4
Topology aware denoising

In this chapter our topology aware denoising technique will be detailed. This is a joint work with R. Nascimento (18). Instead of concerning ourselves with preserving the discontinuities of the vector fields as in the previous chapter, here we look at preserving singularities. There are four components in the methodology: generating the scale-space, detecting singularities in the scale-space, giving this information to the user, and finally reconstruct the desired vector field. A simple example to illustrate each component is presented in Section 4.1. Then the scale-space generation and reconstruction are presented in Section 4.2 and 4.3 respectively. At the end, in Section 4.4, the results are shown.

4.1
Methodology overview

The basic idea of our methodology is to let the user *locally* select the noise scale to remove, defining a scale parameter $s(x, y)$ at each point. We start by generating a scale-space from the original vector field and let the user choose a central scale s_0. In order to avoid the arduous task of defining the scale parameter $s(x, y)$ sample by sample, we display to the user the singularities that appear or disappear at different scales nearby s_0. When the user selects a topological change at a singular point (x_0, y_0), we define $s(x_0, y_0)$ to be the closest scale to s_0 that reverts the change. Finally, we return the reconstructed vector field as a smooth mixture of different scales of the scale-space.

Before entering in detail for each step, let's illustrate our technique on the example of Figure 4.1. This field contains some relatively clean parts at the bottom, and noisy parts at the top. The singular points at the bottom should be retained, almost all the singularities at the top should be cleaned, except for a sink that many streamlines point to.

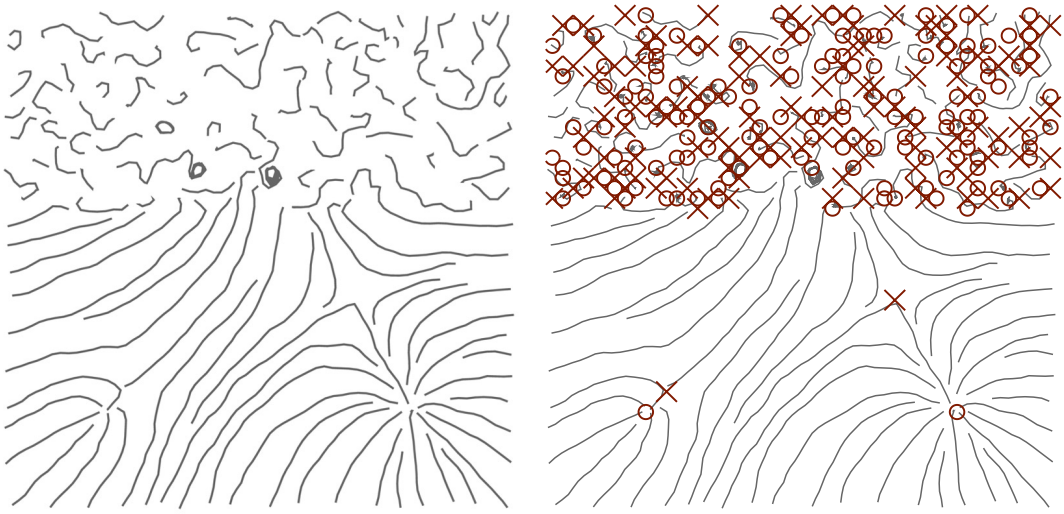

Figure 4.1: An artificial vector field represented by its streamlines (left) with its singularities marked (right).

Scale-space. In this example, we use a simple gaussian filter to generate a scale-space (see Figure 4.2). Our method can build on any denoising scale-space, as exemplified in Section 4.2 using isotropic or anisotropic filters.

Singularity detection. All the vector fields of the scale-space are available to the user at any time. We display the singularities of the field in each scale. There are different methods to detect singularities and our technique is independent of a specific choice of detection. As it can been seen in Figure 4.2, even though the field is still noisy at scale $s_0 = 10$, the meaningful singularity shown in the bottom left of Figure 4.1 was lost in the denoising process. The top part of the field is still noisy, needing more filtering.

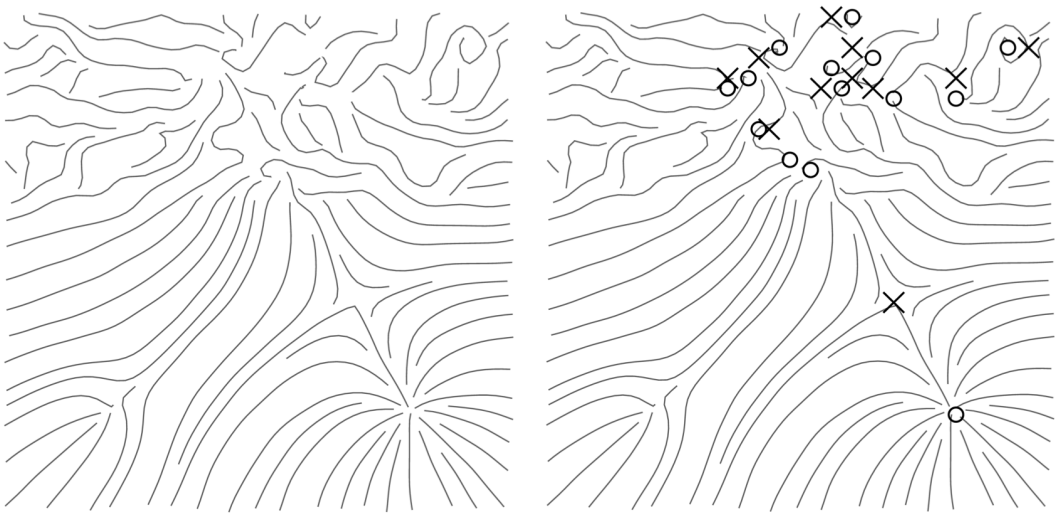

Figure 4.2: The vector fields at scale $s_0 = 10$ of its gaussian scale-space (left) with its singularities marked (right).

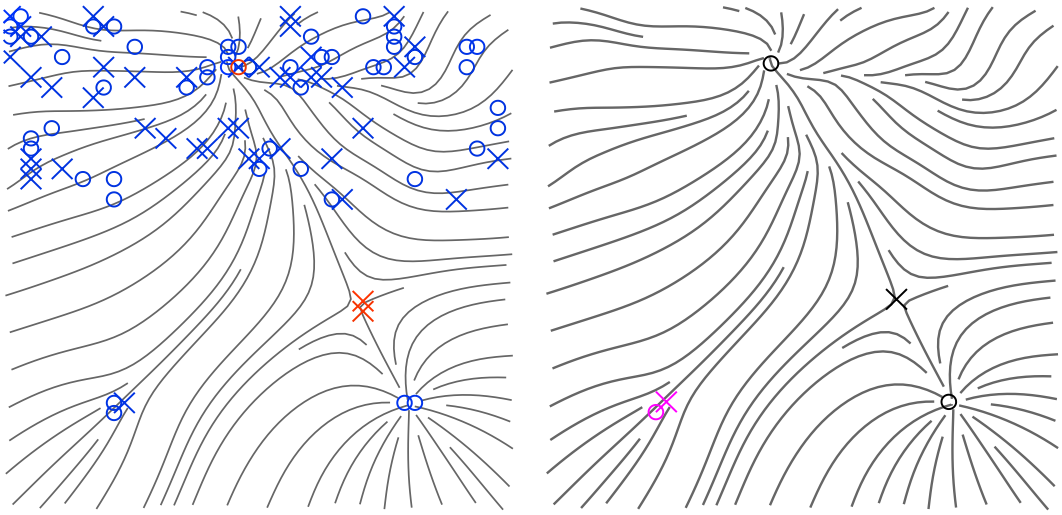

Figure 4.3: Our interface shows to the user the topological changes in nearby scales, here from 5 to 15 (left). The user then selects which topological changes he wants to revert (in purple on the right image).

Interface. In order to allow the user to denoise more of the top part while denoise less of the bottom to keep the meaningful singularity, we display to the user the topological changes at $s_0 = 10$ (see Figure 4.3). The user then selects which topological changes he wants to revert by a simple click.

Reconstruction. Each user selection defines a scale at the chosen point as the closest scale to $s_0 = 10$ that reverts the topological change. This gives a sparse sampling of the per-point scale parameter, which is smoothly interpolated to the whole domain. Our scheme supports different interpolations, and we provide two examples in Section 4.3. From this interpolation we can reconstruct an adaptively denoised vector field (see Figure 4.4).

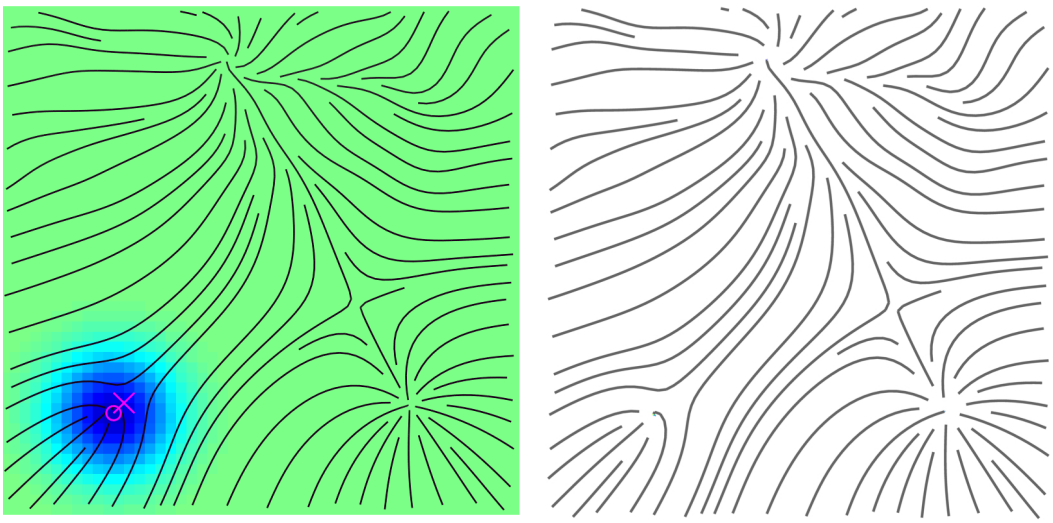

Figure 4.4: We finally interpolate the scales indicated by the user into a smooth function (left) which defines the reconstructed vector field (right).

4.2
Progressive filters and scale-space generation

The scale-space representation of the vector field is a collection of progressively denoised versions of the vector fields such as the one generated by the progressive method of Section 3.3. Each version is associated to an increasing scale parameter s. We denote $\bar{\mathbf{v}}(s, x, y)$ the vector value of the field at scale s and point (x, y). The fundamental example of a scale-space on continuous vector fields is the gaussian scale-space, obtained by convolving with a gaussian kernel of increasing variance: $G_\sigma(x, y) = \exp(-\frac{x^2+y^2}{2\sigma^2})$: $\bar{\mathbf{v}}(s, x, y) = \mathbf{v}(x, y) * G_s(x, y)$ (13).

In the discrete setting, this convolving approach fits into the more general framework of random walks from Chapter 3, which ensures nice scale-space properties from local convolution masks. The scale parameter is then the number of convolutions applied or the number of steps in the random walk. We exemplify our editing interface using two types of similarity functions to generate the space-scale: the gaussian G_σ and the feature preserving similarity function from the previous chapter:

$$A_{\sigma,\tau}(x, y, \mathbf{v}) = \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \exp\left(-\frac{||\mathbf{v}||^2}{2\tau^2}\right) ,$$

which takes into account the direction of the vector field and better preserves discontinuities. The scale-space is then directly generated by the repeated application of a 3×3 mask with the above kernels.

4.3
Reconstruction

The singularities selected by the user provides a sampling of the scale function $s(x, y)$ on the domain. To reconstruct the whole vector field, we interpolate this sampling. Then denoting $\bar{\mathbf{v}}_{i,j}(s)$ the vector field sample at scale s, we define the reconstructed vector field $\tilde{\mathbf{v}}$ at grid point (x_i, y_j) by:

$$\tilde{\mathbf{v}}_{i,j} = \bar{\mathbf{v}}_{i,j}(s(x_i, y_j)) .$$

Virtually any interpolation scheme works, although with different resulting qualities. If the interpolation is not smooth enough, the rapid changes in the scale parameter may create artifacts in the reconstructed field. Moreover, the interpolation must maintain the scale in a neighborhood of the singularity to preserve it. We implemented two methods for the interpolation of s that gave satisfactory results: radial basis functions (RBF), with gaussian basis,

and kernel Shepard interpolation (2) with gaussian kernel.

The RBF interpolation of $s(x, y)$ from the scales of the used selected singularities s_k at (x_k, y_k) is obtained by a least-squares minimization on the coefficients α_k of

$$\min_{\{\alpha_k\}} \sum_k \|s_{rbf}(x_k, y_k) - s_k\|^2 \ , \quad where \tag{4-1}$$

$$s_{rbf}(x, y) = \sum_k \alpha_k G_\sigma \left(x - x_k, y - y_k\right) \ . \tag{4-2}$$

The kernel Shepard method modifies the original Shepard interpolation (26) by using kernels instead of the Euclidean distance:

$$s_{ks}(x, y) = \frac{1}{\sum_k G_\sigma(x-x_k, y-y_k)} \cdot \sum_k G_\sigma(x-x_k, y-y_k) \cdot s_k \ .$$

A important property of this method is that the image is limited to $[\min_k s_k, \max_k s_k]$.

4.4 Results

In this section we present our experimental results on synthetic, simulated and measured vector fields. Since we work with relatively small 2D vector fields stored in regular lattices compared to the computing power of actual hardware, the interface responds in real-time to user interactions, except for the initial scale-space generation (see Table 4.1). In all the experiments presented here, the singularities detected by the winding number method and the bilinear one were the same, although they may differ in very particular cases.

Table 4.1: Timings, in milliseconds, for each step of the edition.

Data	**Fig**	**Size**	**Filter**		**Singularity**		**Scale**	**Reconstruction**		
			type	*(ms)*	*type*	*(ms)*	**select**	*type*	*solve*	*eval*
Analytic	7	2500	G_σ	18.9	w_Γ	98.0	7.3	KS		0.1
Granular	8	2500	$A_{\sigma,\tau}$	587.5	w_Γ	110.8	8.3	RBF	0.8	0.9
PIV 1	1	15624	G_σ	135.0	$\mathbf{b}=0$	947.6	65.6	RBF	0.1	7.6

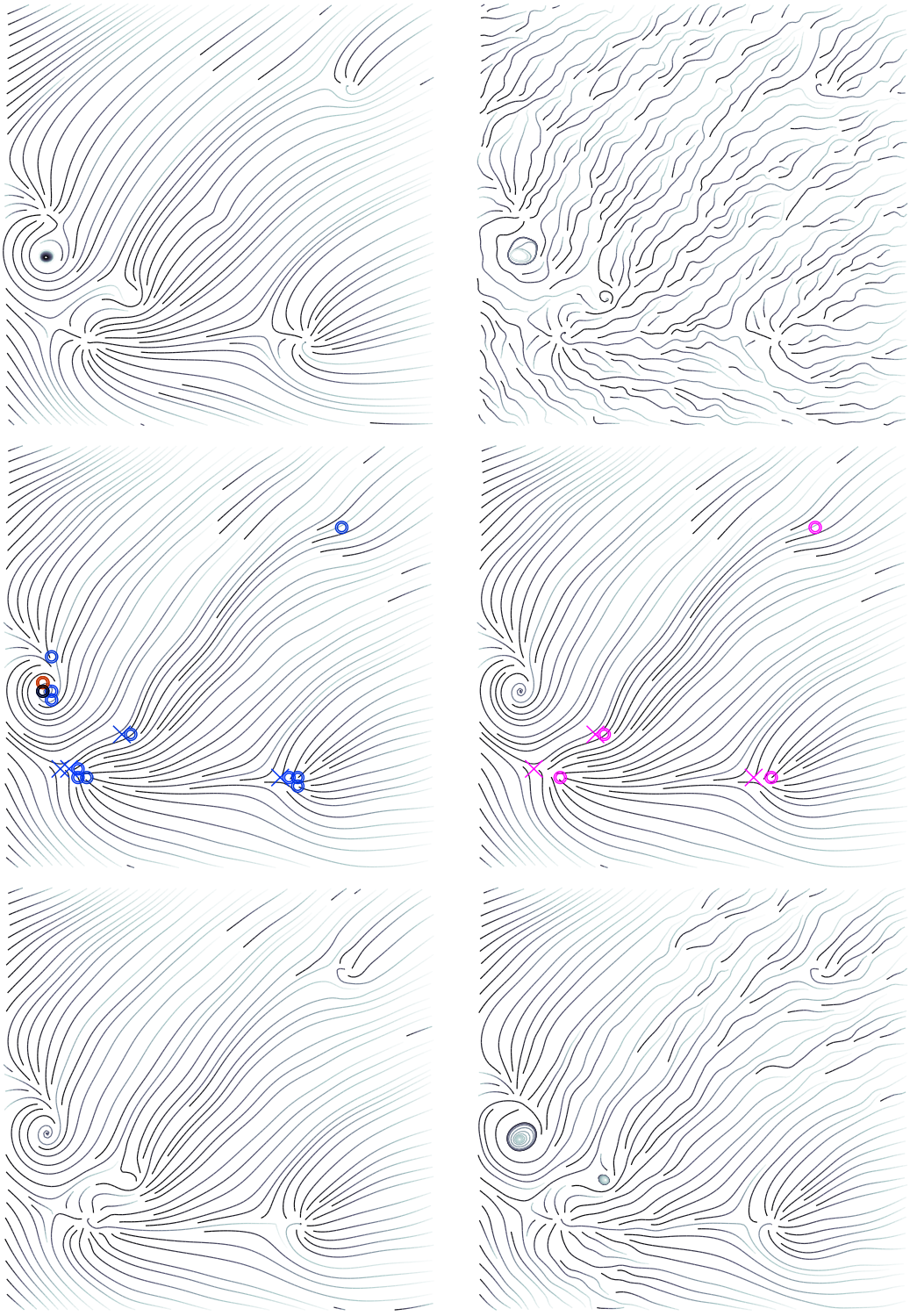

Figure 4.5: Experiments on an analytic vector field (top left) artificially corrupted by non-gaussian noise (top right). The user can choose between singularities that disappeared before scale s_0 (in blue) or singularities that could be smoothed out at scale $s > s_0$ (in red) (middle left). From the user selection (middle right), we reconstruct the vector field maintaining the selected scale in a small (bottom left) or larger radius (bottom right).

Synthetic data. We first validate our approach on a synthetic vector field, corrupted by an artificial, non-gaussian noise (see Figure 4.5). We can denoise adaptively the vector field, recovering the original singularities. We use gaussian scale-space with a kernel Shepard interpolation. Observe that, varying the σ of the kernel used in the reconstruction, we can carry larger portion of the fields at the selected scale.

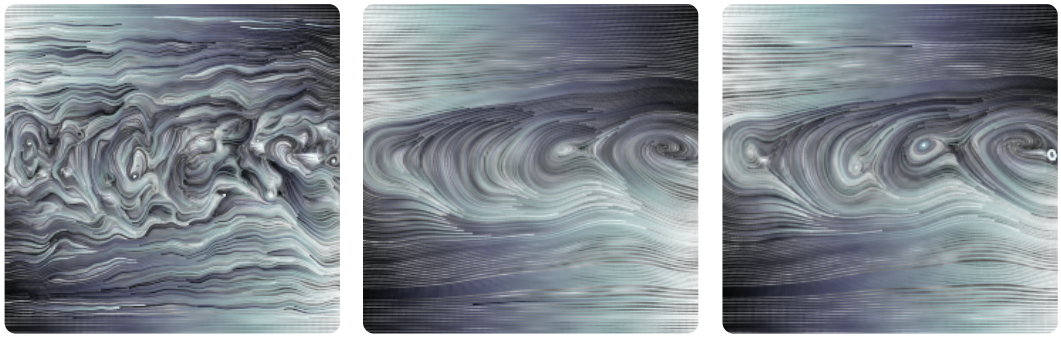

Figure 4.6: On a vector field from a simulated shear band granular system (left) 40 steps of denoising recovers the granular bands but loses one of the main vortices (middle). Selecting that vortex in our interface allow for a denoised reconstruction with the main singularities (right).

Simulation data. We then experimented on a vector field of 2500 samples issued by a granular simulation (3). The shearing of the granular system generates five main vortices between the shear bands, which are clearly visible in Figure 4.6 besides the noise. We use an anisotropic filter to generate the scale-space, requiring around $s = 40$ steps to denoise the granular bands at the top and bottom. However, this smoothens out one of the main vortices. Selecting it in our interface allows to reconstruct a clean vector field with the main singularities, using here the RBF interpolation.

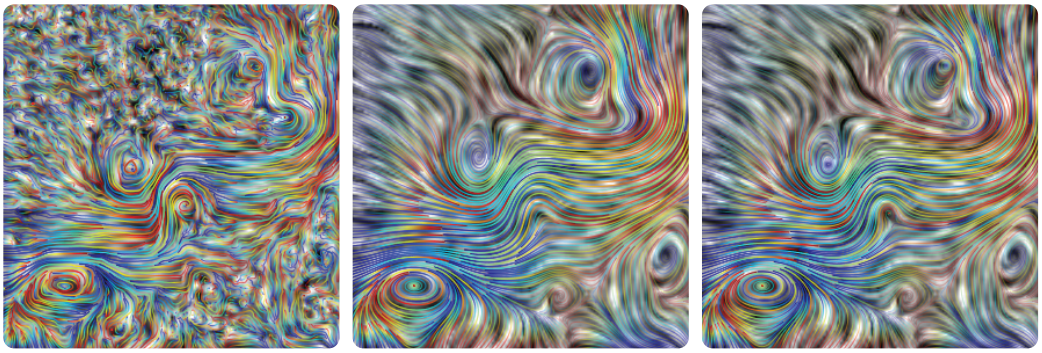

Figure 4.7: Topology-aware denoising of a measured fluid velocity field: (left) original field, (middle) gaussian denoising, (right) gaussian denoising preserving topological singularities selected through our interface.

Measured data. We finally experimented our method on real measured vector field of 15624 samples, acquired through PIV imaging. The experiments of Figure 4.7 and Figure 4.8 are measured from a wall-jet setup, where water is injected from the left of the image and kicks on the wall on the right. The images correspond to the top half of the jet. The water injection is stronger in the experiment of Figure 4.7 as compared to the one of Figure 4.8. In both cases, the top left part of the image is very noisy since there is less water, while the right part is turbulent. This leads to several important singularities on the right part of the field to disappear before the singularities caused by the noise. In the reconstructed vector field, those singularities are recovered. We used a gaussian scale-space for this experiment. While denoising this PIV data set with the Random Walk filter in Chapter 3, we saw how tricky it was since the information on the right side was considered noise and remove before the true noise on the left. Here, with our proposed methodology, we were able to get around the problem by selecting the right scales and reconstructing the desired field.

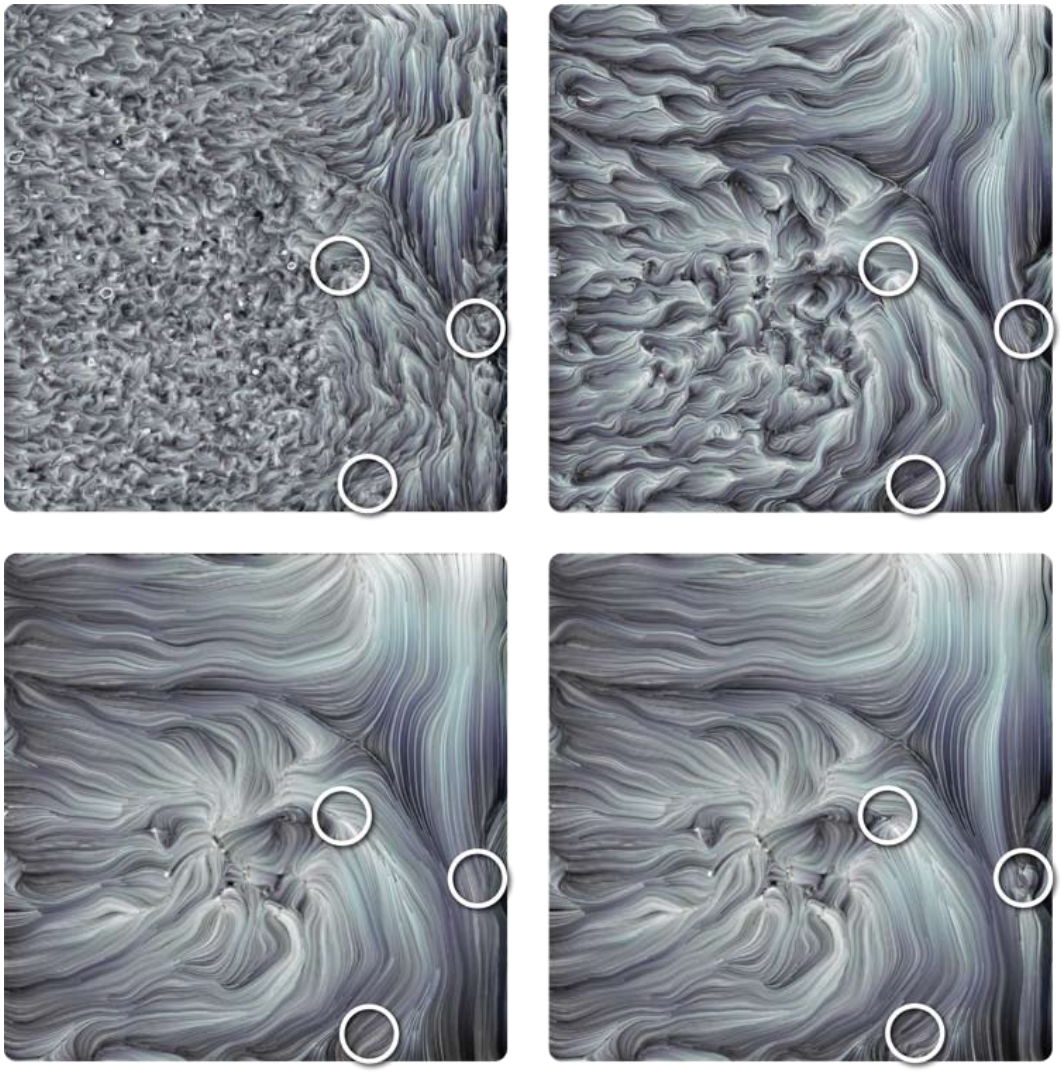

Figure 4.8: Denoising a vector field (top left) measured by a PIV device on a wall-jet experiment: the scale-space at steps 30 and 100 (middle images) smoothens out the important singularities, at the right part of the image, while keeping some singularities related to the noise at the left part of the image. Selecting singularities at the right of the image better recovers the behavior of the fluid (bottom right).