Structure, Concept and Result Reproducibility of the Benchmark on Vesselness Filters

Jonas Lamy, Bertrand Kerautret, Odyssée Merveille, Nicolas Passat

Structure, Concept and Result Reproducibility of the Benchmark on Vesselness Filters*

Jonas Lamy[0000−0002−0547−1341]1,
Bertrand Kerautret[0000−0001−8418−2558]1,
Odyssée Merveille[0000−0002−9918−3761]2, and
Nicolas Passat[0000−0002−0320−4581]3

1 Université Lyon 2, LIRIS (UMR 5205), Lyon, France
2 Univ Lyon, INSA-Lyon, Université Claude Bernard Lyon 1, UJM-Saint Etienne, CNRS, Inserm, CREATIS UMR 5220, France
3 Université de Reims Champagne Ardenne, CReSTIC, EA 3804, 51097 Reims, France

Abstract. This paper focuses on the structure and the concept of the framework used in the vesselness filters benchmark that was recently introduced. Vesselness filters are used to detect the presence of vessels in an image. There exists a wide variety of such filters and comparing their respective strengths and weaknesses is a non-trivial task, especially given the different contexts in which they are published. This benchmark was designed to ease such comparison process whereas remaining easy to customize. More specifically, this paper presents the benchmark structure and architecture. It also shows how to integrate new vesselness filters and/or new metrics in the benchmark with the requirements for future comparisons and online demonstrations.

Keywords: Benchmark · Vesselness filtering · CT Images · Replicability

1 Introduction

Vessel enhancement is an important step of the vessel segmentation process. Many vessel enhancement algorithms have been proposed over the last twenty years. However, the enhancement step is often overlooked, and very few filters are actually used in medical applications. Having a deeper look at these algorithms, one quickly realizes that it is hard to evaluate and compare them by relying on the associated literature. Indeed, most of them are tested on different (often private) datasets, which blurs the meaning of the filter scores across different papers. Based on these considerations, we decided to design a benchmark framework that allows for a comparison between vessel enhancement filters for 3D images.

In this paper, we first briefly recall the benchmark [7] (Section 2). Then, we propose a detailed description of the benchmark conception with a focus on how

* This work was funded by the French *Agence Nationale de la Recherche* (R-Vessel-X, grant ANR-18-CE45-0018).

to add new algorithms that fit the benchmark framework and how to add new metrics (Section 3). We also describe how to reproduce the results obtained in [7] (Section 4). Finally, an online demonstration is highlighted in Section 5 before conclusion.

2 Overview of the Benchmark of Vesselness Filters

The different algorithms used in the benchmark [7] are summarized in Table 1. The aim is to cover the main reference approaches starting from the pioneering ones with Sato [13] and Frangi [4] that exploit the Hessian matrix in a scale space analysis. These two approaches are able to take into account a certain amount of noise level and can carry out reconnection between vessel parts. Based on Hessian analysis, four more recent algorithms were considered in the benchmark: (i) the Meijering approach initially designed for neurite detection [11]; (ii) OOF which prevents response overflow from the scale space using a spherical framework equivalent to the Hessian matrix [8]; (iii) Jerman, that uses a volume ratio of tubular structures to better exploit the eigen values, producing a more consistent response; and (iv) Zhang that improves Jerman solution with a specialized preprocessing using a K-means classification combined with a sigmoid filter [14]. Finally, to cover other types of approaches, we integrated in the framework a method based on morphological filters that uses path opening and path-based structuring elements [12].

Method	Base	Main ideas	Date
Sato *et al.* [13]	Hessian	Vessel reconnection, noise control	1997
Frangi *et al.* [4]	Hessian	Blobs and plates removal with noise control	1998
Meijering *et al.* [11]	Hessian	Neurite detection	2004
OOF [8]	Hessian	Analysis restricted by a sphere	2010
Jerman [5]	Hessian	Volume ratio of tubular structures	2016
Zhang [14]	Hessian	K-mean with sigmoid using Jerman base	2018
RORPO [12]	Morphology	Vote on path opening	2018

Table 1: List of the methods currently available in the benchmark framework with their characteristics.

Main measures and metrics To evaluate the impact of the different algorithms, the responses of the filters are segmented by thresholding and compared with ground-truth. Then, the amounts of true positives, true negatives, false positives and false negatives are computed to define other metrics. In particular, we consider the Dice score that accounts for the overlap between the thresholded volume and the ground-truth, and the Matthew's Correlation Coefficients (MCC). The latter one has a similar purpose but also takes into account the true negatives, leveraging the metric for highly imbalanced datasets.

Evolutive structure As it will be described in the following sections, the proposed framework is generic enough to handle different types of images and filters. It can also be used to integrate other algorithms. We benchmarked the enhancement of liver vessels but any other kinds of structures and images can be considered. The only parts that need to be swapped in that case are the mask and reference images. The addition of new metrics is also possible in order to focus on other types of quality features.

Open framework with online demonstration The source code of the benchmark is available on a *GitHub* repository:

`https://github.com/JonasLamy/LiverVesselness`

The main organization of the benchmark is described hereafter and a direct access allows to test the different algorithms from an online demonstration that allows to upload specific data:

`https://kerautret.github.io/LiverVesselnessIPOLDemo`

From this work, the aim is to gather existing and future new algorithms in order to cover state of the art algorithms. In the sequel, we first show how to replicate the results and apply each filter using different data.

3 Filter Design and Integration

Since vessel segmentation is generally the final target application, the benchmark compares the thresholded output of a vesselness filter with a binary ground-truth. For each threshold value, several metrics are computed and aggregated in a CSV format. In medical applications, the area of interest is often an organ, for instance the liver in a CT scan of the torso. Our benchmark thus supports the use of masks to compute the metrics only in chosen/relevant areas.

The benchmark is implemented in C++ and the ITK library [6], which handles multiple medical images formats such as nifti, mhd, dicom series, etc.

3.1 Design of base usage

A vessel enhancement filter is designed to highlight the vessels in a 3D volume. This is often performed by improving the contrast of tubular structures whereas removing or decreasing the signal of the other structures and the background. In our benchmark framework, we wanted the filter implementations to be standalone programs so that they could be reused in other applications. Thus, a candidate filter should satisfy the following rules for a proper inclusion into the benchmark pool of vesselness filters:

– parameters should have `--input` for input option;
– parameters should have `--output` for output option;

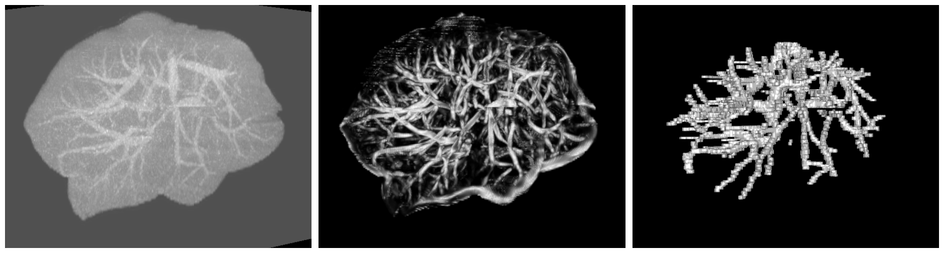


Fig. 1: Illustration of the Frangi algorithm (Antiga implementation) (b) applied in the masked liver (a), and compared to the ground-truth (c).

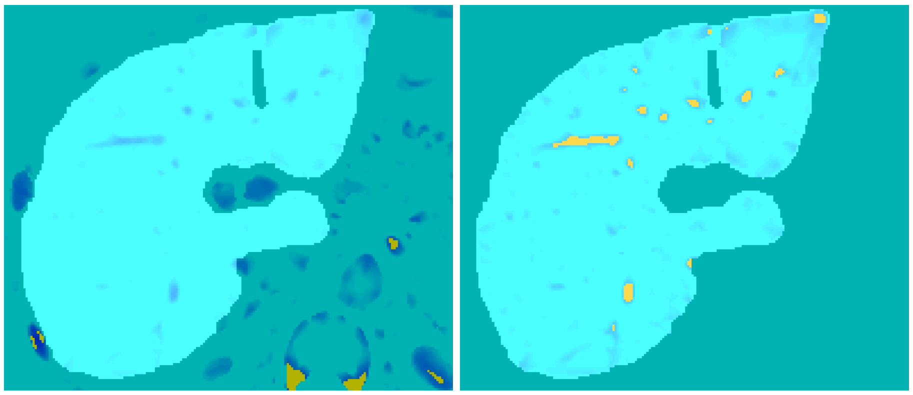
- a mask option `--mask` should be available where the filtered pixel values are set to zero where mask is zero and unaltered otherwise. We recommend to implement the masking as the final step of the filter, so that the masking does not generate phantom structures with high responses;
- the output of the enhancement filter should be normalized between $[0, 1]$;
- finally if *dicom* series are likely to be used, a `--inputIsDicom` option should also be available.

The CodeList 1 illustrates a commande line example defined to apply the algorithm Antiga on a sample image of the `Data` directory that contains the Ircad database (see links on the *GitHub* repository). The input sample, ground-truth and results are visible on Figure 1.

```
./Antiga --input ../../data/3Dircadb1.10/patientIso.nii --output antiga.nii
--mask ../../data/3Dircadb1.10/liverMaskIso.nii --sigmaMin 2.0 --sigmaMax
3.0. --nbSigmaSteps 3 --alpha 0.5 --beta 0.5 --gamma 5
```

Code List. 1: Command line example to apply Antiga algorithm (Fig. 1) from the build directory.

Providing a mask or a region of interest greatly modifies the results of some filters. For instance, Zhang filter uses a K-means-based enhancement specifically designed for the hepatic vessels; then it performs very well on images of the liver alone, but using a whole CT scan, it will shift the K-means intensity classes resulting in poor results, see Figure 2.



(a) no mask (b) with mask

Fig. 2: Zhang filtering results using same parameters with (b) and without masks (a). Color scale spreads from blue (low response) to yellow (high response).

```
{
    "Antiga" :
    [
        {
            "Output":"antiga1.nii",
            "Arguments":[
                {"sigmaMin":"2.0"},
                {"sigmaMax":"2.5"},
                {"nbSigmaSteps":"3"},
                {"alpha":"0.7"},
                {"beta":"0.1"},
                {"gamma":"5"}
                ]
        },
        {
            "Output":"antiga2.nii",
            "Arguments":[
                {"sigmaMin":"2.6"},
                {"sigmaMax":"2.3"},
                {"nbSigmaSteps":"3"},
                {"alpha":"0.5"},
                {"beta":"0.5"},
                {"gamma":"5"}
            ]
        },
    ],
    "Meijering" :
    {
        "Output":"meijering.nii",
        "Arguments":[
            {"alpha":"0.4"},
            {"sigmaMin":"1.6"},
            {"sigmaMax":"1.8"},
            {"nbSigmaSteps":"5"}
        ]
    }
}
```

Code List. 2: Examples of two parameter sets defined to run several instances of two different algorithms: two executions for the Antiga algorithm and another for the Meijering algorithm.

3.2 Design of a parameter set

The effectiveness of a filter in an experiment often depends on its parameterization. In our framework, each set of parameters is represented by a json object. An example of file containing several sets of parameters is illustrated on CodeList 2. A parameter object name should reflect the name of the vesselness filter program. It has two attributes: “output” (i.e. the name of the output of the filter), and “arguments” (a list of the filters arguments as they are defined in the program). The output naming convention is left to the user’s choice.

Here are some naming conventions we used. If the parameter set file is a mix of several filters, the name of the filter should be in the output volume naming scheme. For instance, the output filenames defined in the three parameter sets are prefixed with the algorithm name in CodeList 2. If the parameter set file is composed of the same filter with several variants of parameters, then the values of the moving parameters should be in the name for easier post-analysis.

The parameters are then used by the benchmark to call the corresponding command line to run the enhancement filter, so that the first parameter set will produce the following command line given in CodeList 3.

```
./Antiga --input inputVolume.nii --output antiga1.nii --sigmaMin 2.0
--sigmaMax 3.0 --alpha 0.7 --beta 0.1 --gamma 5
```

Code List. 3: First command line generated from the parameter set file of CodeList 2.

3.3 Database listing

The benchmark uses a database described by a listing text file. The listing file format should follow this pattern: name of the image instance (a.k.a. volume name / patient Id), path to the input volume, path to the binary ground-truth, path to the mask volume (ROI). At least one mask is required, but any arbitrary number of masks can be added.

```
3Dircadb1.10 // Name
PathToFolder/patientIso.nii //input image
PathToFolder/vesselsIso.nii  // groundtruth
PathToFolder/liverMaskIso.nii // first mask
PathToFolder/dilatedVesselsMaskIso.nii // second mask
```

Code List. 4: Example of database listing file.

CodeList 4 shows an example of database listing file. In this example, all the filters and their associated parameters are applied to `patientIso.nii` and compared to the ground-truth `vesselsIso.nii`. Metrics are computed in two areas: the mask of the liver, and the mask formed by the dilated vessels. All the associated resulting vesselness output volumes and csv files are stored in a folder named `3Dircad1.10`.

3.4 Benchmark parameters

Once the filter parameters and the database file are ready, the last step is to configure the benchmark. Once again, we chose a json file so that the tracking of carried out experiments is easier (see CodeList 5).

```
{
    "Settings":{
        "name":"MyBenchmark",
        "path":"PathToDirectory",
        "inputVolumesList":"fileLists/DatabaseFileList.txt",
        "algorithmSets":"paramSets/all_algorithms.json",
        "maskList":["Organ","Vessels"],
        "enhancementMask":"",
        "nbThresholds":200,
        "removeResultsVolumes":false
    }
}
```

Code List. 5: Benchmark parameters.

In addition to the location of the benchmark output directory and the location of the required files, the benchmark includes several options. The first one is the list of areas of interest `MaskList` where the metrics will be computed. The number of masks in that list should match the number of masks added to the database listing. The option `enhancementMask` allows the user to choose one of the above ROIs as a mask for the enhancement filter (effects demonstrated on Zhang filter on Figure 2 of Section 3.1). If the string is empty, then the metrics are computed on the whole input image. The number of thresholds (`nbThresholds`) allows to control the precision of the ROC curve. Finally, the benchmark is also designed for low disk memory usage with the option `removeResultsVolume`. If this option is set to true, only the resulting *csv* files will be kept and the vesselness filter outputs are removed as soon as the metrics are computed.

3.5 Extra metrics

The addition of extra metrics requires to modify the C++ code. The benchmark is composed of two classes: the Benchmark class which manages I/O and launches the scripts according to the parameter files, and the Eval class that computes the metrics for a given binary image and the associated ground-truth, or a confusion matrix.

Adding a new metric is rather simple. It requires to implement it in the Eval class and overload the << operator so that the results are included with the rest of the metrics in the csv file. One should not forget to add the name of the metric in the header of the csv in the benchmark.cpp file.

The metrics already available are:

- true positives, true negatives, false positives, false negatives;
- accuracy, sensitivity, precision, specificity;
- Dice, Matthew's Correlation Coefficients (MCC).

3.6 Results analysis

Since the outputs of the benchmark are *csv* files, the post-analysis can be done using tools such as *pandas*, *matlab* or any *csv* file reader. In the associated work [7], we were interested in measuring the most efficient filters when it comes to maximizing the mean MCC over the whole dataset. In other words, we aimed to determine the filter and parameter set that led to the best results in average, instead of seeking a per volume fine tuning.

4 Reproducibility of Benchmark Results

In this section, the focus is made on the reproducibility of the results presented in [7]. The reproducibility term follows the ACM definition: *"it consists of reproducing the results from a different research team by using the same experimental setup"* [3]. For this purpose of reproducibility, the requirements and main steps are presented in the following.

Requirements The dependencies to construct the benchmark programs are the *ITK* library [10], the *JsonCPP* library (a C++ Json parser) [2] and the *cmake* (3.10.2 [1]) build architecture. Note that to improve the reproducibility success probability, a *git submodule* is integrated in the main repository to link to external library. The post-analysis script requires *python3* with *matplotlib*, *pandas* and *numpy*. Note that a *virtualenv* based configuration is also provided for this script analysis step.

Experiment process The experiments for the benchmark described in [7] follow two main steps. These experiments are relatively complex and some manual analyses are required. In particular, we chose to find optimal parameters with a two steps strategy. First, using default intrinsic parameters, we searched the best scale parameter set that maximizes the mean MCC over the whole dataset. Once these optimal scale parameters were found, a second run was performed to find the optimal intrinsic parameters. For instance, if we consider the Frangi algorithm, it means looking first in a three dimensions scale space, and then in a two dimensional intrinsic parameter space instead of handling a five dimensional space as a whole.

Step 1: scale search. For a chosen method, the first step is carried out by launching a scale parameter search, which can be done for all samples of the databases in one command line call (see CodeList 6). Approximately 24 hours of computation are required to process a full database such as Ircad. However, it is possible to run several methods in parallel using a server with sufficient memory and computational power. At the end of this step, three *csv* files per method are produced, corresponding to each mask. Table 2 shows a sample of an aggregated result of this first step.

```
// To do for each method of the benchmark
./Benchmark -s scaleSearchIrcad<NameOfTheVesselnessFilter>.json
```

Code List. 6: Scale search command line.

Once the benchmark has been run for all the methods, a python script is in charge of summarizing the results in a pdf file. It gathers all the *csv* files in a folder and invokes `generatePDF.sh`. The produced pdf file will contain the top parameter set for each filters, and the top seven parameter sets per filters for each mask maximizing both MCC and Dice.

Table 2: Best scale parameter sets maximizing MCC.

	Ircad - Whole liver				Vascusynth - Whole volume			
Method	σ_{min}	σ_{max}	nb steps	Best MCC	σ_{min}	σ_{max}	nb steps	Best MCC
Sato *et al.* [13]	1.4	2.4	4	0.269 ± 0.065	1.4	2.8	4	0.541 ± 0.044
Frangi *et al.* [4]	1.4	3.0	4	0.344 ± 0.061	1.4	2.8	4	0.543 ± 0.040
OOF [8]	0.6	2.8	4	0.191 ± 0.039	0.6	1.6	4	0.382 ± 0.038
Meijering *et al.* [11]	1.2	2.2	4	0.138 ± 0.038	1.4	2.8	4	0.356 ± 0.040
Jerman *et al.* [5]	1.4	2.4	4	0.282 ± 0.063	1.4	2.6	4	0.612 ± 0.039
Zhang *et al.* [14]	1.4	2.4	4	0.344 ± 0.106	1.4	3.0	4	0.432 ± 0.040
Method	path size	factor	nb steps	Best MCC	path size	factor	nb steps	Best MCC
RORPO *et al.* [12]	60	1.2	3	0.384 ± 0.077	10	1.6	4	0.311 ± 0.032

Step 2: intrinsic parameter search Once the scale search is done, we perform an intrinsic parameter search with the fixed best scale parameters. The results are summarized in Table 3.

Finally, the filtering results are shown in Table 4 and illustrated in Figures 3 and 4.

5 Online Demonstration for Simple Custom Experiments

The different algorithms of the benchmark are available in the online demonstration mentioned in Section 2. The user can choose to apply a particular algorithm and change the default parameters in order to assess the behavior and stability of

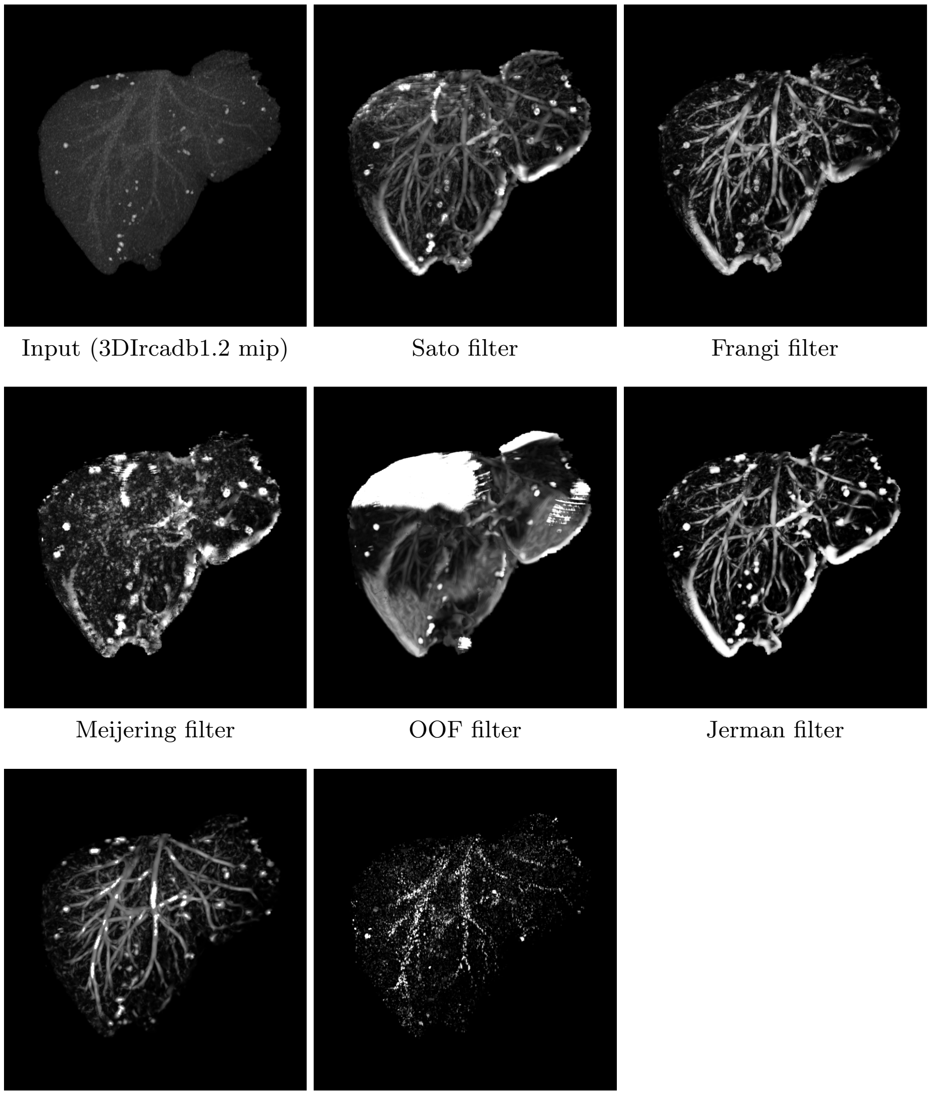


Fig. 3: Results on the sample 3DIrcadb1.2 obtained for the parameter sets obtaining the best mean MCC.

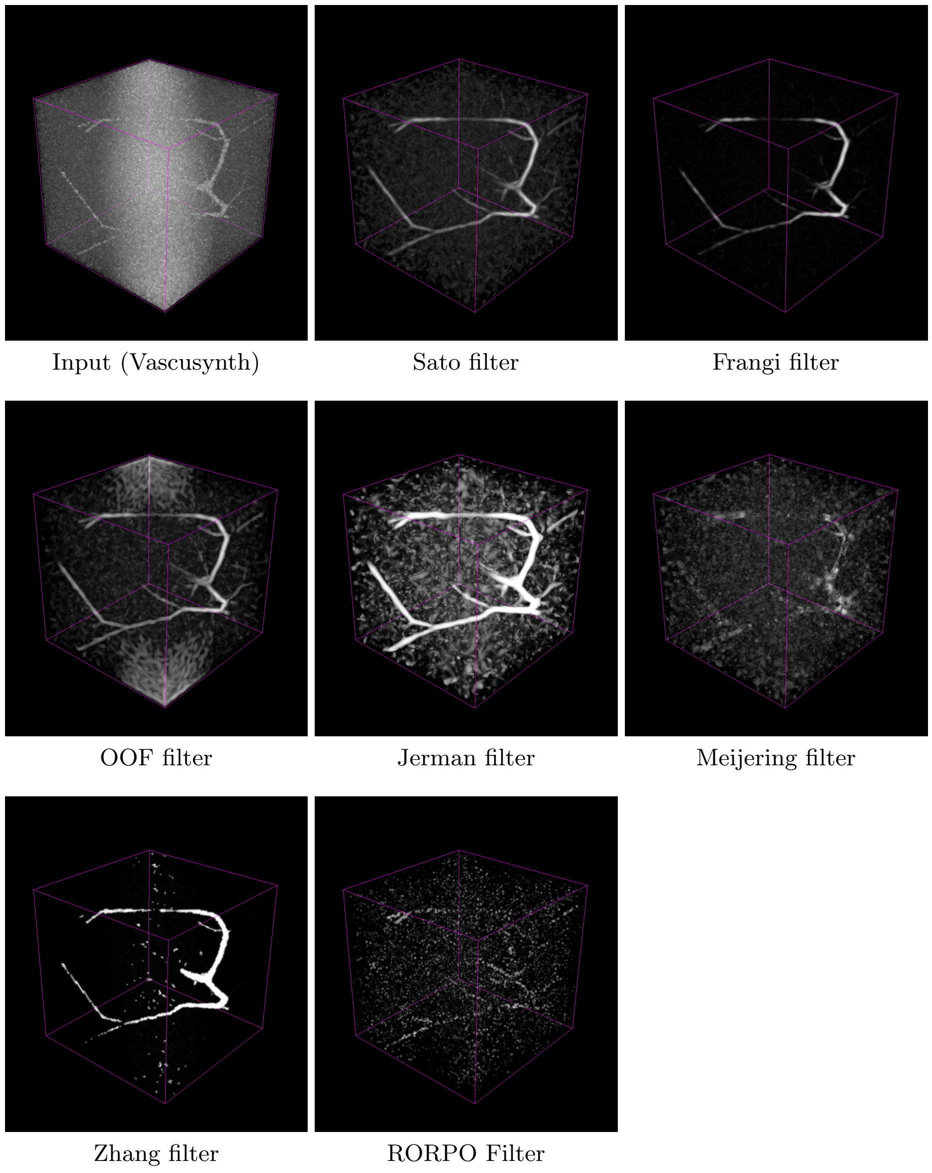


Fig. 4: Results on sample data 11 of group 4 obtained with the parameter set obtaining the best mean MCC.

Table 3: Best parameter sets maximizing MCC. Note that the RORPO and Meijering methods are not mentioned here since they do not have intrinsic parameters.

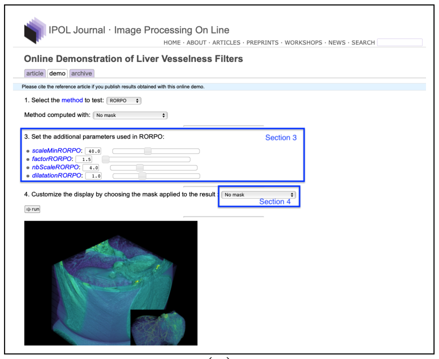
	Ircad - Whole liver			Vascusynth - Whole volume		
	α	β	MCC	α	β	MCC
Sato	0.3	1	0.275 ± 0.066	0.9	2.8	0.544 ± 0.043
	α	β	MCC	α	β	MCC
Frangi	0.6	0.4	0.356 ± 0.079	0.2	0.8	0.602 ± 0.042
	σ (smoothing)		MCC	σ (smoothing)		MCC
OOF	0.5		0.190 ± 0.041	0.5		0.343 ± 0.035
	τ		MCC	τ		MCC
Jerman	0.2		0.318 ± 0.081	0.8		0.612 ± 0.040
	τ		MCC	τ		MCC
Zhang	1.0		0.346 ± 0.106	0.6		0.478 ± 0.041

Table 4: Results sum up table.

	Best MCC	
	Ircad - Liver mask	Vascusynth - Whole volume
Sato	0.275 ± 0.066	0.544 ± 0.043
Frangi	0.356 ± 0.079	$\mathbf{0.602} \pm 0.042$
Meijering	0.138 ± 0.038	0.356 ± 0.040
Jerman	0.318 ± 0.081	0.612 ± 0.040
Zhang	0.346 ± 0.106	0.478 ± 0.041
OOF	0.190 ± 0.041	0.343 ± 0.035
RORPO	$\mathbf{0.384} \pm 0.077$	0.311 ± 0.032

the algorithm (see Figure 5 (a)). He/she can also choose to apply the filter on a restricted area of interest around a particular organ by selecting predefined mask images such as liver, vessel or bifurcation areas (see Figure 5 (a)). Moreover the user interface offers the possibility to upload new volume data and to check the response filter on any new images. In this case, the default mask images cannot be applied, but the user can choose to use his/her own custom mask before the image upload. Any kind of 3D volumetric images supported by ITK can be used such as *.vol*, *.nii*, *.mhd*, or *.mha* and the maximal size is fixed to 50 MB.

The demonstration provides complementary feedback for the user through the 3D display of the resulting response. The 3D viewer is the *itk-vtk-viewer* [9] that provides a 3D volume display with the ground-truth (when available). With this viewer, the user can focus on the areas of interest directly from the interaction with the online demonstration. Figure 6 illustrates the viewer embedded in a web browser. Thanks to this advanced viewer, the result of any user upload volume data can be displayed (Figure 6 (b)) and different display settings can


IPOL Journal · Image Processing On Line

HOME · ABOUT · ARTICLES · PREPRINTS · WORKSHOPS · NEWS · SEARCH

Online Demonstration of Liver Vesselness Filters

article demo archive

Please cite the reference article if you publish results obtained with this online demo.

1. Select the method to test: RORPO

Method computed with: No mask

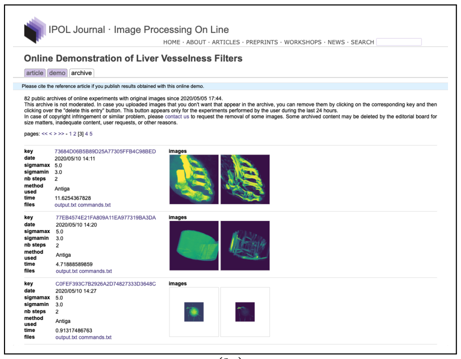
3. Set the additional parameters used in RORPO: Section 3

- scaleMinRORPO: 40.0
- factorRORPO: 1.5
- nbScaleRORPO: 4.0
- dilatationRORPO: 1.0

4. Customize the display by choosing the mask applied to the result : No mask Section 4

run

(a)


IPOL Journal · Image Processing On Line

HOME · ABOUT · ARTICLES · PREPRINTS · WORKSHOPS · NEWS · SEARCH

Online Demonstration of Liver Vesselness Filters

article demo archive

Please cite the reference article if you publish results obtained with this online demo.

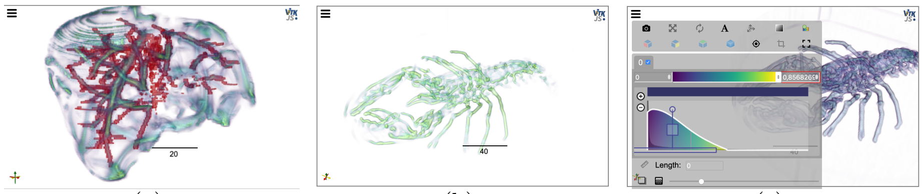
82 public archives of online experiments with original images since 2020/05/05 17:44.
This archive is not moderated. In case you uploaded images that you don't want that appear in the archive, you can remove them by clicking on the corresponding key and then clicking over the "delete this entry" button. This button appears only for the experiments performed by the user during the last 24 hours.
In case of copyright infringement or similar problem, please contact us to request the removal of some images. Some archived content may be deleted by the editorial board for size matters, inadequate content, user requests, or other reasons.

pages: << < > >> - 1 2 [3] 4 5

key	73684D06B5B89D25A77305FFB4C98BED	images
date	2020/05/10 14:11	
sigmamax	5.0	
sigmamin	3.0	
nb steps	2	
method used	Antiga	
time	11.6254367828	
files	output.txt commands.txt	
key	77EB4574E21FA809A11EA977319BA3DA	images
date	2020/05/10 14:20	
sigmamax	5.0	
sigmamin	3.0	
nb steps	2	
method used	Antiga	
time	4.71888589859	
files	output.txt commands.txt	
key	C0FEF393C7B2926A2D74827333D3648C	images
date	2020/05/10 14:27	
sigmamax	5.0	
sigmamin	3.0	
nb steps	2	
method used	Antiga	
time	0.91317486763	
files	output.txt commands.txt	

(b)

Fig. 5: Illustration of the online demonstration interface. (a) Main interface allowing to select and change the default parameters, including the intrinsic parameters and mask image (highlighted in blue). (b) Archive section of experiments given from user uploaded images.



(a) (b) (c)

Fig. 6: Illustration of the 3D display obtained from the online demonstration. (a) Result of a filter displayed with ground-truth (in light red). Experiments with user's data can also be carried out (b) and the viewer allows to set the contrast display (c).

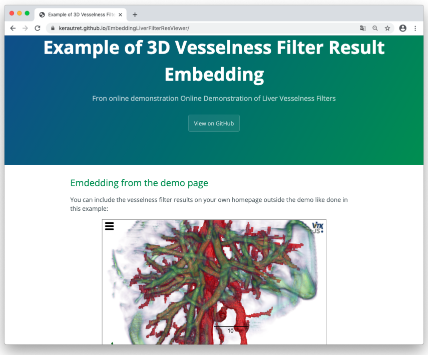
be adjusted such as the contrast and filter intensity scale (Figure 6 (c)), or the type of display by using 2D cutting planes.

The website interface also provides archives of the uploaded user's experiments (Figure 5 (b)). The access to the user's results is interesting to highlight the domain of interest and to show the global weaknesses and strengths of a particular algorithm. For now, the volumetric source images are stored on the server but depending on the user's upload usage, the result archive could be restricted in the future to the image previews of experiments with the parameters used.

Filter results embedding in other web pages The online structure of the demonstrations allows the user to export the 3D view of the filter results in other web pages by simply relying on few lines of HTML code. A typical example is illustrated on the following *GitHub* repository:

`https://kerautret.github.io/EmbeddingLiverFilterResViewer`

The main instructions needed to embed a filter result are described in the example page. They consist of copying few lines of code and updating two links (one for the filter result and one for the mesh reference). An overview of a result view embedding is illustrated in Figure 7. This behavior is possible thanks to *itk-vtk-viewer* and online demonstration archive coupled together. Such a feature can be useful to illustrate the performance of filter results in various conditions like in a research project web page and also for teaching activities.


Example of 3D Vesselness Filte
kerautret.github.io/EmbeddingLiverFilterResViewer/
View on GitHub
10

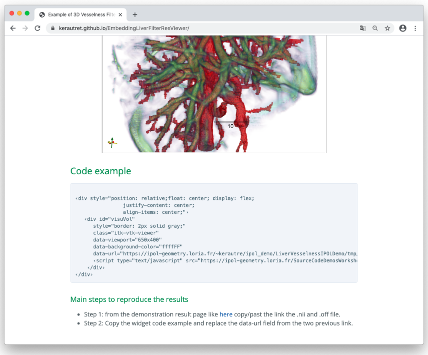
Example of 3D Vesselness Filter Result Embedding

Fron online demonstration Online Demonstration of Liver Vesselness Filters

Emdedding from the demo page

You can include the vesselness filter results on your own homepage outside the demo like done in this example:

(a)


Example of 3D Vesselness Filte
kerautret.github.io/EmbeddingLiverFilterResViewer/
10

Code example

```
<div style="position: relative;float: center; display: flex;
                justify-content: center;
                align-items: center;">
   <div id="visuVol"
      style="border: 2px solid gray;"
      class="itk-vtk-viewer"
      data-viewport="650x400"
      data-background-color="ffffFF"
      data-url="https://ipol-geometry.loria.fr/~kerautre/ipol_demo/LiverVesselnessIPOLDemo/tmp
      <script type="text/javascript" src="https://ipol-geometry.loria.fr/SourceCodeDemosWorksh
    </div>
</div>
```

Main steps to reproduce the results

- Step 1: from the demonstration result page like here copy/past the link the .nii and .off file.
- Step 2: Copy the widget code example and replace the data-url field from the two previous link.

(b)

Fig. 7: Example of 3D result embedding in other web pages. (a) Interactive 3D viewer embedded in a *GitHub* web page. (b) HTML container source code with main steps to construct the given page.

Online demonstration repository source Following the purpose to integrate other future reference methods, the source code on the online demonstration is available in the following repository:

`https://github.com/kerautret/LiverVesselnessIPOLDemo`

The main idea is to invite authors to propose their new filter algorithms. The integration in the online demonstration can be done in two main steps (see `Readme.md` file of the above repository). The first step is to address the new source code to the main benchmark repository. Then, the authors can request an issue to integrate their new methods with the description of specific parameters (or propose directly the demonstration template edition through a *GitHub Pull Request*).

6 Conclusion

In this paper, the architecture of a fully reproducible benchmark experiment was presented including benchmark set up, use of different masks and databases. We also provided an online demonstration to perform quick tests and visualization without any software installation. We also took special care to design the benchmark so that the addition of new filters would be very easy both in the benchmark structure and on the online demonstration. We highly encourage the community to contribute to the algorithm pool through the different *GitHub* repositories so that future state of the art algorithms could be compared with existing literature.

References

1. Cmake, `https://cmake.org`, (accessed October 14, 2020)
2. jsoncpp, `https://github.com/open-source-parsers/jsoncpp`, (accessed October 14, 2020)
3. Artifact review and badging (2020), `https://www.acm.org/publications/policies/artifact-review-badging`, Revised August 24, Accessed October 14
4. Frangi, A.F., Niessen, W.J., Vincken, K.L., Viergever, M.A.: Multiscale vessel enhancement filtering. In: MICCAI. pp. 130–137 (1998)
5. Jerman, T., Pernus, F., Likar, B., Spiclin, Z.: Enhancement of vascular structures in 3D and 2D angiographic images. IEEE Transactions on Medical Imaging **35**, 2107–2118 (2016)
6. Johnson, H.J., McCormick, M., Ibáñez, L., Consortium, T.I.S.: The ITK Software Guide. Kitware, Inc., 3rd edn. (2013), `http://www.itk.org/ItkSoftwareGuide.pdf`
7. Lamy, J., Merveille, O., Kerautret, B., Passat, N., Vacavant, A.: Vesselness filters: A survey with benchmarks applied to liver imaging. In: International Conference on Pattern Recognition (ICPR) (2020), `https://hal.archives-ouvertes.fr/hal-02544493`
8. Law, M.W.K., Chung, A.C.S.: Three dimensional curvilinear structure detection using optimally oriented flux. In: ECCV. pp. 368–382 (2008)
9. McCormick, M., Jourdain, S., Wittenburg, S., Smyth, D., Girault, A., Ouyang, W., Bilkey, J., Kerautret, B.: Kitware/itk-vtk-viewer: v10.8.0 (Oct 2020), `https://doi.org/10.5281/zenodo.4064952`
10. McCormick, M.M., Liu, X., Ibanez, L., Jomier, J., Marion, C.: Itk: enabling reproducible research and open science. Frontiers in Neuroinformatics **8**, 13 (2014)
11. Meijering, E., Jacob, M., Sarria, J.C., Steiner, P., Hirling, H., Unser, M.: Neurite tracing in fluorescence microscopy images using ridge filtering and graph searching: Principles and validation. In: ISBI. pp. 1219–1222 (2004)
12. Merveille, O., Talbot, H., Najman, L., Passat, N.: Curvilinear structure analysis by ranking the orientation responses of path operators. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**, 304–317 (2018)
13. Sato, Y., Nakajima, S., Atsumi, H., Koller, T., Gerig, G., Yoshida, S., Kikinis, R.: 3D multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. In: CVRMed-MRCAS. pp. 213–222 (1997)
14. Zhang, R., Zhou, Z., Wu, W., Lin, C.C., Tsui, P.H., Wu, S.: An improved fuzzy connectedness method for automatic three-dimensional liver vessel segmentation in CT images. Journal of Healthcare Engineering **2018**, 1–18 (2018)