft2dview

$\rightarrow$



Author : Lionel Moisan

Image rotation and translation using Fourier interpolation

fftrot

$\longrightarrow$

Author : Pascal Monasse

Image zooming by zero-padding

fftzoom

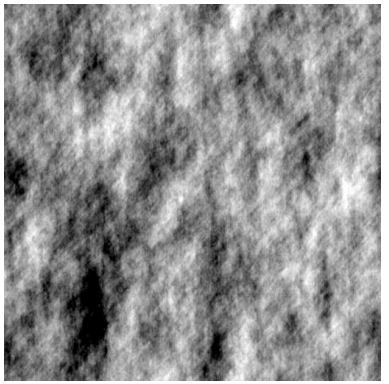
$\longrightarrow$

Author : Lionel Moisan

Phase randomization

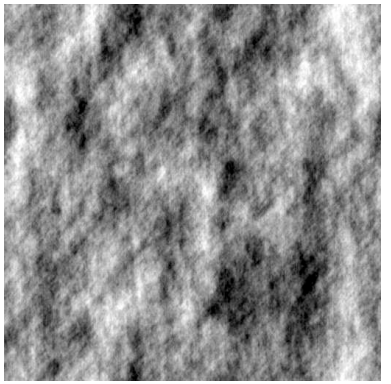
frandphase

$\rightarrow$



frandphase

$\rightarrow$



Author : Lionel Moisan

Wiener filtering

wiener

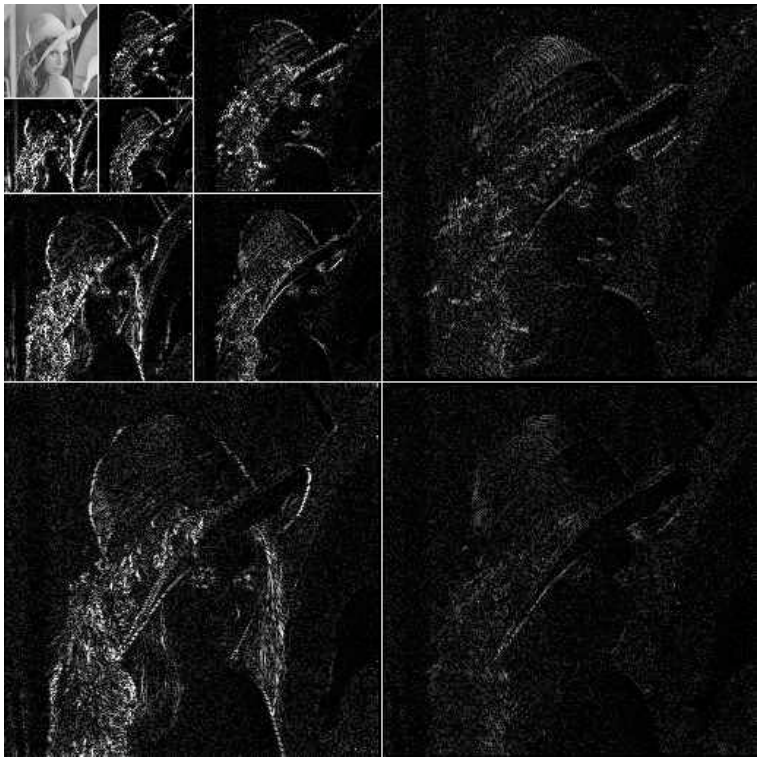
$\longrightarrow$

Author : Lionel Moisan

Orthogonal and biorthogonal wavelet transforms

biowave2

$\rightarrow$

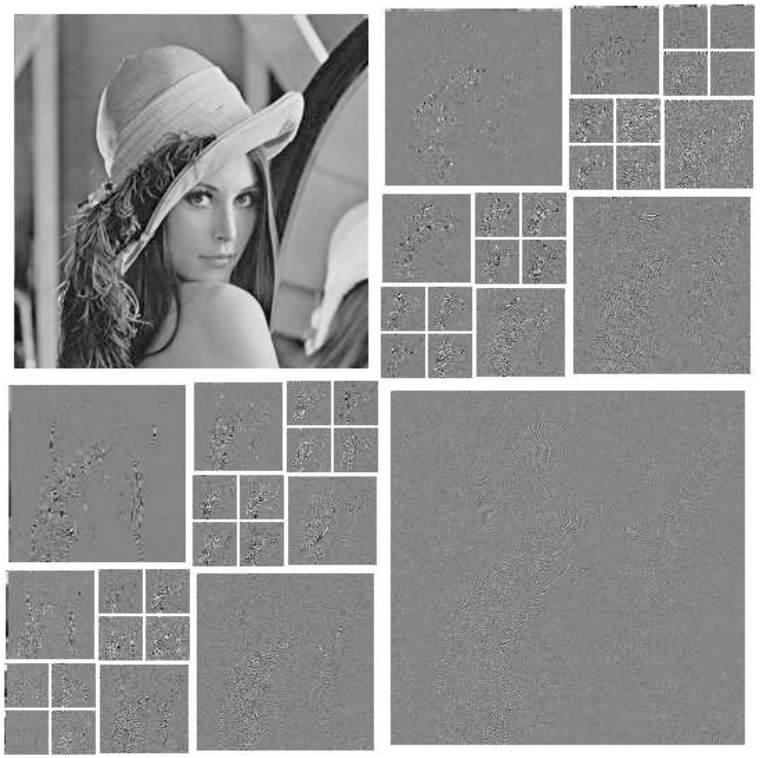


Author : Jean-Pierre D'Alès

2D wavelet packets decomposition

wp2ddecomp

$\rightarrow$



Author : François Malgouyres

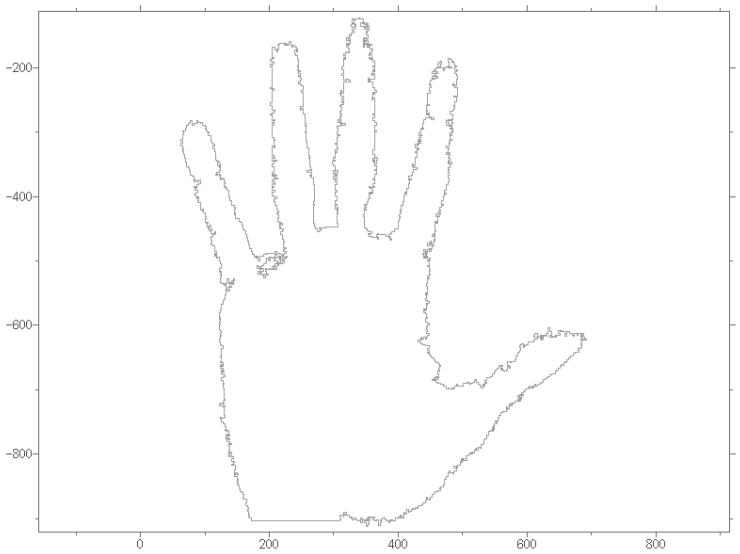
Outline

1. Overview
   - Basic concept
   - History
   - People
   - Present and future
   - Content
2. Examples of released modules
   - Fourier and wavelet transforms
   - Curves and level lines representation and applications
   - Image and signal filtering
   - Other feature detection
3. Example of a module's source : fftconvol.c

Geometric Affine Scale Space of Curves

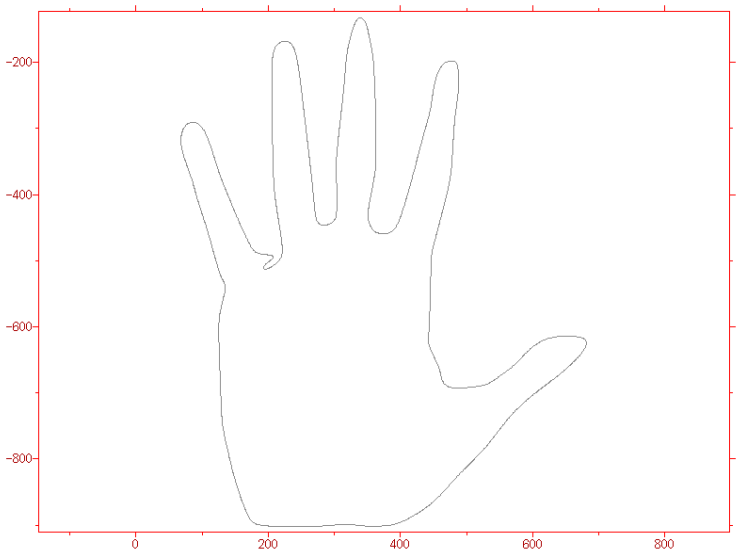
Evolution equation

$$\frac{\partial \mathbf{C}}{\partial t}(s,t) = \kappa^{1/3}(s,t)\,\mathbf{N}(s,t)$$



gass

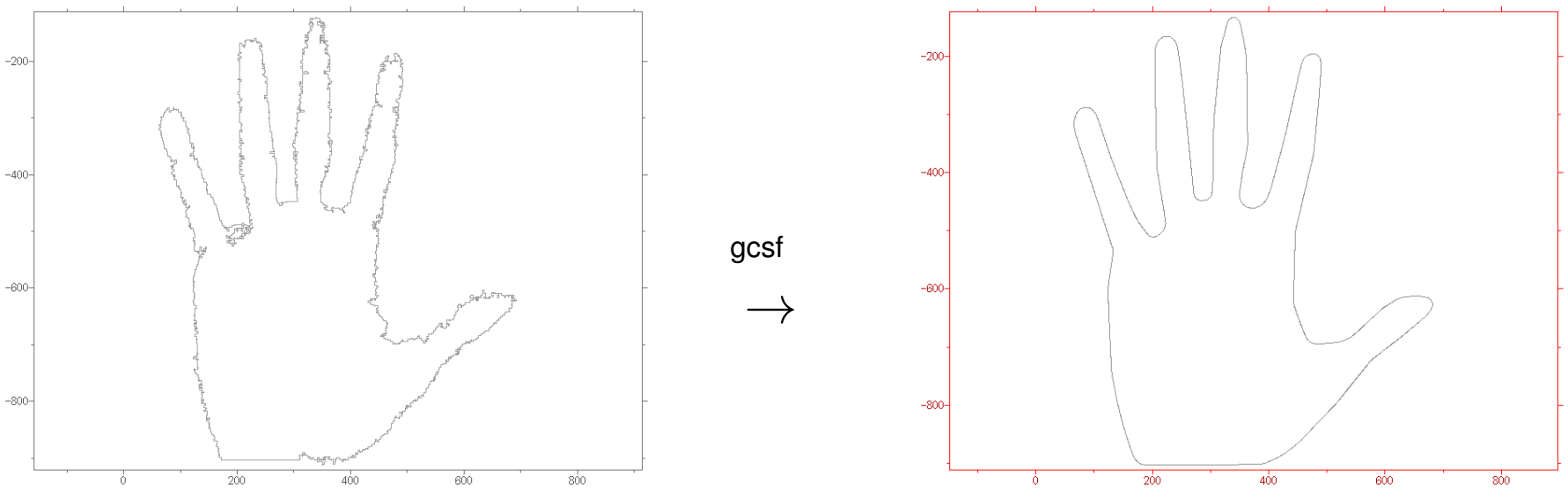
$\rightarrow$



Author : Lionel Moisan

Generalized Curve Shortening Flow of a curve

Evolution equation $\frac{\partial C}{\partial t} = (t\kappa)^g \mathbf{N}$

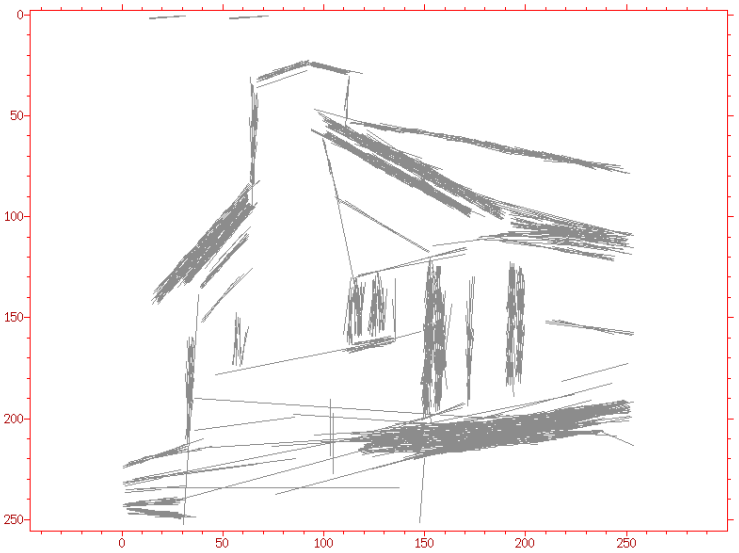


Authors : Frédéric Cao, Lionel Moisan

Detect meaningful alignments in an image

falign

$\longrightarrow$



Author : Lionel Moisan

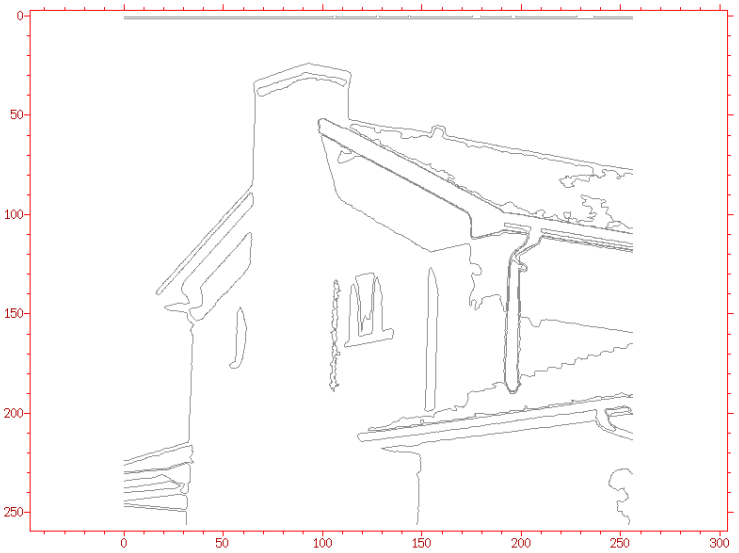
Detect maximal meaningful segments

falign_mdl

Authors : Lionel Moisan, Andrés Almansa

Extract meaningful contrasted level lines

ll_boundaries2

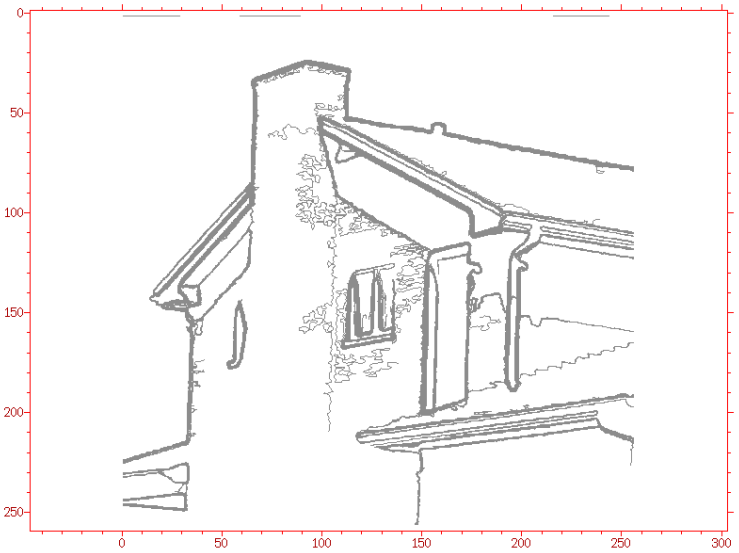


Authors : Lionel Moisan, Frédéric Cao

Extract maximal meaningful edges

ll_edges

$\rightarrow$



Author : Lionel Moisan

Grain filter

Use the Fast Level Set Transform (FLST)

fgrain

$\longrightarrow$

Authors : Pascal Monasse, Frédéric Guichard

Image disocclusion

disocclusion

$\rightarrow$

Author : Simon Masnou

Outline

1. Overview
   - Basic concept
   - History
   - People
   - Present and future
   - Content
2. Examples of released modules
   - Fourier and wavelet transforms
   - Curves and level lines representation and applications
   - Image and signal filtering
   - Other feature detection
3. Example of a module's source : fftconvol.c

AMSS and MCM by anisotropic diffusion

Affine Morphological Scale Space : $\frac{\partial u}{\partial t} = |Du|(curv(u))^{\frac{1}{3}}$

Mean Curvature Motion : $\frac{\partial u}{\partial t} = |Du|(curv(u))$

amss

$\rightarrow$

Authors : Frédéric Guichard, Lionel Moisan

AMSS as a stack filter

osamss

$\rightarrow$

Author : Lionel Moisan

Rudin Shock filter

$$\frac{\partial u}{\partial t} = -\mathrm{sgn}(\Delta u) \cdot |Du|$$

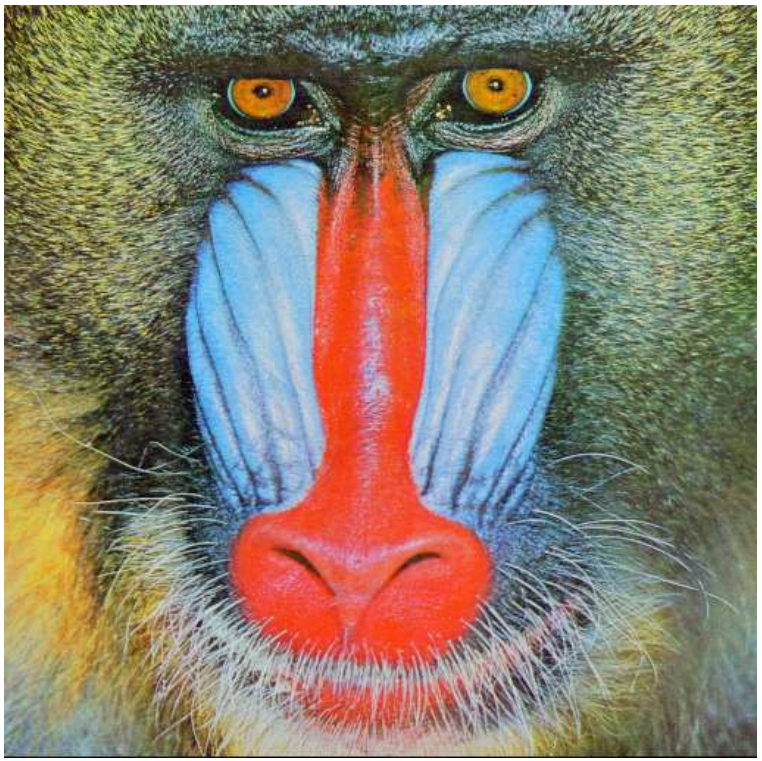
shock

$\rightarrow$

Author : Lionel Moisan

Image denoising by TV minimization

(Implemented with a relaxation algorithm)



cfdiffuse

$\longrightarrow$

Author : Antonin Chambolle

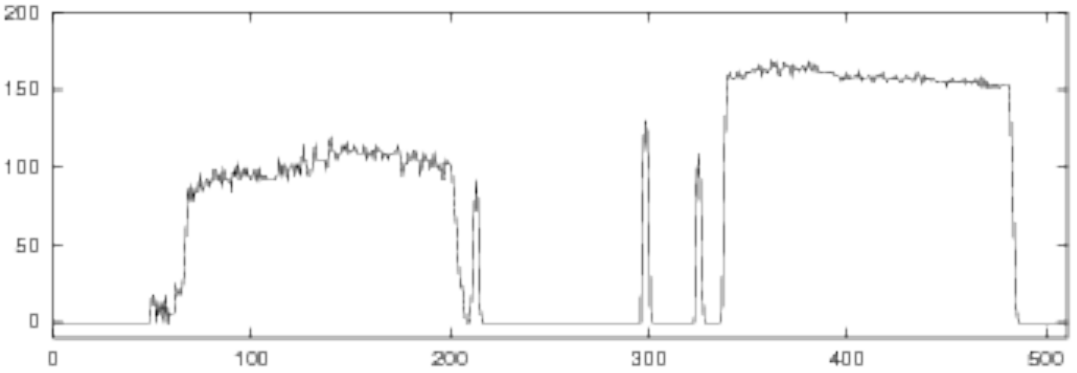
Image deblurring by TV minimization

tvdeblur

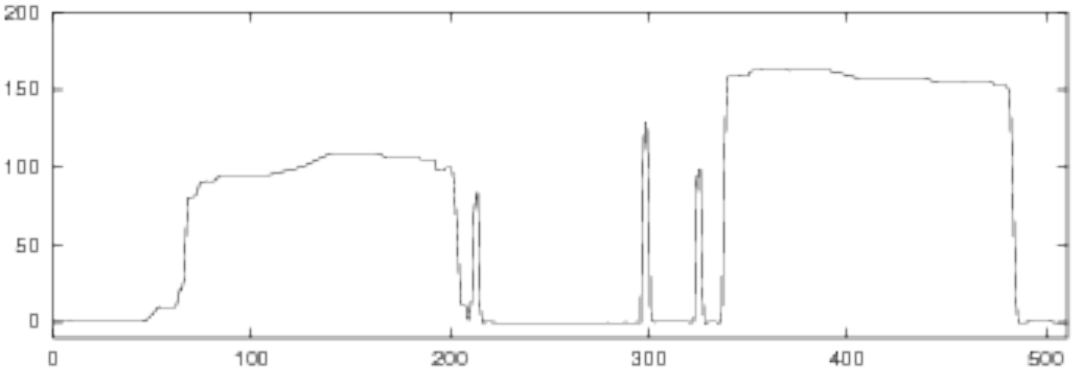
$\longrightarrow$

Author : Lionel Moisan

Signal restoration using TV and wavelets



↓ stvrestore



Author : Jacques Froment

Non-Local Means image denoising

nlmeans

$\longrightarrow$

Author : Lionel Moisan

Outline

1. Overview
   - Basic concept
   - History
   - People
   - Present and future
   - Content
2. Examples of released modules
   - Fourier and wavelet transforms
   - Curves and level lines representation and applications
   - Image and signal filtering
   - Other feature detection
3. Example of a module's source : fftconvol.c

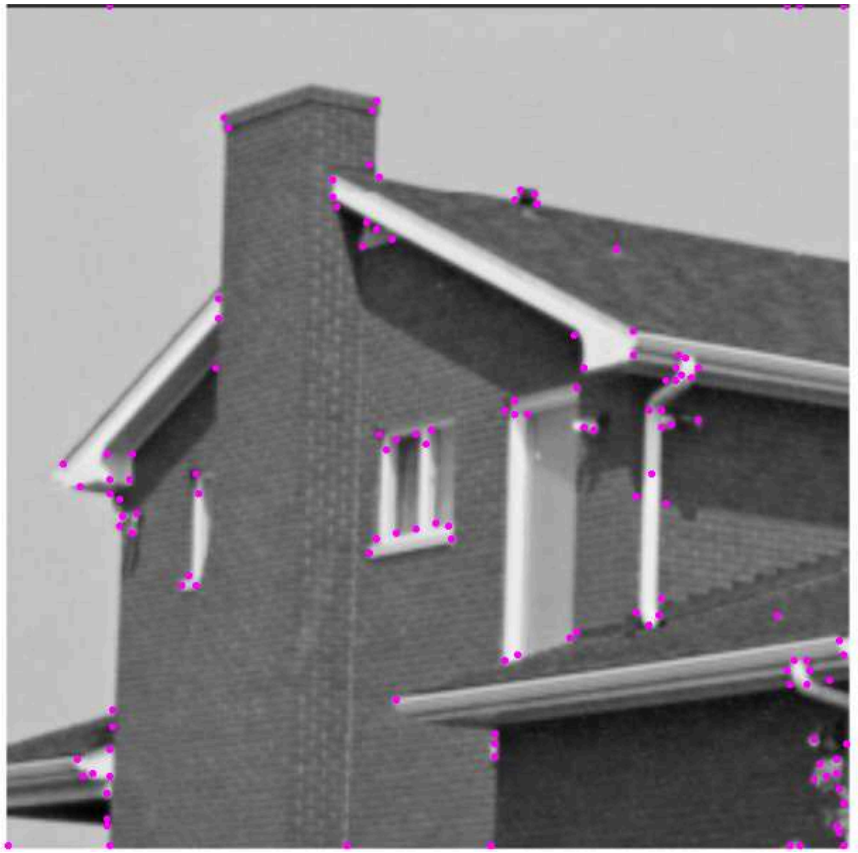
Canny-Deriche's Edge Detector

canny

$\longrightarrow$

Author : Yann Guyonvarc'h

Harris corner detector



harris

Author : Frédéric Cao

Image segmentation

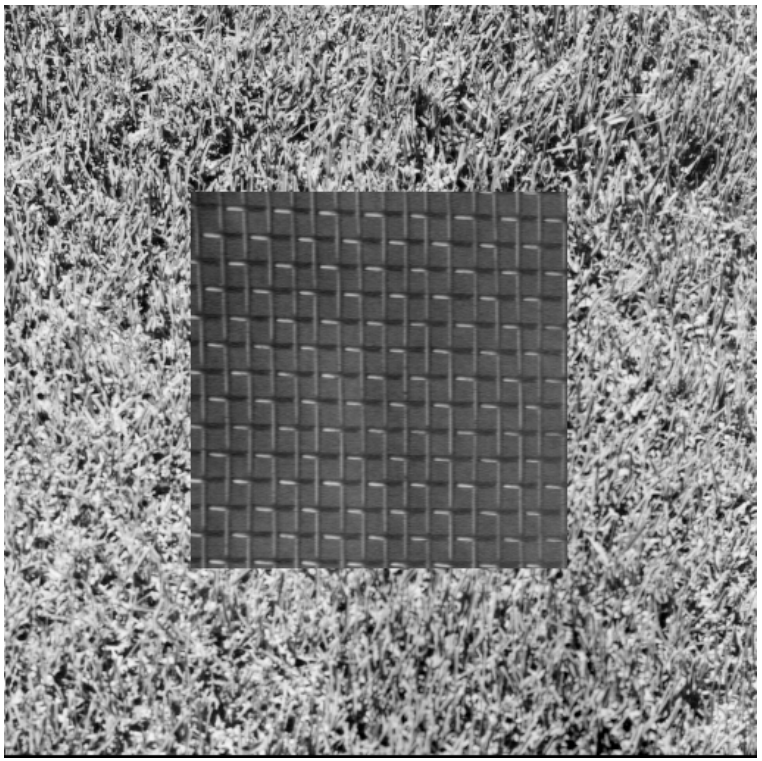
Region-growing method using the energy of Mumford and Shah

segct

Author : Georges Koepfler

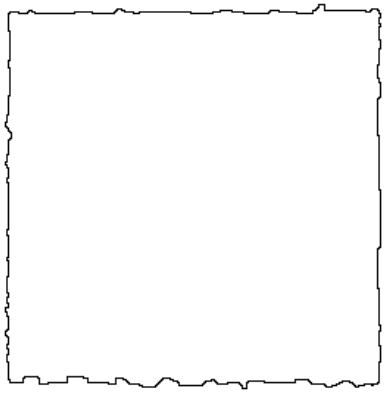
Texture segmentation

Multiscale version of the region-growing method using the energy of Mumford and Shah



segtxt

$\rightarrow$



Author : Yann Guyonvarc'h

Outline

1. Overview
   - Basic concept
   - History
   - People
   - Present and future
   - Content
2. Examples of released modules
   - Fourier and wavelet transforms
   - Curves and level lines representation and applications
   - Image and signal filtering
   - Other feature detection
3. Example of a module's source : fftconvol.c

What fftconvol computes

Convolution in the Fourier domain

- of an input gray level image `in`
- with a real kernel `filter`

$$\widehat{\text{out}}(i) = \widehat{\text{in}}(i) \times \text{filter}(i)$$

Source of fftconvol I

```
/*-------------------------- MegaWave2 Module -----------------------------*/
/* mwmodule
 name = {fftconvol};
 version = {"1.2"};
 author = {"Lionel Moisan"};
 function = {"2D Fourier-convolution of a fimage"};
 usage = {
    in->in           "input Fimage",
    filter->filter   "convolution filter in Fourier domain (Fimage)",
    out<-out         "output Fimage"
};
-------------------------------------------------------------------------*/
#include <stdio.h>
#include <math.h>
#include "mw.h"

extern void fft2d(); // Use this module

Fimage fftconvol(in,filter,out)
     Fimage in,filter,out;
{
  int i,nx,ny;
```

Source of fftconvol II

```
  Fimage re,im;

  nx = in->ncol; ny = in->nrow; // Size of the input image

  if (filter->ncol!=nx || filter->nrow!=ny) // Check image and filter size
    mwerror(USAGE,1,"Input image and filter dimensions do not match !\n");

  re = mw_new_fimage();  im = mw_new_fimage(); // Create images <re> and <im>

  fft2d(in,NULL,re,im,0); // Compute the 2D Fourier transform

  for (i=nx*ny;i--;) { // Multiplication in the Fourier domain
    re->gray[i] *= filter->gray[i]; im->gray[i] *= filter->gray[i];  }

  out = mw_change_fimage(out,ny,nx); // Allocate memory for the output image
  if (!out) mwerror(FATAL,1,"Not enough memory\n"); // If allocation failed

  fft2d(re,im,out,NULL,1); // Inverse 2D Fourier transform

  mw_delete_fimage(re); mw_delete_fimage(im); // Free memory <re> and <im>

  return(out); // Return the computed image (optional)
}
```