Spatiotemporal Extension of color Decomposition model and Dynamic Color structure-Texture extraction

Lugiez Mathieu, Dubois Sloven, Ménard Michel
L3i, 17042 La Rochelle, FRANCE

El Hamidi Abdallah
MIA, 17042 La Rochelle, FRANCE

Fédération PRIDES: Pôle régional de Recherche en Images, Données et Systèmes

Abstract

A new issue in texture analysis is its extension to temporal domain, a field known as Dynamic Texture analysis. Dynamic, or temporal, texture is a spatially repetitive, time-varying visual pattern that forms an image sequence with a certain temporal stationarity. Following recent work, color image decomposition into geometrical, texture and a noise components appears as a good way to extracting meaningful information, i.e texture component, independently of noise and geometrical information.

In this way, we propose to extend spatial color decomposition model to spatiotemporal domain, and attempt to separate static texture present in video and real dynamic texture. To our best knowledge, no such time adaptation is currently available.

Introduction

Motivation

A new issue in texture analysis is its extension to temporal domain, a field known as Dynamic Texture analysis. Dynamic, or temporal, texture is a spatially repetitive, time-varying visual pattern that forms an image sequence with a certain temporal stationarity. In Dynamic Texture, the notion of self-similarity central to conventional image textures is extended to spatiotemporal aspect.

Dynamic textures are typically result from processes such as water flows, smoke, fire, a flag blowing in the wind, a moving escalator, or a walking crowd. Important tasks are thus detection, segmentation and perceptual characterization of dynamic textures. The ultimate goal is to be able to support video queries based on the recognition of the actual natural and artificial dynamic texture processes.

Following recent work, color image decomposition into geometrical, texture and a noise components appears as a good way to reach this aim in extracting meaningful information, i.e texture component, independently of noise and geometrical information. In this way, we propose to extend spatial color decomposition model to spatiotemporal domain, and attempt to separate static texture present in video and real dynamic texture. To our best knowledge, no such time adaptation is currently available.

Overview of the paper

The aim of this work is to extend a model, which decompose color image into three components, a first one containing geometrical structure: U, a second V, holding the textural information and the last one containing the noise: W, to spatiotemporal domain. So, we aim to deal with color image sequences in extending to time existing reliable model. Moreover, in the decomposition texture component, we attempt to determinate spatial texture from texture showing a real dynamicity, which will be suit for future work on dynamic texture.

In the first place of this paper we introduce the extended minimization functional problem and the associate discrete framework in which we place ourself and which is an appropriate one in image sequence processing. In a second part we present the extended time decomposition model and its grayscale implementation. Then in the third part, we will be examining the various challenges arising from the introduction of color model, our proposition to solve this problem and its numerical implementation. Finally, in the last part, we present two way to discriminate static from dynamic textures in image sequence. We will present too, choice and influence of parameters and show and discuss some significant results.

Time extension of decomposition model

Spatiotemporal structure

In order to decompose image sequences in suitable components we propose to extend the Aujol-Chambolle [2] decomposition model. Its rely on dual norms derived from BV^1,G^2 and E^3spaces. Authors propose to minimize the following discretized functional:

$$\inf_{(u,v,w)\in X^3} F(u,v,w) = \underbrace{J(u)}_{\substack{\text{Regularization}\\ \text{TV}}} + \underbrace{J^*\left(\frac{v}{\mu}\right)}_{\substack{\text{Texture}\\ \text{extraction}}} + \underbrace{B^*\left(\frac{w}{\delta}\right)}_{\substack{\text{Noise extraction}\\ \text{by shrinkage}}} + \underbrace{\frac{1}{2\lambda}\|f-u-v-w\|_X^2}_{\text{Residual part}} \tag{1}$$

where X is the Euclidian space $\mathbb{R}^{N\times N}$.

To take into account the spatiotemporal structure, we consider a video as an 3-D image [1], i.e a volume, so that we can apply 2-D image algorithms extended to the 3-D case. We assume that we have a given image sequence $f \in L^2(\Omega)$, where Ω is an open and bounded domain on $\mathbb{R}^3$, with Lipschitz boundary. In order to recover u, v, w from f, we propose:

- An extended total variation definition:

$$\int_t \int_\Omega \left|\nabla_{xyt} u\right| dx\, dy\, dt \tag{2}$$

  where $\nabla_{xyt} u$ the spatiotemporal gradient of u.

[1] $BV(\Omega)$ is the subspace functions $u \in L^1(\Omega)$ such that the following quantity, called the total variation of u, is finite:

$$J(u) = sup\left\{\int_\Omega u(x)div(\xi(x))dx\right\}$$

such that $\xi \in C_c^1(\Omega,\mathbb{R}^2), ||\xi||_{L^\infty(\Omega)} \leq 1$

[2] G is the subspace introduced by Y.Meyer for oscillating patterns.

[3] E is a dual space to model oscillating patterns: $E = \dot{B}^{\infty}_{-1,\infty}$ dual space of $\dot{B}^{1}_{1,1}$

- A new definition of G extended to a third dimension: G is the Banach space composed of the distributions f which can be written $f = \partial_1 g_1 + \partial_2 g_2 + \partial_3 g_3 = div_{xt}(g)$ with g_1, g_2 and g_3 in $L^\infty(\Omega)$. On G, the following norm is associate:

$$\|f\|_G = inf\{\|g\|_{L^\infty(\Omega,\mathbb{R}^3)}; f = div(g)\} \tag{3}$$

Discretization

From now on, we consider the discrete case. We take here the same notation as in [2] and we present the Total Variation discretization. Let ∇u the gradient vector given by:

$$(\nabla u)_{i,j,k} = \left((\nabla u)^1_{i,j,k}, (\nabla u)^2_{i,j,k}, (\nabla u)^3_{i,j,k}\right) \tag{4}$$

$$(\nabla u)^1_{i,j,k} = \begin{cases} u_{i+1,j,k} - u_{i,j,k} & \text{if } i < N \\ 0 & \text{if } i = N \end{cases}$$

$$(\nabla u)^2_{i,j,k} = \begin{cases} u_{i,j+1,k} - u_{i,j,k} & \text{if } j < N \\ 0 & \text{if } j = N \end{cases}$$

$$(\nabla u)^3_{i,j,k} = \begin{cases} u_{i,j,k} - u_{i,j,k-1} & \text{if } k < N \\ 0 & \text{if } k = N \end{cases}$$

The discrete TV of u is given by:

$$J(u) = (\nabla u)^1_{i,j,k} + (\nabla u)^2_{i,j,k} + c(\nabla u)^3_{i,j,k} \tag{5}$$

we introduce the c constant to maintain homogeneity between space and time component. It's mainly for numerical implementation, to avoid discretization problem due to quantization step, which be different along space and time dimension. In practice, we often set it to one, but user can adapt it to less, more or in function of frame per second, or quickness of movement present in sequence, to ensure most reliability and homogeneity.

Spatiotemporal grayscale decomposition

The Chambolle's projection algorithm [4] is a smart way to numerically solve the different minimization problems induced by the functional (1), using fixed point method: $p^0 = 0$, and

$$p^{n+1}_{i,j,k} = \frac{p^n_{i,j,k} + \tau(\nabla(div(p^n) - f/\lambda))_{i,j,k}}{1 + \tau\left|(\nabla(div(p^n) - f/\lambda))_{i,j,k}\right|} \tag{6}$$

As shown in [4] if τ is small enough, that ensure the convergence of the algorithm.

So, to solve (1) authors propose to solve successively three different problems:

- v and w fixed:

$$\underset{u \in X}{inf} \left(\frac{1}{2\lambda}\|f - u - v - w\|^2_X + J(u)\right) \tag{7}$$

$$\tilde{u} = f - u - v - w - P_{G_\lambda}(f - v - w) \tag{8}$$

- u and w fixed:

$$\underset{v \in G_\mu}{inf} \|f - u - v - w\|^2_X \tag{9}$$

$$\tilde{v} = P_{G_\mu}(f - u - w) \tag{10}$$

- u and v fixed:

$$\underset{w \in \delta B_E}{inf} \|f - u - v - w\|^2_X \tag{11}$$

$$\tilde{w} = P_{\delta B_E}(f - u - v) \tag{12}$$

$$= f - u - v - W_{ST}(f - u - v, \theta) \tag{13}$$

where W_{ST} stands for the wavelet soft-thresholding, extended to time, with non linear diffusion equations [8], of $f - u - v$ with threshold θ:

$$S_\theta(w_i) = w_i \left(1 - 13\theta\left(\sqrt{w_x^2 + w_y^2 + w_t^2 + 2w_{xy}^2 + 2w_{xt}^2 + 2w_{yt}^2 + 4w_{xyt}^2}\right)^{-1}\right)$$

$$\text{if } S_\theta(w_i) \geq (13\theta), 0 \text{ otherwise.} \tag{14}$$

Figure 1 present our grayscale decomposition. We can distinctly see time influence, reed's branch oscillating under water flow is clearly highlight. Moreover, waves present in basin's fountain are well regularized in U component, water dynamicity is totaly catch as texture.

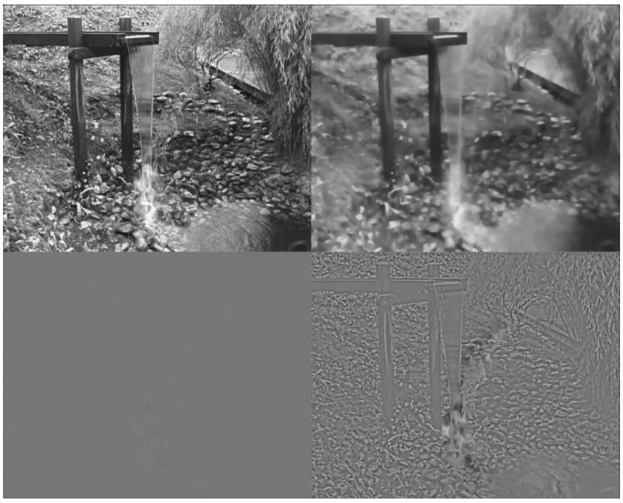

Figure 1. Spatiotemporal grayscale decomposition: Top left: f the original image from sequence, top right: U geometrical component, bottom left: W noise component, bottom right: V + 127 texture component.

Reader can report to Figure 5 and 6 to have comparison and difference between classic color decomposition and its spatiotemporal extension. Difference between two successive images of U, statically decomposed, presents about four times less information than our time extended decomposition.

Spatiotemporal color decomposition

In order to solve total variation minimization of color image sequence, we adapt the solution of Aujol and Ha Kang [3], to time. In fact, Chambolle's projection is not suitable to deal with color image sequences, due to its single plan limitation. To avoid the problem of regularization, authors use classic TV minimizing functional, solving Euler-Lagrange, into Chromaticity and Brightness (CB) and don't extract noise component. To realize their numerical implementation, they use digital TV filter, based on work of Chan, Osher and Chen [5].

Digital TV filter implementation

In order to adapt the solution of [3] to spatiotemporal aspect we readapt their solution to temporal aspect. We reformulate the energy functional of [5], using spatio-temporal gradient and extending neighborhood graph to time neighbors (as seen in Figure 2).

Given a noisy image's sequence u^0, we redefine energy fonctional (presented in [5]), adapted to spatiotemporal gradient formulation as (with λ, the Lagrange multiplier):

$$J(u)=\int_t\int_\Omega \left|\nabla_{xyt}u\right| dxdydt+\frac{\lambda}{2}\int_t\int_\Omega (u-u^0)^2 dxdydt \tag{15}$$

The data that need to be regularized are assumed to be living on a graph. A general digital domain is modeled by a graph $[\Omega,E]$, with a finite set Ω of nodes and an edge dictionary E. If α and β are linked by an edge, whether spatially or temporally, we write $\alpha\sim\beta_{st}$. A digital scalar signal u is a function on Ω, $u:\Omega\rightarrow\mathbb{R}$. The value at node α is denoted by u_α and local variation at any node is defined as $|\nabla_\alpha u| := \sqrt{\sum_{\beta_{st}\sim\alpha}(u_{\beta_{st}}-u_\alpha)^2}$, and the regularized local variation, in its conditioned form (to avoid singularity for $|\nabla u|$ in denominator of associated Euler-Lagrange equation), for any positive number ε is:

$$|\nabla_\alpha u|_\varepsilon=\sqrt{|\nabla_\alpha u|^2+\varepsilon^2} \tag{16}$$

So, for a given noisy spatiotemporal signal u^0, the digital TV filter, $\mathscr{F}_\alpha^{\varepsilon,\lambda}$, is defined as:

$$\mathscr{F}_\alpha^{\varepsilon,\lambda}\left(u,u^0\right)=\sum_{\beta_{st}\sim\alpha} h_{\alpha\beta_{st}}(u)u_{\beta_{st}}+h_{\alpha\alpha}(u)u^0_\alpha \tag{17}$$

where the low-pass coefficients filters are given by:

$$h_{\alpha\beta}(u)=\frac{w_{\alpha\beta}(u)}{\lambda+\sum_{\gamma\sim\alpha}w_{\alpha\gamma}(u)}$$
$$h_{\alpha\alpha}(u)=\frac{\lambda}{\lambda+\sum_{\gamma\sim\alpha}w_{\alpha\gamma}(u)}$$
$$\text{with } w_{\alpha\beta}(u)=\frac{1}{\sqrt{|\nabla_\alpha u|^2+\varepsilon^2}}+\frac{1}{\sqrt{|\nabla_\beta u|^2+\varepsilon^2}}$$

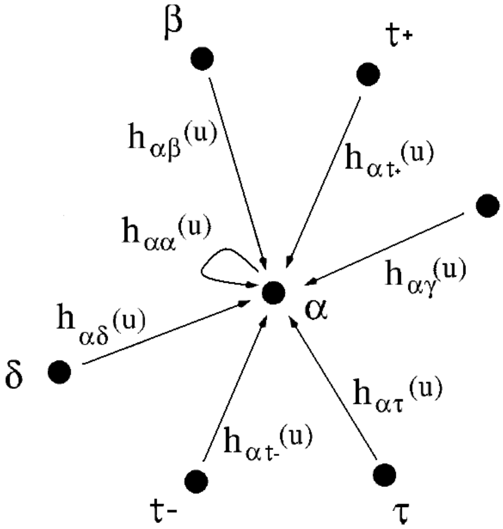

Figure 2. Digital TV filter at node α. The β, δ, τ and γ are α's space neighbors and $t+$ and $t-$ are α's time neighbors. Each arrow means that the u value at the tail node is multiplied by the filter coefficient beside and added to α. The exception is the loop arrow at α, for which one uses the original un-regularized data u^0, instead of the u value.

Color decomposition algorithm

We present the algorithm to decompose color image sequences in two components, u and v.

```
(1) Initialization of $f,u,v$ where $f_0$ is the original sequence
        $f=f_0,\quad u^0=f_0 \quad$ et $v^0=0$

(2) Iterate $m$ times
    (a) Separate $f,u$ and $v$ to Brightness $(f_b,u_b,v_b)$ and
        Chromaticity $(f_c,u_c,v_c)$ components
            $f_b=||f|| \qquad f_c=\frac{f}{||f||}$
            $u_b^n=||u_n|| \qquad u_c^n=\frac{u_n}{||u_n||}$
            $v_b^n=||v_n|| \qquad v_c^n=\frac{v_n}{||v_n||}$

    (b) Iterate $n$ times for update $u_c$ and $u_b$
            $u_c^{n+1}=\mathscr{F}_\alpha^{\varepsilon,\lambda_c}\left(u_c^n,f_c-v_c^n\right)$
            $u_b^{n+1}=\mathscr{F}_\alpha^{\varepsilon,\lambda_b}\left(u_b^n,f_b-v_b^n\right)$
            $u_c^n=u_c^{n+1}$ et $u_b^n=u_b^{n+1}$

    (c) Update $u$ and calculate the residual $r$
        $u^{n+1}=u_c^{n+1}*u_b^{n+1}$
        $r^n=f-u^{n+1}-v^n$

    (d) Iterate $n$ times for update $r$
            $r^{n+1}=\mathscr{F}_\alpha^{\varepsilon,\mu}\left(r^n,f-u^{n+1}-v^n\right)$
            $r^n=r^{n+1}$

    (e) Update $v$
        $v^{n+1}=f-u^{n+1}-r^{n+1}$

    (f) Preparation for the next iteration
        $u^n=u^{n+1}$
        $v^n=v^{n+1}$
```

Influence of parameters and numerical results

All images and results are compute from DynTex, the dynamic texture database [7] which provide a large and diverse database of high-quality dynamic textures. Dyntex sequences come from natural scene presenting a wide variety of moving process as flowing water, leaves blowing in wind, walking crowd... Such diversity grants user to identify and emphasize a lot of aspects in testing purpose.

Influence of parameters

The parameter which defined the wide of oscillations captured in texture is μ. It's represent, in some sense, the detail level of our decomposition in space and time. Parameters λ_b and λ_c in filter process control intensity of regularization process, they represent, in some sense, the scale of regularization.

To obtain week regularization, we use parameters as: μ between 0.5 and 1, λ_b and λ_c near to 1.

For classic parameters, which work well for most of dynamic sequence aspect, we set μ at 0.01, λ_b at 0.04 and λ_c at 0.01, we compute three total iterations of our algorithm and ten loops for each call of filtering.

For strong regularization and to catch lot of space and time oscillation we set λ_b and λ_c smaller: $0,001$ or less, μ at 0.001 or less and iterated our algorithm twenty times or more computed on sixteen or thirty-two images bloc.

Separation of dynamic from static texture

In this part we present two methods to separate real dynamicity presents in dynamic texture, from static component. In fact we consider the time part in our computation of texture component, only if enough dynamicity is present.

The first method rely on optical flow computation, and on a thresh on the norm of movement vectors.

The second one is determinate by the proximity between time and spatial gradient, thanks to ratios computing into grayscale projection algorithm.

We obtain visually good results (better with second method) and separate well dynamic from static component of the texture part. We can clearly see movement of flowing water, extract in Figure 4. So in our process we only take out moving (or non moving) objects in V component and regularized the corresponding part in U component. Such method present interest for segmentation or characterization on dynamic texture tasks.

Numerical results

We present, in Figure 3, a part of a decomposed sequence of flowing water under wood bridge. We can see the static aspect of U component, regularized in space and in time, seems to be freezed, although texture component, V, present a real dynamic, strengthened by time influence. Only moving things or objects presenting dynamicity are enhanced into V component. In this way we obtain the dynamicity present in video through oscillations along time dimension. Geometrical structures are well regularized and time varying details are strengthen and well captured with our method.

In order to prove that our dynamic decomposition method show more significant result than static decomposition, we present a comparison between two methods (static and dynamic decomposition are both computed with same classic parameters). We can easily see that time impact in result, water in Figure 3 and Figure 6 is well regularized and fluid aspect is well represent in the V component. Moreover, if user tunes parameters to obtain stronger regularization, our algorithm is able to catch wider waves in spatiotemporal texture component: see the circumference of foutain in Figure 6, more regularized (in U component) than wider waves. It's a matter of deep in spatiotemporal texture extraction, wich our algorithm is able to deal with.

In Figure 5 we can clearly see the reenforcement of moving cars texture without that static part and objects present in sequence are taken into account. For example, the simple difference between V component in dynamic and successive classic decomposition, as presented in Figure 5 and Figure 6, show a factor between two and four more details in our model (for a sequence presenting a real dynamic). Moreover reconstruction U + V is faithful to original at about nighty-six percent against about nighty percent in static model.

For more details, demonstration sequence, wider range of results and for a prsentation of similar method, rellying on real different approache [6], please consult this URL: http://perso.univ-lr.fr/mlugiez.

References

[1] G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations (second edition)*, volume 147 of *Applied Mathematical Sciences*. Springer-Verlag, 2006.

[2] Jean-Francois Aujol and Antonin Chambolle. Dual norms and image decomposition models. *International Journal of Computer Vision*, 63(1):85–104, 2005.

[3] Jean-Francois Aujol and Sung Ha Kang. Color image decomposition and restoration. *J. Visual Communication and Image Representation*, 17(4):916–928, 2006.

[4] Antonin Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.*, 20(1-2):89–97, 2004.

[5] Tony F. Chan, Stanley Osher, and Jianhong Shen. The digital TV filter and nonlinear denoising. *IEEE Transactions on Image Processing*, 10(2):231–241, 2001.

[6] Mathieu Lugiez, Michel Ménard, and Abdallah El-Hamidi. Dynamic color texture modeling and color video decomposition using bounded variation and oscillatory functions. To appears in Lecture Notes in Computer Science. Springer, 2008.

[7] R. Peteri, M. Huskies, and S. Fazekas. Dyntex: A comprehensive database of dynamic textures, 2005.

[8] Martin Welk, Joachim Weickert, and Gabriele Steidl. A four-pixel scheme for singular differential equations. In Ron Kimmel, Nir A. Sochen, and Joachim Weickert, editors, *Scale-Space*, volume 3459 of *Lecture Notes in Computer Science*, pages 610–621. Springer, 2005.

Author Biography

Mathieu Lugiez received his Master degree (Applied Informatics and Mathematic) from La Rochelle University (2007). He work on his PhD in informatics and applied mathematic in La Rochelle University under Michel Ménard and Abdallah El-Hamidi direction. His work focuses on extraction and characterization of dynamic texture with variational methods; mainly on spatiotemporal aspect of these problematic.

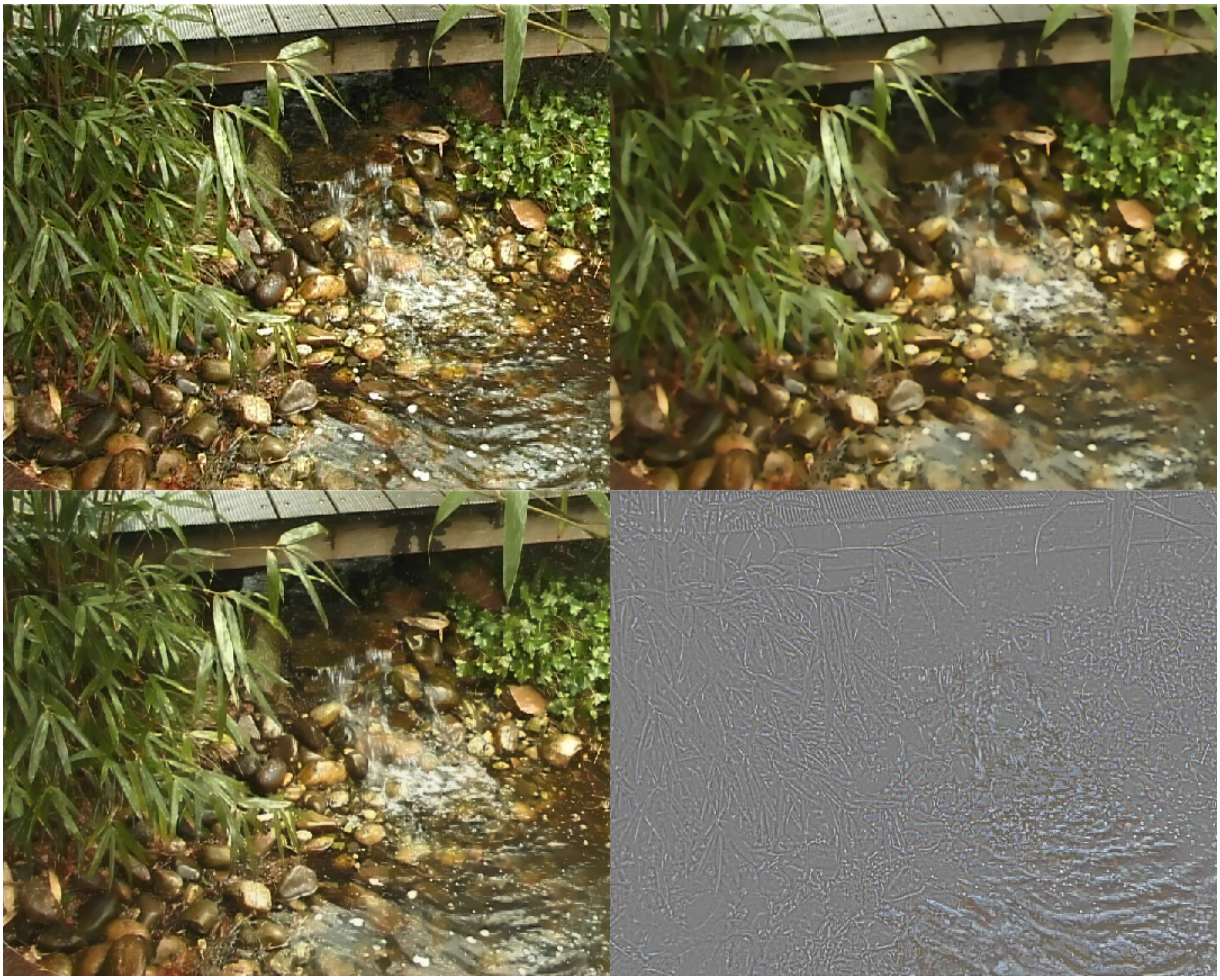

Figure 3. *Top left: f the original image from sequence, top right: U geometrical component, bottom left: Reconstruction U + V, bottom right: V + 127 texture component*

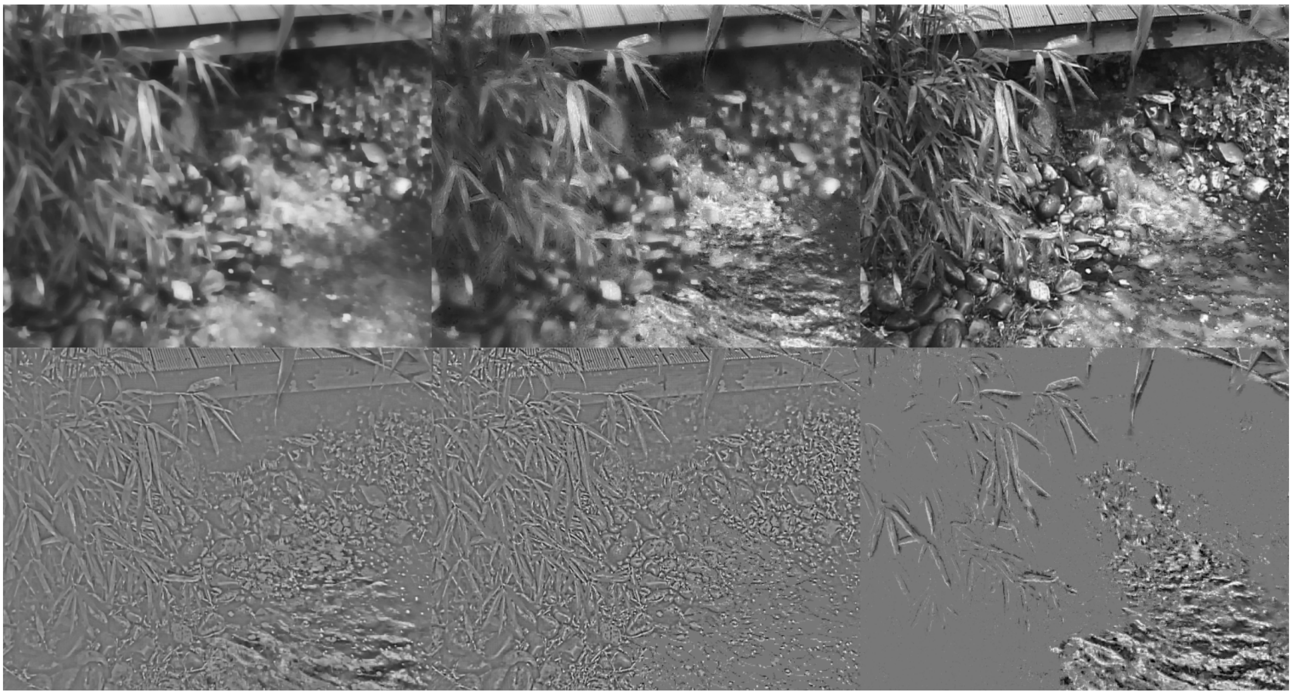

Figure 4. *From top to bottom, U component and V component of spatiotemporal grayscale decomposition. From left to right spatiotemporal decomposition, its static part and its dynamic part taking into account gradient proximity with.*

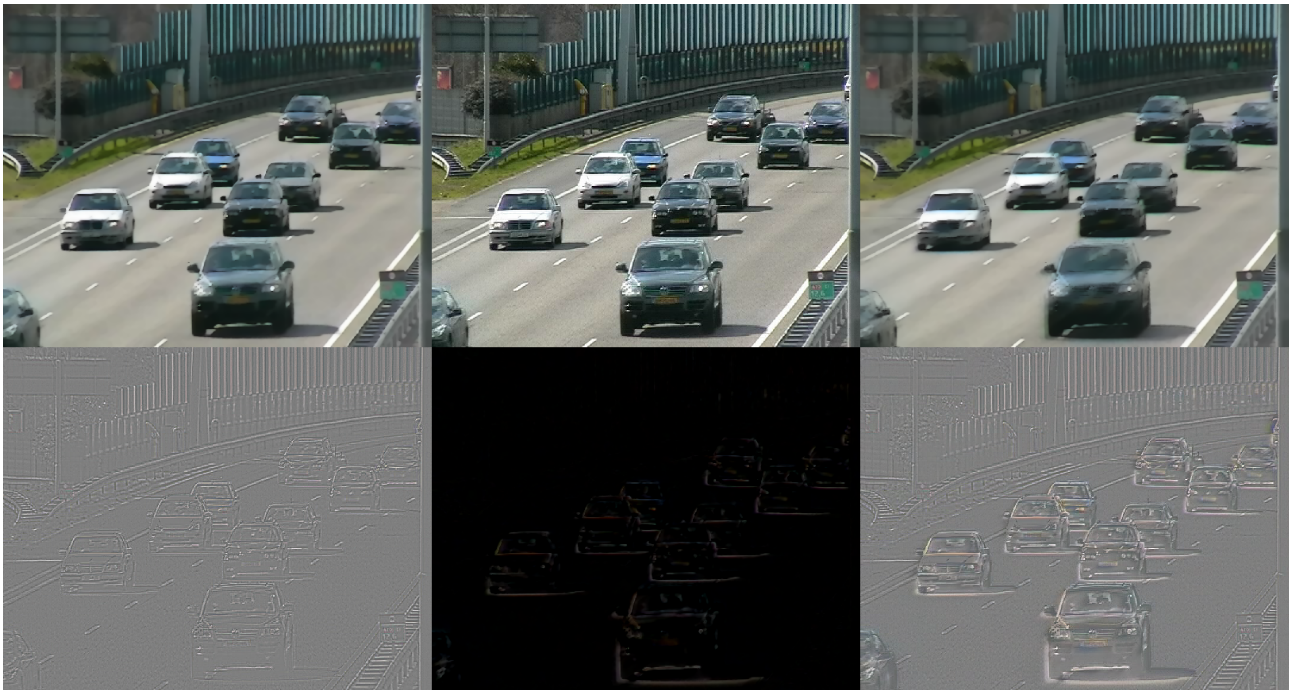

Figure 5. *In left, static decomposition, top: the geometrical component U, bottom: the texture component V. At center, top: image from original sequence f, bottom: simple difference between texture component from classic decomposition and from spatiotemporal decomposition. In right, top: the geometrical component U, bottom: the spatiotemporal texture component V (computed with same parameters than classic decomposition). We can clearly see that only objects in movement are reenforced in our dynamic texture component.*

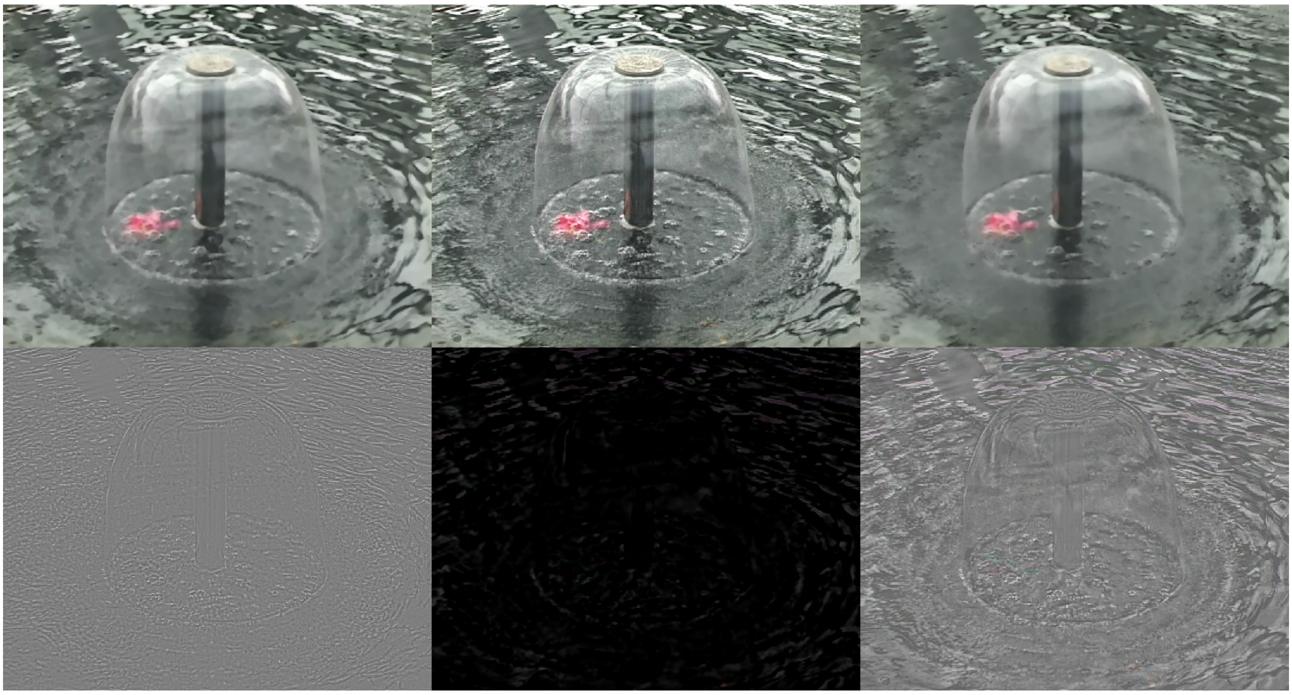

Figure 6. *In left, static decomposition, top: the geometrical component U, bottom: the texture component V. At center, top: image from original sequence f, bottom: simple difference between texture component from classic decomposition and from spatiotemporal decomposition. In right, top: the geometrical component U, bottom: the spatia-temporal texture component V (computed with same parameters than classic decomposition). We can clearly see that water seems to be freezed at the circumference of the fountain on the geometrical component of our decomposition. Moreover lot of movement details appears in undulations of water.*