evropský sociální fond v ČR
EVROPSKÁ UNIE
MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Streamlining the Applied Mathematics Studies at Faculty of Science of Palacký University in Olomouc
CZ.1.07/2.2.00/15.0243

International Conference Olomoucian Days of Applied Mathematics

ODAM 2013

Department of Mathematical analysis and Applications of Mathematics
Faculty of Science
Palacký Univerzity Olomouc

Computational Intelligence for Image Processing

Irina Perfilieva

Centre of Excellence IT4Innovations
Division of the University of Ostrava
Institute for Research and Applications of Fuzzy Modeling

30. dubna 22, 701 03 Ostrava 1, Czech Republic
Irina.Perfilieva@osu.cz

ODAM 2013, Olomouc, June 12 - 14

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

OP Výzkum
a vývoj pro inovace

Outline

Outline

Image Processing in Problems

- ✓ Representation
- Quality Improving (contrast enhancement, sharpening, etc.)
- ✓ Edge Detection
- Compression (Reduction) and Reconstruction
- ✓ Reconstruction of Damages
- Fusion
- Segmentation
- ✓ Registration

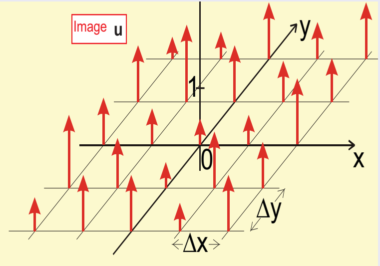
Computation Intelligence. Why?

- To bring intuition in a form of knowledge
- To keep connection between a problem and a method of solving
- To explain "**why it works**"

Image as an Object

Image

2D gray-scaled image **u** : $D \longrightarrow R$ is identified with **intensity function** of two variables such that the domain D is sampled, and the range R is quantized.



Domain -
$D = \{1, \ldots, N\} \times \{1, \ldots, M\}$
Range -
$R = \{0, 1, \ldots, 255\}$
Pixel -
image sample.

Representations of Image

Image

$$\mathbf{u} : \{1,\dots,N\} \times \{1,\dots,M\} \longrightarrow \{0,1,\dots,255\},$$

can be represented by:

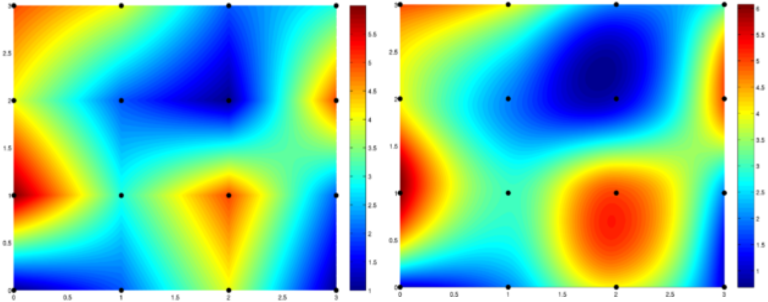
- *$N \times M$ matrix* $\mathbf{U} = (u_{ij})$ where $u_{ij} = \mathbf{u}(i,j)$
- *interpolating (approximating)* function

$$\hat{\mathbf{u}} : [1,N] \times [1,M] \longrightarrow [0,255],$$

such that for all $i = 1,2,\dots,N$, $j = 1,2,\dots,M$,

$$\mathbf{u}(i,j) = \hat{\mathbf{u}}(i,j) \quad (\mathbf{u}(i,j) \approx \hat{\mathbf{u}}(i,j)).$$

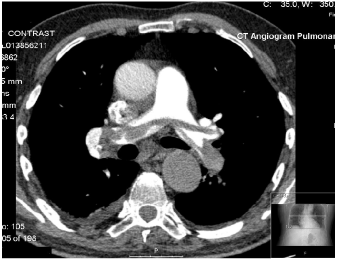
Representations by Interpolation

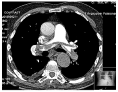
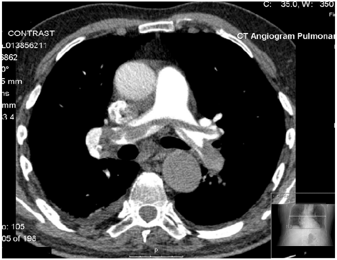


Bilinear (left) and bicubic interpolations as **magnifiers**.

Approximate Representation by F-transform

$$\mathbf{u} \Rightarrow F[\mathbf{u}] \Rightarrow \hat{\mathbf{u}}$$



$\mathbf{u}$ and $F[\mathbf{u}]$ are represented by matrices, $\hat{\mathbf{u}}$ – analytically.

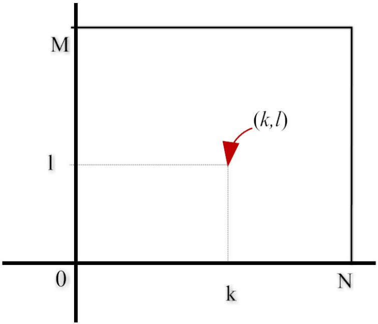
Outline

F-transform

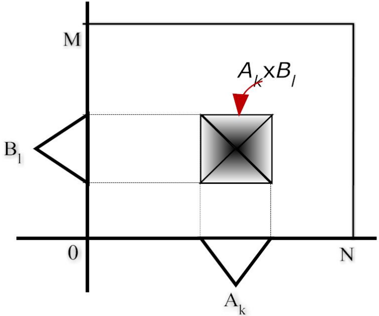
F-transform – result of a weighted projection of an image on a finite set of granules that establish a fuzzy partition of a domain.

Granulation of Domain $[1, N] \times [1, M]$

Rectangular area $[1, N] \times [1, M]$

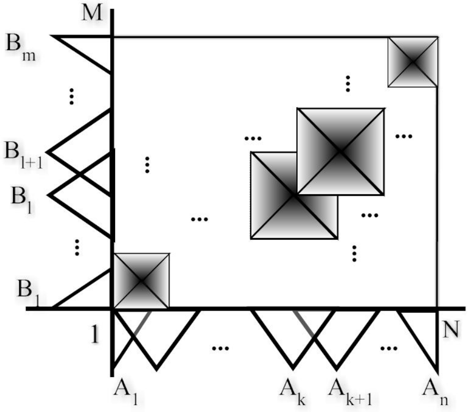


Granule $A_k \times B_l$



Fuzzy sets $A_k : [1, N] \longrightarrow [0, 1]$, $B_l : [1, M] \longrightarrow [0, 1]$.

Fuzzy Partition of $[1, N] \times [1, M]$



Granules $A_k \times B_l$, $k = 1, \ldots, n$, $l = 1, \ldots, m$, are associated with **selected nodes** (k, l), $3 \leq k \leq N$, $3 \leq l \leq M$.

Matrix Representation of Fuzzy Partition of $[1, N]$

- $A_1, \ldots, A_n$ – **fuzzy partition** of $[1, N]$
- Denote $a_{ki} = A_k(i)$, $a_k = \sum_{i=1}^{N} a_{ki}$, $a^i = \sum_{k=1}^{n} a_{ki}$, $k = 1, \ldots, n$, $i = 1, \ldots, N$
- $A_{n \times N} = \begin{pmatrix} a_{11} & \ldots & a_{1N} \\ \ldots & \ldots & \ldots \\ a_{n1} & \ldots & a_{nN} \end{pmatrix}$ – matrix of fuzzy partition
- $P(A)_{n \times n} = \begin{pmatrix} \frac{1}{a_1} & \ldots & 0 \\ \ldots & \ldots & \ldots \\ 0 & \ldots & \frac{1}{a_n} \end{pmatrix}$ – raw normalization
- $Q(A)_{N \times N} = \begin{pmatrix} \frac{1}{a^1} & \ldots & 0 \\ \ldots & \ldots & \ldots \\ 0 & \ldots & \frac{1}{a^N} \end{pmatrix}$ – column normalization

Direct F-transform of image u (Reduction)

- $\mathbf{u}$ is represented by matrix $\mathbf{U}_{N\times M} = (u_{ij})$ where $u_{ij} = \mathbf{u}(i,j)$
- fuzzy partition $A_1, \ldots, A_n$ – by matrix $A_{n\times N}$, $n < N$
- fuzzy partition $B_1, \ldots, B_m$ – by matrix $B_{m\times M}$, $m < M$

The $n \times m$ matrix $F[\mathbf{u}]$ is the direct **F-transform** of $\mathbf{u}$ w.r.t. $A_1 \times B_1, \ldots, A_n \times B_m$ where

$$F[\mathbf{u}]_{n\times m} = (P(A)A)\mathbf{U}_{N\times M}(P(B)B)^T$$

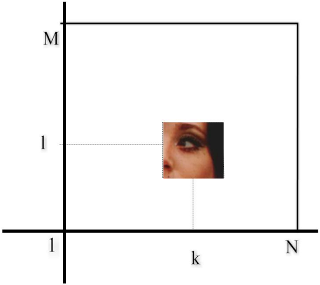
and the components are

$$F[\mathbf{u}]_{kl} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\mathbf{u}(i,j)A_k(i)B_l(j)}{\sum_{i=1}^{N}\sum_{j=1}^{M}A_k(i)B_l(j)}.$$

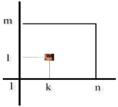
F-transform as a Weighted Projection

Image u

Image block on $A_k \times B_l$



$F[\mathbf{u}]_{kl}$



Inverse F-transform of image u (Magnification)

The $N \times M$ matrix $\hat{\mathbf{u}}$ is the representation of the inverse **F-transform** of $\mathbf{u}$ on $\{1, \ldots, N\} \times \{1, \ldots, M\}$ where

$$\hat{\mathbf{u}}_{N \times M} = (AQ(A))^T F[\mathbf{u}]_{n \times m} (BQ(B))$$

and the pixels are as follows

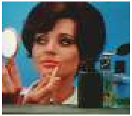
$$\hat{\mathbf{u}}(i,j) = \frac{\sum_{k=1}^{n} \sum_{l=1}^{m} F[\mathbf{u}]_{kl} A_k(i) B_l(j)}{\sum_{k=1}^{n} \sum_{l=1}^{m} A_k(i) B_l(j)},$$

$i = 1, \ldots, N, j = 1, \ldots, M.$

Inverse F-transform as Magnifier

$\hat{\mathbf{u}}$ – inverse F-transform of $\mathbf{u}$

$F[\mathbf{u}]$ – F-transform of $\mathbf{u}$



Main Properties of F-Transform

Best Approximation

Component $F[\mathbf{u}]_{kl}$, $k = 1, \ldots, n$, $l = 1, \ldots, m$, **minimizes the following criterion**

$$\sum_{i=1}^{N} \sum_{j=1}^{M} (\mathbf{u}(i,j) - x)^2 A_k(i) B_l(j)$$

IRAFM

Main Properties of F-Transform

F-Transform of Constants

Components of a **constant image** $\mathbf{u}$ where $\mathbf{u}(i, j) = c$ coincide with c, i.e. for all $k = 1, \ldots, n, l = 1, \ldots, m$,

$$F[\mathbf{u}]_{kl} = c.$$

Main Properties of F-Transform

Linearity

The map $\mathbf{F} : \mathbf{u} \mapsto F[\mathbf{u}]$ is **linear**, i.e. for all $\mathbf{u}, \mathbf{v}$, and for all $\alpha, \beta \in \mathbb{R}$,

$$\mathbf{F}(\alpha\mathbf{u} + \beta\mathbf{v}) = \alpha\mathbf{F}(\mathbf{u}) + \beta\mathbf{F}(\mathbf{v}).$$

Outline

F-Transform for Image Reduction

Image Reduction aims at adjusting image to a smaller screen.

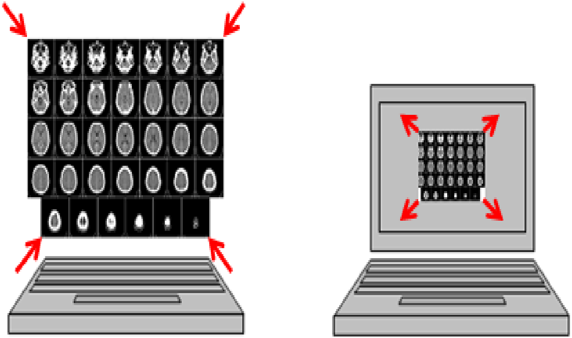


Image Reduction in Examples, Ratio $= 1/9$

The image reduction is performed by the **direct F-transform**.

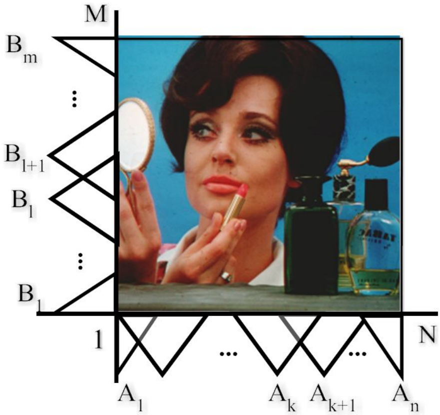


Image Reduction in Examples, Ratio $= 1/9$

Filtering by Reduction

Original image contains 20% of impulsive noise

MSE-quality of Reductions: FT versus Interpolation

Various reduction algorithms are compared on a set that contains 53 color images from http://sipi.usc.edu/database/database.php/volume=textures.

Type	FT	Bilinear	Bicubic	Lanczos
Min	**32.6**	43.6	44.7	41.9
1st Qu.	**81.2**	113.7	113.8	121.4
Median	**102.1**	145.6	150.0	170.8
Mean	**146.5**	197.3	199.6	212.0
3rd Qu.	**163.0**	246.3	242.0	262.6
Max	**517.0**	606	607.0	626.7

All statistics show the advantage of the FT-reduction !

Outline

Image Reconstruction

Image reconstruction takes a damaged image (noise, text, scratch, etc.) and restores it.

Assumption. The damaged area must be separated !

Image Reconstruction. Illustration

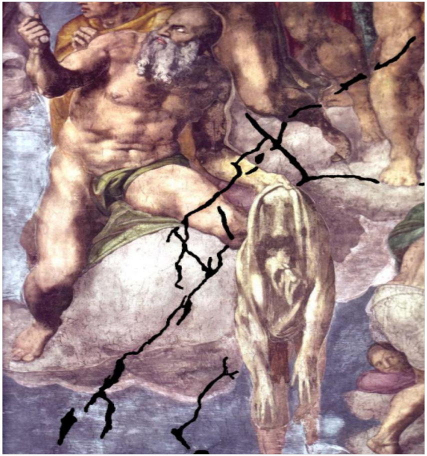


Image Reconstruction. Illustration

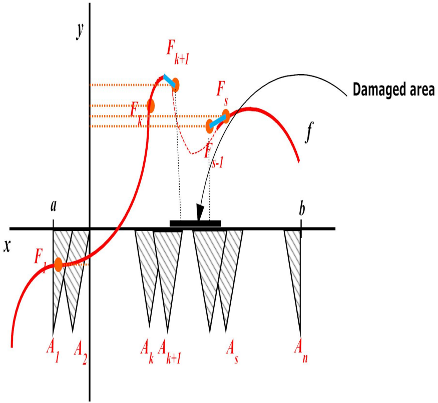
F-transform for Image Reconstruction

Proposed Method

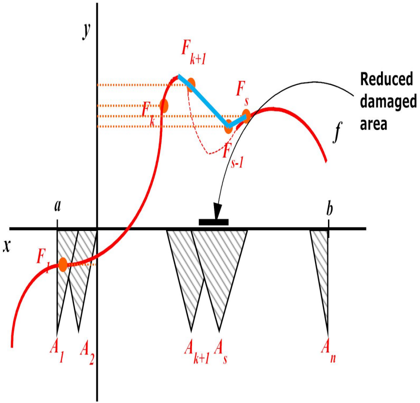
Repeat until the image is fully reconstructed

- choose a fuzzy partition and compute the respective inverse F-transform;
- if a pixel of the original image is undamaged then it is chosen for the reconstructed image, otherwise it is replaced by the value of the inverse F-transform.

F-transform Reconstruction in steps

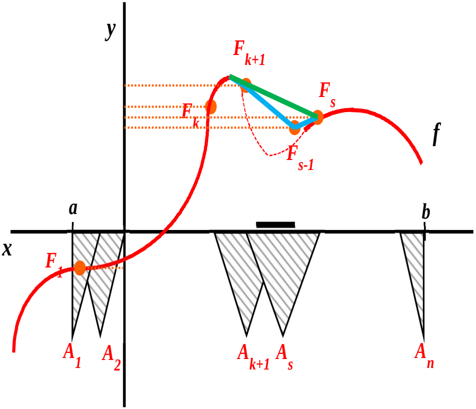


"Narrow" granules in partition



"Wide" granules in partition

F-transform versus Interpolation



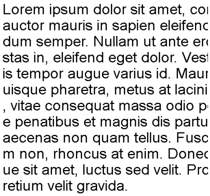
- Bilinear interpolation
- F-transform

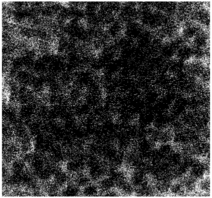
Illustration of the Method

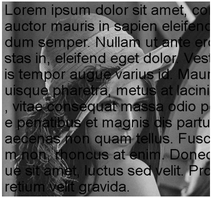
RMSD = 4.56

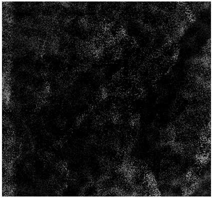
Illustration of the Results

Types of damage









Impaint **Text** **70% Noise**

Illustration of the Results

Reconstruction

MSE-quality of Reconstructions: FT versus Interpolation

Damage	**RBF**	**FT**
Noise	12.47	11.72
Inpaint	5.23	4.87
Text	4.62	4.56

Outline

IRAFM

"Edge" as a Notion

- "Edge" is an area where image intensity **abruptly** changes.
- Edge detection is aiming at searching **local maxima** of the gradient magnitude.

F^1-transform

- Let f be a differentiable function on $[a, b]$.
- The F-transform (with constant components) can be generalized to the **F^1-transform with linear components:**

$$F^1[f] = [F_1^1, \dots, F_n^1]$$

where for each $k = 1, \dots, n$

$$F_k^1 = c_{k,0}\ P_k^0 + c_{k,1}\ P_k^1$$

F^1-transform and Orthogonal Projections

- Every component

$$F_k^1 = c_{k,0}\, P_k^0 + c_{k,1}\, P_k^1, \quad k = 1, \ldots, n,$$

is a **weighted orthogonal projection** of f onto a linear subspace of functions with the basis of polynomials $P_k^0 = 1, P_k^1 = (x - x_k)$,

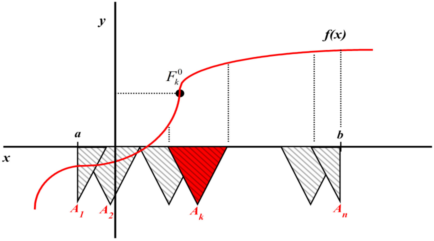
$$F_k^1 = c_{k,0} + c_{k,1}\,(x - x_k),$$

where

$$c_{k,0} = F[f]_k, \quad c_{k,1} \approx f'(x_k).$$

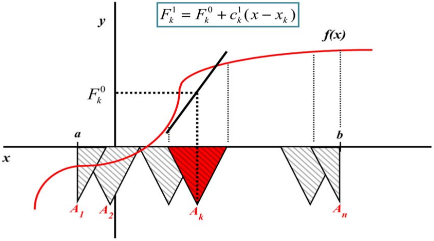
F-transform and F^1-transform

F-transform



$$F_k = F_k^0 = \frac{\int_a^b (f(x)\bullet 1) A_k(x)dx}{\int_a^b A_k(x)dx}$$

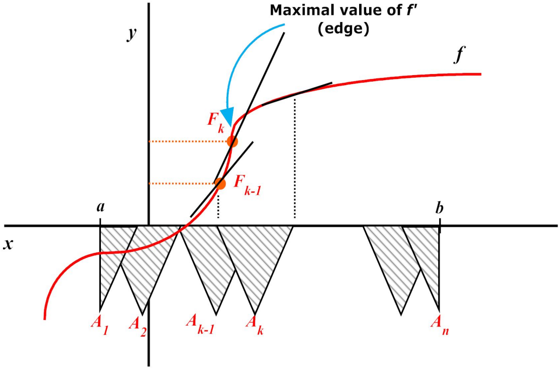
F^1-transform



$$c_k^1 = \frac{\int_a^b f(x)\bullet(x - x_k) A_k(x)dx}{\int_a^b (x - x_k)^2 A_k(x)dx}$$

F^1-transform for Edge Localization

Edge as a local maximum of a derivative magnitude



Approximation of the gradient

- Gradient vector $(\mathbf{u}_x, \mathbf{u}_y)$ of the image function $\mathbf{u}$ at (k, l) is approximated by $(c^1_{kl,1}, c^1_{kl,2})$ – vector of coefficients of the respective linear component of F^1-transform of $\mathbf{u}$:

$$F^1(x, y) = c^1_{kl,0} + c^1_{kl,1}(x - k) + c^1_{kl,2}(y - l),$$

$k = 1, \ldots, n,\ l = 1, \ldots, m$.

F^1-transform edge detection

Original image

F^1-transform detected edges

Original image - Lena

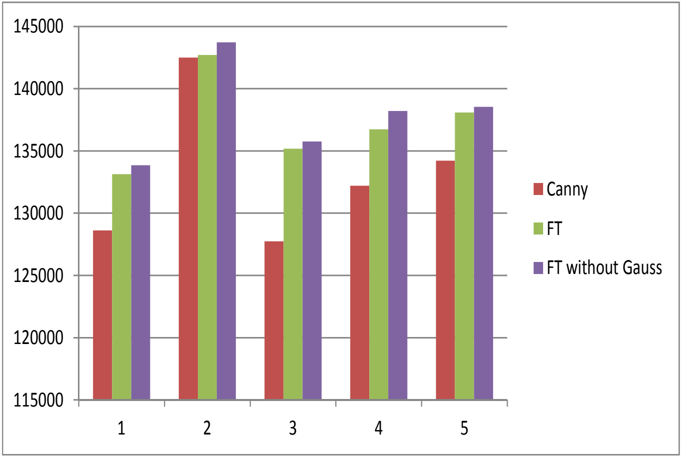
Blurred background, texture on the hat, perfect main contours, noise.

Comparison - Lena

“Classical” Canny algorithm

F^1-transform edge detector

Number of Detected Edges



Outline

IRAFM

Compression

Original “Cameraman”

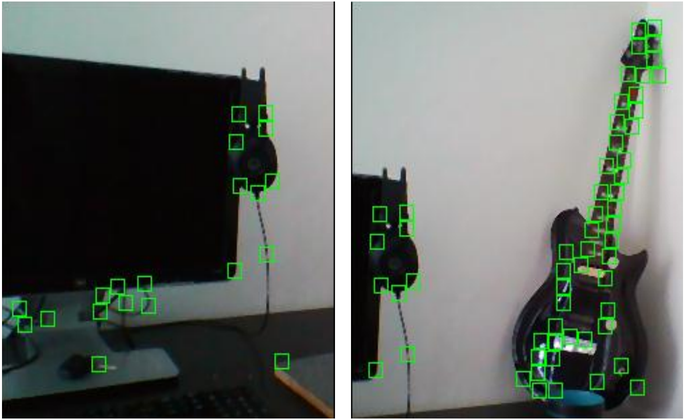
Compression

F-transform reconstructed

left CR=0.25, PSNR=37 (JPEG 39); **right** CR=0.44, PSNR=43 (46)

Registration

Registration



Registration

Outline

Conclusion

- F-transform makes manipulating with objects in a target domain much easier
- FT reduction is fast and efficient
- FT reconstruction is easier than the interpolation based reconstruction
- FT edge detection has better quality than Canny's detector

Conclusion

- F-transform makes manipulating with objects in a target domain much easier
- FT reduction is fast and efficient
- FT reconstruction is easier than the interpolation based reconstruction
- FT edge detection has better quality than Canny's detector

Conclusion

- F-transform makes manipulating with objects in a target domain much easier
- FT reduction is fast and efficient
- FT reconstruction is easier than the interpolation based reconstruction
- FT edge detection has better quality than Canny's detector

Conclusion

- F-transform makes manipulating with objects in a target domain much easier
- FT reduction is fast and efficient
- FT reconstruction is easier than the interpolation based reconstruction
- FT edge detection has better quality than Canny's detector