RT-LoG operator for scene text detection

Keynote talk at the
Computational Intelligence Laboratory (CIL)

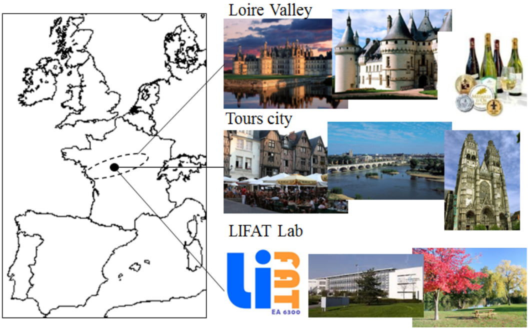
Mathieu Delalandre

LIFAT Laboratory, RFAI group, Tours city, France
firstname.lastname@univ-tours.fr

March 19, 2019

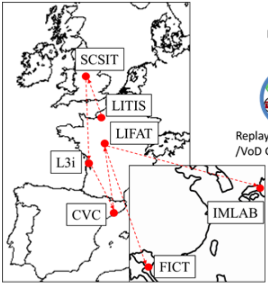
Mathieu Delalandre - CV in short (1/2)

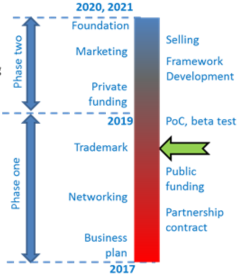
- PhD in computer science with 14 years of experience,
- Assistant Professor at the LIFAT Lab (Tours city, France),
- image processing (local detectors, template matching),
- application domains (video and document analysis),



Mathieu Delalandre - CV in short (2/2)

- international experience as research fellows in Europe (< 2009), visiting positions in Asia (> 2013),
- PhD supervisor of Cong Nguyen - VIED P911,
- Co Associate of the Todd.tv startup,
- more about myself: http://mathieu.delalandre.free.fr/.





Summary

Introduction (1/3)

Scene text detection: is a core image processing problem, the methods must be made:
(i) robust against text variabilities.

(a) (b) (c)

Figure: Text degradations (a) blurring (b) scaling (c) illumination changes

(ii) time-efficient to deal with the real-time, mass of data, low cost hardware, energy consumption issues.

Introduction (2/3)

RT-LoG: has attracted attention over the last years for scene text detection. It is the real-time implementation of the LoG operator (the Laplacian ∇^2 of the Gaussian function).

$$g(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1}$$

$$\begin{aligned} \nabla^2 g(x, y, \sigma) &= g_{xx}(x, y, \sigma) + g_{yy}(x, y, \sigma) \\ &= \frac{1}{2\pi\sigma^4} \left(\frac{x^2+y^2}{\sigma^2} - 2 \right) e^{-\frac{x^2+y^2}{2\sigma^2}} \end{aligned} \tag{2}$$

A discrete mask obtained from (2) is applied with convolution to obtain a LoG-filtered image.

Introduction (3/3)

RT-LoG: the LoG filtering must be embedded into a full pipeline to design an end-to-end operator.

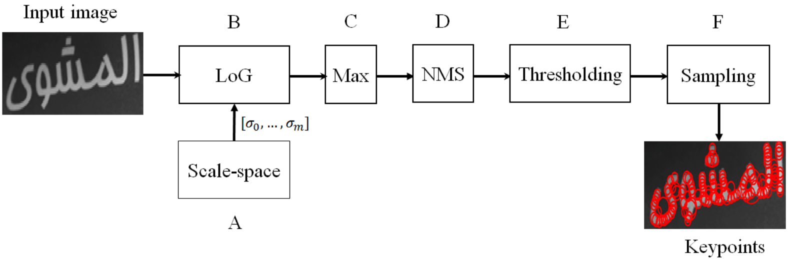


Figure: End-to-end operator

The end-to-end operator can be tuned in RT-LoG with optimization in the spatial and scale-space domains (steps A, B).

Summary

Fast spatial LoG filtering (1/5)

Key problem: the LoG filter (2) is not separable, convolution has a complexity $O(N\omega^2)$ with N, ω^2 the image and mask sizes.

Strategy: fast LoG filtering applies an estimator cascade methodology LoG $\approx$ DoG $\approx \widehat{DoG}$ to shift to a $O(N)$ complexity.

LoG $\approx$ DoG: approximation with the Difference of Gaussian (DoG), $k \in]1, \sqrt{2}]$ controls the precision of the scale-space derivative.

$$g(x, y, k\sigma_2) - g(x, y, \sigma_2) \approx (k-1)\sigma^2 \nabla^2 g(x, y, \sigma_1) \qquad (3)$$

Fast spatial LoG filtering (2/5)

DoG $\approx \widehat{DoG}$: fast Gaussian filtering methods can accelerate the filtering with approximation of the Gaussian kernel $\widehat{g}(x, y, \sigma)$.

Category	Methods	Time optimization	Accuracy
Box filter	Box	$++$	$+++$
	SII	$+++$	$++$
	KII	$+$	$+$
Recursive filter	Deriche	$++$	$++$
	TCF	$+++$	$+++$
	VYV	$+$	$++$

Table: Fast Gaussian filtering $(+++)$ best case $(+)$ medium case

The box filter method is common to approximate the DoG.

Fast spatial LoG filtering (3/5)

DoG $\approx \widehat{DoG}$: while using the box filter method, $\widehat{g}(x, y, \sigma)$ is obtained by a linear combination of averaging filters $\Pi_i(x, y)$ with weight parameters λ_i. $n \in [4, 6]$ is the number of filters.

$$\widehat{g}(x, y, \sigma) = \sum_{i=1}^{n} \lambda_i \Pi_i(x, y) \tag{4}$$

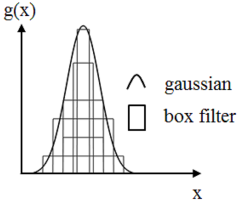


Figure: Approximation $\widehat{g}(x, y, \sigma)$ of a gaussian function

Fast spatial LoG filtering (4/5)

DoG $\approx \widehat{DoG}$: the n, $\Pi_i(x,y)$, λ_i parameters can be obtained with minimization of the MSE and regression (e.g. LASSO).

$$MSE = \sum_{(x,y)\in[\pm 3\sigma]} (g(x,y) - \widehat{g}(x,y))^2 \tag{5}$$

We can approximate the DoG operator (3) by a $\widehat{DoG}$ estimator with two sets of box filters.

$$\begin{aligned} DoG \approx \widehat{DoG} &= \widehat{g}(x,y,k\sigma) - \widehat{g}(x,y,\sigma) \\ &= \sum_{i=1}^{n} \lambda_i \Pi_i(x,y) - \sum_{j=1}^{n} \lambda_j \Pi_j(x,y) \end{aligned} \tag{6}$$

Fast spatial LoG filtering (5/5)

DoG $\approx \widehat{DoG}$: the filtered image is obtained by convolution between the input image $f(x, y)$ and the $\widehat{DoG}$ operator.

$$\begin{aligned} & (\widehat{g}(x, y, k\sigma) - \widehat{g}(x, y, \sigma)) \otimes f(x, y) \qquad (7) \\ = \; & \widehat{g}(x, y, k\sigma) \otimes f(x, y) - \widehat{g}(x, y, \sigma) \otimes f(x, y) \\ = \; & \sum_{i=1}^{n} \lambda_i \Pi_i(x, y) \otimes f(x, y) - \sum_{j=1}^{n} \lambda_j \Pi_j(x, y) \otimes f(x, y) \end{aligned}$$

The $\Pi_i(x, y) \otimes f(x, y)$ products of Eq. (7) can be obtained with integral image at a complexity $O(N)$.

Summary

Scale-space representation (1/3)

Key problem: a maximum response corresponds to a stroke. It relies on the correlation between σ and the size of the stroke. A filter bank $[\sigma_0, \ldots, \sigma_m]$ must be used for scale-invariance.

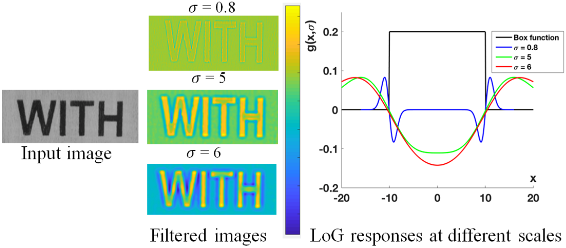


Figure: Text detection with different σ values

Scale-space representation: is the design of a time-efficient and accurate filter bank.

Scale-space representation (2/3)

Stroke model: defines how an optimum parameter σ_s is able to be derived from the stroke width parameter w within a DoG operator.

$$\sigma_s = \varpi(w) = \frac{w}{2k}\sqrt{\frac{k^2-1}{2lnk}} \tag{8}$$

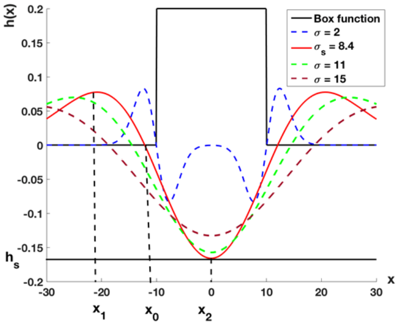


Figure: LoG responses around σ_s to a boxcar function of size $w = 21$

Scale-space representation (3/3)

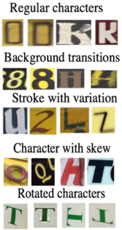
Stroke model: an optimal quantization is attained with $\sigma_s = \varpi(w)$ and $w \in [w_{min}, w_{max}]$ as a discrete value. This requires $2(m+1)$ Gaussian kernels with $m = w_{max} - w_{min}$.

Summary

Key-point detection (1/7)

Key problem: to characterize the RT-LoG as a key-point detector and estimator of the NRT-LoG (Non Real-time-LoG) corresponding to a LoG filtering $\oplus$ brute-force scale-space.

Dataset: there is no public dataset / groundtruth on the task. Segmented character images are used with normalization.



Images	7705
Classes	62
Size (Kpixel)	0.3K-10K
$[w_{min}, w_{max}]$	[5, 30]
Resolution of full images	640 x 480 (VGA)

Figure: The subset of segmented characters of The Chars74K dataset

Key-point detection (2/7)

Groundtruthing process: targets near optimum parameters for the NRT-LoG, fixed by a user and a closed-loop methodology.

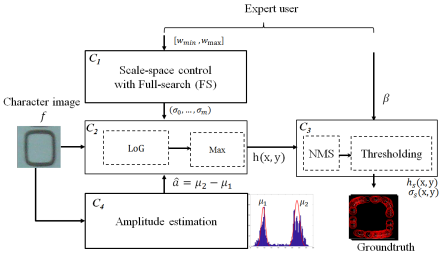
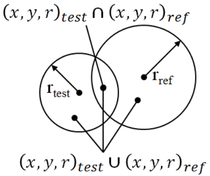


Figure: Semi-automatic process for groundtruthing

Key-point detection (3/7)

Characterization metric: the repeatability is standard for local detectors. Two regions are repeated if they respect an overlap error ϵ. $r = w/2$ is fixed with the stroke model Eq. (8).



$$1 - \frac{|(x,y,r)_{test} \cap (x,y,r)_{ref}|}{|(x,y,r)_{test} \cup (x,y,r)_{ref}|} \leq \epsilon$$

Figure: The repeatability criteria

The repeatability score is (9), the ϵ parameter is relaxed to get a ROC-like curve with $\epsilon = 0.4$ the maximum overlap error.

$$r(f_{ref}, f_{test}, \epsilon) = \frac{D(f_{ref}, f_{test}, \epsilon)}{Mean(n_r, n_t)} \tag{9}$$

Key-point detection (4/7)

Characterization protocol: the NRT-LoG architecture has been relaxed step-by-step to get RT-LoG operators. The goal is to characterize approximation at any stage.

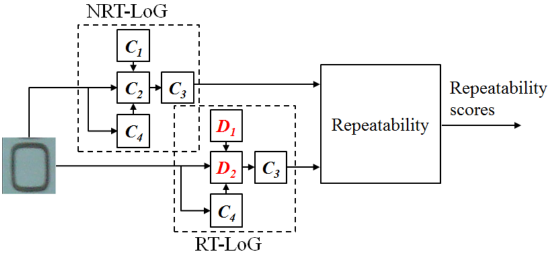
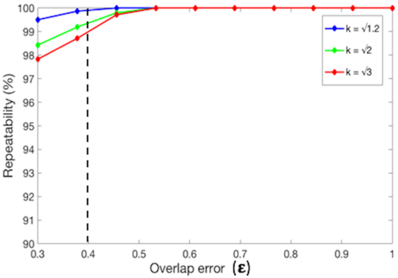


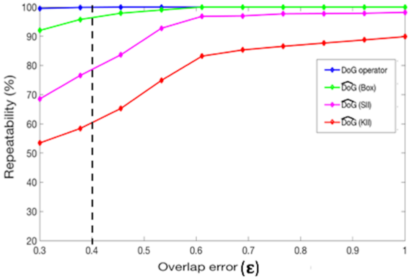
Figure: The characterization protocol

Key-point detection (5/7)

Results I: fast spatial filtering with LoG $\approx$ DoG $\approx$ $\widehat{DoG}$.

- LoG $\approx$ DoG has almost none impact for $k \in]1, \sqrt{2}]$ ($<1\%$ of repeatability error at $\epsilon = 0.4$).
- DoG $\approx$ $\widehat{DoG}$ results in $< 5\%$ of repeatability error at $\epsilon = 0.4$ with box filtering ($n = 5$). The other methods introduce severe degradations.

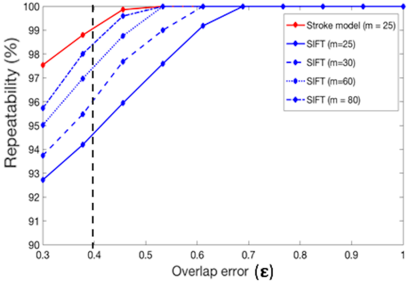




Key-point detection (6/7)

Results II: scale-space with Stroke Model (SM) vs. SIFT.

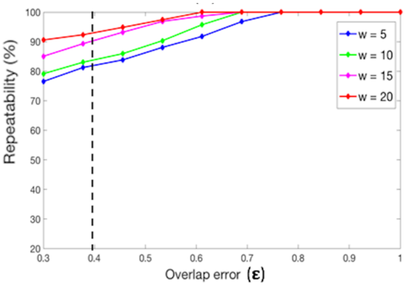
- *Equal kernels:* $m + 1 = 26$ Gaussian kernels resulting in 26 (SM), 52 (SIFT) scales with $k = \sqrt{1.06}$. The SM outperforms SIFT with a 5% repeatability gain at $\epsilon = 0.4$ and a near optimal result ($< 1\%$ of error).
- *Equal repeatability:* SIFT $(m = 80) \simeq SM$ $(m = 25)$ with $< 1\%$ repeatability error at $\epsilon = 0.4$.



Key-point detection (7/7)

Results III: low resolution.

- *Low resolution:* we filter out the key-points at w_i. 15% of repeatability error emerge for $w \leq 15$ due to the quantization. Full HD will give a robust repeatability with $w_{min} \approx 20$.



Summary

End-to-end text detection (1/3)

Key problem: to characterize the RT-LoG for end-to-end scene text detection and to compare with other RT operators (FASText, Canny Text Detector, MSER, . . .).

Dataset: the up-to-date dataset ICDAR2017 RRC-MLT is used offering the deeper challenge for scalability.

Task	Multiscript	Training images	Testing images	Ground truth level
Text scene	Yes	9K	9K	Bounding box / Word

Table: The dataset of ICDAR2017 RRC-MLT, (K) in thousands.

Characterization metric: the Intersection over Union (IoU) with the precision (P), recall (R) and Area Under the Curve (AUC).

End-to-end text detection (2/3)

Characterization protocol: to compare operators with the ICDAR2017 RRC-MLT dataset, text verification & segmentation must be applied. We use standard and time-efficient methods.

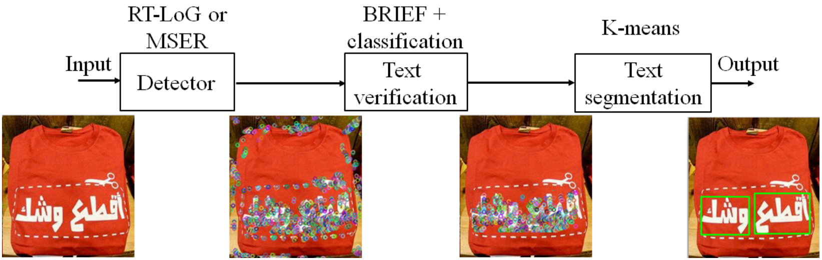
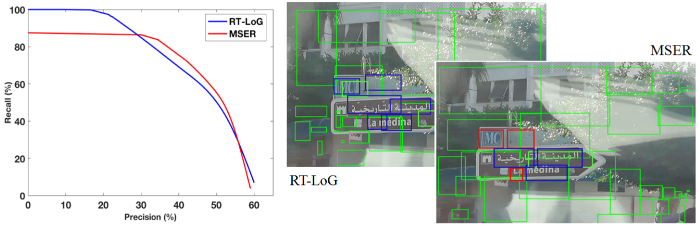


Figure: The system for operator comparison

End-to-end text detection (3/3)

Results: RT-LoG vs MSER.

- The P/R results are close (normalized AUC $\in [0,1]$ is 0.46 vs 0.44), the RT-LoG gets $R = 100\%$ whereas the MSER fails with some texts due to blurring.
- $R = 100\%$, $P = 20\%$ will result in $\simeq 20 - 100$ RoIs per image, the operator could be used within a R-CNN architecture.



(a) (b)

Figure: (a) Precision / Recall plot (b) detection results
(blue) true positive (green) false positive (red) missed case

Summary

Processing time (1/3)

Key problem: to characterize the processing time of the RT-LoG vs. the other RT and NRT operators.

Dataset: ICDAR2017 RRC-MLT is used where the parameters are $w = [5 - 55], m = 50$.

Characterization protocol: the operators are developed "from scratch", tested on one core and set with single/multi-thread and auto-vectorization.

Processing time (2/3)

Results I: comparison of operators with single thread.

- The LoG has a $O(N\omega^2)$ complexity not investigated.
- The RT operators (MSER, RT-LoG) result in one to two orders of magnitude for acceleration.
- The RT-LoG operator scales better at high resolutions with a $\simeq 2$ acceleration factor compared to MSER.

Resolutions	SD	HD	Full-HD	Quad HD	4K
SIFT (m=160)	1308	18816	40390	$\sim$	$\sim$
SIFT (m=50)	409	5880	12622	$\sim$	$\sim$
MSER	121	850	2075	7496	14551
RT-LoG	110	621	1284	4550	8406

Table: Processing time of operators in (ms), single thread / core C++/ Mac- OS 2.2 GHz Intel Core i7

Processing time (3/3)

Results II: FPS of the RT-LoG operator with multi-thread.

- the RT-LoG can support a high FPS up to the HD resolution.
- $\simeq \times 6 - 9$ acceleration factor will be obtained with a full parallelism (multi-core, intrinsic instructions).
- The RT-LoG is designed for the real-time processing of the Quad HD resolution with a regular CPU / full parallelism.

Threads	2	4	8	16
SD	13.3	17.1	30.1	54
HD	3.8	8.5	16.3	31.25
Full-HD	0.8	1.9	3.7	9.8
Quad-HD	0.2	0.4	1	3.3

Table: FPS with the RT-LoG operator, multi-thread / single core C++ / Mac- OS 2.2 GHz Intel Core i7 system

Summary

- CV in short
- State-of-the-art
  - Introduction
  - Fast spatial LoG filtering
  - Scale-space representation
- Performance evaluation
  - Key-point detection
  - End-to-end text detection
  - Processing time
- Conclusions and perspectives
  - Conclusions and perspectives

Conclusions and perspectives (1/2)

Conclusions: The RT-LoG

- results in one to two orders of magnitude for acceleration.
- guaranties a near-exact LoG filtering RT-LoG $\approx$ LoG.
- outperforms the MSER operator with close performances (AUC $= 0.46$ vs 0.44) and a $\simeq 2$ acceleration factor.
- is designed for a $\simeq 30$ FPS / Quad HD with a regular CPU.

Conclusions and perspectives (2/2)

Perspectives: The RT-LoG will be

- tested against other RT operators (FASText, Canny Text Detector, ...) for end-to-end text detection.
- tested within a R-CNN architecture while keeping the real-time constraint (20-100 RoIs).
- designed with a new real-time implementation while aggregating the boxes Π_i within the Stroke Model (SM).
- revisited with a full SM.

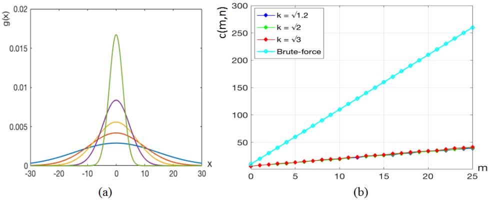


Figure: Within the SM (a) Gaussian distribution (b) aggregating Π_i

References I

- C.D. Nguyen, M. Delalandre, D. Conte and T.A. Pham. Performance Evaluation of Real-time and Scale-invariant LoG Operators for Text Detection. Conference on Computer Vision Theory and Applications (VISAPP), 2019.
- C.D. Nguyen. Real-time LoG-based Operators for Text Detection. PhD Thesis, Tours University, France, (in progress).