FROM TV-L^1 TO GATED RECURRENT NETS

Yuqiang Fang[⋆§], *Haiyan Fan*[⋆§], *Lin Sun*[†], *Yulan Guo*[‡], *Zhihao Ma*[⋆]

[⋆]Space Engineering University,
[†]Hong Kong University of Science Technology, [‡] National University of Defense Technology

ABSTRACT

TV-L^1 is a classical diffusion-reaction model for low-level vision tasks, which can be solved by a duality based iterative algorithm. Considering the recent success of end-to-end learned representations, we propose a TV-LSTM network to unfold the duality based iterations into long short-term memory (LSTM) cells. To provide a trainable network, we relax the difference operators in the gate and cell update of TV-LSTM to trainable parameters. Then, the proposed end-to-end trainable TV-LSTMs can be naturally connected with various task-specific networks, e.g., optical flow estimation and image decomposition. Extensive experiments on optical flow estimation and structure + texture decomposition have demonstrated the effectiveness and efficiency of the proposed method.

Index Terms— total variation, optical flow, recurrent neural network, image decomposition

1. INTRODUCTION

The total variation (TV) has been introduced in computer vision by Rudin, Osher, and Fatemi (ROF) [1] as a regularizing criterion for solving inverse problem. In this paper, we study a version of the ROF model that uses the L^1-norm as a measure of the fidelity,which is named as TV-L^1 problem. A duality based iterative algorithm can be used to solve this problem. We will demonstrate in this paper that, when judiciously unfold, the duality based iterations can be formed into variants of long short-term memory (LSTM) cells. The resulting network, namely TV-LSTMs, outperforms the existing duality based method in solving TV-L^1 with a minimal computational budget.

Though the TV-L^1 can be directly mapped to TV-LSTMs, such network is feed-forward, parameter-free and untrainable. To provide a trainable network with greater flexibility, it is essential to connect the TV-LSTMs with a task-specific network and to back-propagate the error differentials. Specifically, for the gate and cell update of TV-LSTMs, we can relax the convolution kernel (used to compute the gradient and divergence) to be trainable parameters. Then, the gate layers and candidate cell layer become fully differentiable. The proposed TV-LSTMs is an end-to-end trainable network and can be naturally connected with a task-specific network.

§ Yuqiang Fang and Haiyan Fan contributed equally.

2. RELATED WORKS

In this work, we focus on emdedding the TV-L^1 optimization procedure into a gated recurrent neural network. In more general terms, this is an example on how to implement iterative algorithms using recurrent neural networks, with similar ideas recently being applied to other domains. For example, Gregor and LeCun [2] proposed LISTA to unfold an iterative soft-thresholding algorithm into a recurrent neural network. This work has prompted the development of RNN based ℓ_1 solver. After that, a number of LISTA-like algorithms have been proposed to achieve a data driven sparse coding. These methods remarkably improve the inference speed [3][4][5]. However, existing methods only focus on the relationship between proximal gradient-based iterations and RNN units.In our work, the primal-dual iteration of the TV-L^1 solver is formed into a variant of LSTM-like unit.

More recently, convolutional nerual networks and energy based models are unified to solve low-level vision tasks, e.g., PDE-Net [6], TVNet [7], variational networks [8], trainable nonlinear reaction diffusion [9]. Comparing to these works which unfolds the TV-L^1 optimization iterations as convolution layers, the proposed TV-LSTMs can be more easily extended to tasks-specific networks and explain the diffusion-reaction process into a recurrent network framework. To the best of our knowledge, this is the first work to bridge TV-L^1 and gated recurrent network.

3. CONNECTING TV-L^1 AND LSTM NETWORKS

3.1. Original TV-L^1 Model

In this paper, we focus on the generic convex optimization problem:

$$\text{Find} \quad \hat{x} \in \underset{x}{\operatorname{argmin}} \underbrace{\mathrm{TV}_i(x)}_{\text{diffusion term}} + \underbrace{\lambda F(x)}_{\text{reaction term}} , \qquad (1)$$

where the diffusfor each location (n_1, n_2) in the discrete domain $\Omega = \{1, \ldots, N_1\} \times \{1, \ldots, N_2\}$, with Neumann boundary conditions.The diffusion term $\mathrm{TV}_i(x)$ is the isotropic total

variation of x. The reaction term $F(x)$ is incorporated to apply our model to different processing problems. Here, F is a convex, proper, lower semicontinuous function [10]. Specially, we consider the following TV-L^1 model:

$$\text{Find} \quad \hat{x} \in \underset{x}{\operatorname{argmin}} \, \mathrm{TV}_i(x) + \lambda \|f(x)\|_{L^1(\Omega)} \,. \tag{2}$$

Here, $\|\cdot\|_{L^1(\Omega)}$ is the L^1-norm.

3.2. A Duality based Implementation

An efficient way to solve the optimization problem defined in equation (2) is to introduce the following quadratic penalty functional:

$$\text{Find} \quad (\hat{x}, \hat{v}) \in \underset{x,v}{\operatorname{argmin}} \, \mathrm{TV}_i(x) + \frac{1}{2\theta}\|x - v\|^2 + \lambda \|f(v)\|_{L^1(\Omega)} \,. \tag{3}$$

Setting θ to a very small value forces the minimum of (3) to occur when x and v are nearly equal. The relaxation in (3) can be minimized by alternatively fixing x or v, and solving for the other variable.

Fixed v, solve

$$\text{Find} \quad \hat{x} \in \min_{x} \mathrm{TV}_i(x) + \frac{1}{2\theta}\|x - v\|^2 \,. \tag{4}$$

Fixed x, solve

$$\text{Find} \quad \hat{v} \in \underset{v}{\operatorname{argmin}} \, \frac{1}{2\theta}\|x - v\|^2 + \lambda \|f(v)\|_{L^1(\Omega)} \,. \tag{5}$$

The first sub-problem can be solved by Chambolle's duality-based algorithm [11] [12]:

$$\text{Find} \quad (\hat{x}, \hat{p}) \in \min_{x} \max_{\bar{p}} \left\{ \frac{1}{2\theta}\|x - v\|^2 + \langle x, -\mathrm{div}\,(\bar{p}) \rangle \right\} , \tag{6}$$

where $\|\bar{p}\,[n1, n2]\,\| \leq 1, \quad \forall\,(n_1, n_2) \in \Omega$. For each location (n_1, n_2) in the discrete domain $\Omega = \{1, \ldots, N_1\} \times \{1, \ldots, N_2\}$, with Neumann boundary conditions.

In order to obtain the minimum of (6), the solution of the dual variable $\hat{p}$ should satisfy that,

$$\text{Find} \quad \hat{p} \in \min_{\bar{p}} \left\{ \|\theta \mathrm{div}\,(\bar{p}) - v\|^2 : \|\bar{p}\,[n1, n2]\,\| \leq 1 \right\} . \tag{7}$$

(7) can be solved by a semi-implicit gradient descent algorithm. We choose $\tau > 0$, let $p^0 = 0$ and for any iteration $k \geq 0$,

$$p_{i,j}^{k+1} = \frac{p_{i,j}^{k} + \tau \left(\nabla \left(\mathrm{div} p^k - v/\theta \right) \right)_{i,j}}{1 + \tau \| \left(\nabla \left(\mathrm{div} p^k - v/\theta \right) \right)_{i,j} \|}, \quad \forall\,(i,j) \in \Omega \,. \tag{8}$$

Then, in iteration $k+1$, the solution of x^{k+1} can be simply given by

$$x^{k+1} = v^{k+1} - \theta \mathrm{div}\left(p^{k+1}\right) . \tag{9}$$

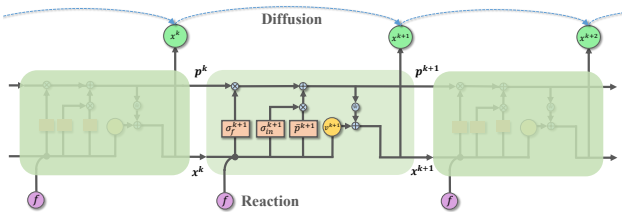

Fig. 1. TV-L^1 iterations as LSTMs network

For the second sub-problem defined in equation (5), without loss of generality, we assume that the function $f(v)$ has the following form:

$$f(v) = Av + b \,, \tag{10}$$

This minimum problem can be solved as:

$$v^{k+1} = x^k + TH\left(x^k, \lambda\theta\right) , \tag{11}$$

with the thresholding operation

$$TH\left(x^k, \lambda\theta\right) = \begin{cases} \lambda\theta A, f(x^k) < -\lambda\theta \|A\|^2 \\ -\lambda\theta A, f(x^k) > \lambda\theta \|A\|^2 \\ -f(x^k)\dfrac{A}{\|A\|^2}, \|f(x^k)\| \leq \lambda\theta \|A\|^2 \end{cases} \tag{12}$$

3.3. Connection with LSTM-like Recurrent Networks

In this subsection, we map specialized TV-L^1 iterations to a LSTM-like network. We partition the TV-L^1 iterations as gate updates:

$$\sigma_f^{k+1} = \sigma\left(\frac{1}{1 + \frac{\tau}{\theta}\|\nabla x^k\|}\right), \quad \sigma_{\text{in}}^{k+1} = \sigma\left(\frac{\frac{\tau}{\theta}}{1 + \frac{\tau}{\theta}\|\nabla x^k\|}\right) , \tag{13}$$

where $v^{k+1} = x^{k+1} + TH\left(x^{k+1}, \lambda\theta\right)$, and the cell updates

$$\bar{p}^{k+1} \leftarrow -\nabla x^k, \quad p^{k+1} \leftarrow \sigma_f^{k+1} \odot p^k + \sigma_{\text{in}}^{k+1} \odot \bar{p}^{k+1} , \tag{14}$$

and output updates

$$x^{k+1} \leftarrow v^{k+1} - \theta \mathrm{div}\left(p^{k+1}\right) . \tag{15}$$

Here, $\odot$ denotes the Hadamard product. $\sigma(\cdot)$ is a sigmoidal activation.

Starting from initial values of p^0 and x^0, the TV-L^1 implementation defined in equations (13)-(15) closely mirrors a canonical LSTM. The output of the network at time-step k is x^k, and p^k is considered as the internal LSTM memory cell, or the latent cell state. Procedding further, p^k is fed to four separate subnetworks (as shown in Figure 1): (i) the forget gate σ_f^{k+1}, (ii) the input gate σ_{in}^{k+1}, (iii) the output update x^{k+1} and (iv) the candidate cell update $\bar{p}^{k+1}$. The forget gate σ_f^{k+1} determines how large we forget the old cell state elements p^k, and the input gate σ_{in}^{k+1} measures how large we rescale

signals from the candidate input update $\bar{p}^{k+1}$. Then, the two re-weighted quantities are combined to form the new cell state p^{k+1}. The output update x^{k+1} is generated as a scaled version of the new cell update p^{k+1} and the internal state v^{k+1}. Thus, the iterative process in TV-L^1 model can be unfolded as a layer-to-layer LSTM-like network.

4. THE END-TO-END TRAINABLE NETWORK

In this section, we will describe our end-to-end trainable network for TV-L^1 model, namely, TV-LSTMs. To imitate the iterative process in TV-L^1, TV-LSTMs has two gates to protect and control the cell state step-by-step.

4.1. Forget Gate

The first step is to determine what information to throw away from the cell p^k. This is achieved by forget gate layer σ_f^{k+1} in equation (13). In σ_f^{k+1}, the computation of gradient $\|\nabla x\|$ can be discretized with upwind difference to define an isotropic TV [13]. By defining the kernel $w = [1, -1]$, $\|\nabla x\|$ can be written as

$$\|\nabla x\| = \sum_{n_1=1}^{N_1} \sum_{n_2=1}^{N_2} ((x * w)^2[n_1, n_2] + (x * w^\top)^2[n_1, n_2])^{\frac{1}{2}}, \tag{16}$$

where $*$ is the convolution operation, $w^\top$ denotes the transposition of w. When we relax the convolution kernel to trainable parameters w_f, the forget gate layer transforms to a fully differentiable layer, which is expressed as a computational graph consisting of convolution.

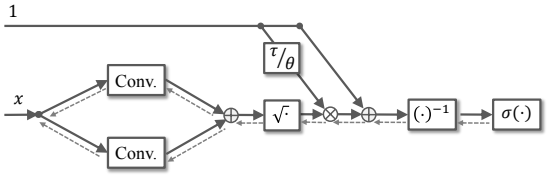

Fig. 2. The computation graph of forget gate.

4.2. Input Gate and Candidate Cell

The next step is to control the new information to store in a cell state. This process is conducted by an input gate layer σ_{in}^{k+1} and a candidate cell $\bar{p}^{k+1}$. Similar to forget gate, sigmoidal unit σ_{in}^{k+1} takes activation from current $\|\nabla x^k\|$. Thus, it is convenient to convert σ_{in}^{k+1} to a differentiable layer following equations (16). Once the gradient of each component of ∇x^k is computed via the forward difference, the candidate update $\bar{p}^{k+1}$ can be relaxed as convolution operations:

$$\bar{p}^{k+1} \leftarrow (-x^k * w_{in}, -x^k * w_{in}^\top). \tag{17}$$

4.3. Output Update

Finally, we put cell state p^{k+1} and internal state v^{k+1} to determine the output x^{k+1}, like equation (15). In equation (15), the

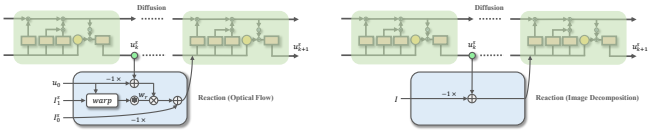

Fig. 3. An illustration of feed forward networks in TV-LSTMs for optical flow (left) and image decomposition (right).

discrete divergence operation of $p = (p_1, p_2)$ can be defined via the backward difference for each location (n_1, n_2) in the discrete domain using Neumann boundary conditions. To take the benefits of end-to-end learning, we write the backward difference to

$$\text{div}\,(p) = \hat{p}_1 * w_o + \hat{p}_2 * w_o^\top, \tag{18}$$

where $\hat{p}$ denotes the shifted p to ensure convolution in a backward direction. In this situation, by replaying w_o as a trainable convolution, the output layer naturally falls into a simple learnable differentiable subnetwork structure.

5. TASK-SPECIFIC NETWORKS

5.1. Task I: Optical Flow Estimation

Our TV-LSTMs can be designed for different computer vision problems. The task of optical flow estimation is to seek the displacement field to describe the pixel shifts between two successive images. Let I_0 and $I_1 : (\Omega \in \mathbb{R}^2) \mapsto \mathbb{R}$ is an image pair, $\mathbf{u} = [u_1, u_2]^\top$ is a two-dimensional displacement field. Assuming that the brightness of an image is constant, we define reaction term in equation (2) as $f(\mathbf{u}) = I_1(\mathbf{x} + \mathbf{u}) - I_0(\mathbf{x})$. Furthermore, the non-linear term $I_1(\mathbf{x} + \mathbf{u})$ can be linearized using Taylor expansions, yielding to $\hat{f}(\mathbf{u}) \approx \nabla I_1(\mathbf{x} + \mathbf{u}_0) \cdot (\mathbf{u} - \mathbf{u}_0) + I_1(\mathbf{x} + \mathbf{u}_0) - I_0(\mathbf{x})$, where $\mathbf{u}_0$ is an initial approximation close to $\mathbf{u}$. Therefore, the specialization of equation (3) to optical flow is defined as:

$$\text{Find} \quad \hat{\mathbf{u}} \in \underset{\mathbf{u}}{\text{argmin}}\, \text{TV}_i(\mathbf{u}) + \lambda \|\hat{f}(\mathbf{u})\|_{L^1(\Omega)}. \tag{19}$$

Here, the gradient of I_1 can be computed via central difference $\nabla I_1 = (I_1 * w_r, I_1 * w_r^\top)$, with $w_r = [-0.5, 0, 0.5]$. Consequently, by introducing the task-specific reaction part $\hat{f}(\mathbf{u})$, we can specialize the fully differentiable TV-LSTM unit for optical flow estimation as shown in Figure 3. For the training of TV-LSTM, we can use the standard End-Point Error (EPE) between estimation $\hat{\mathbf{u}}$ and groundtrue $\mathbf{u}_{gt}$ as the loss function. With the learned weights $\mathcal{W} = \{w_f, w_{in}, w_o, w_r\}$, the flow field can be computed using the forward network of TV-LSTMs. The details of TV-LSTM based optic flow estimation method in a multi-scale scheme are presented in Algorithm 1.

5.2. Task II: Structure + Texture Image Decomposition

We further consider the application of image decomposition with TV-LSTMs. For a given image $I : \Omega \mapsto \mathbb{R}$, the task of

Algorithm 1 Optical flow estimation with forward propagation in TV-LSTMs

Parameters: $\lambda, \theta, \tau, N_{warps}, N_{scales}$
Input: Two consecutive images $I_0, I_1, \mathbf{u}_0$
Output: Flow field $\mathbf{u}$
Compute down-scaled images $I_0^s, I_1^s, s = 1, 2, \ldots, N_{scales}$.
for $s \leftarrow N_{scales}$ to 1 **do**
  for $w \leftarrow 1$ to N_{warps} **do**
    $\mathbf{u}^s = \textbf{ForwardProgation}(I_0^s, I_1^s, \mathbf{u}_0, \lambda, \theta, \tau, \mathcal{W})$;
  end for
  if $s > 1$ **then**
    scale-up $\mathbf{u}^s$ to $\mathbf{u}^{s-1}$, $\mathbf{u}_0 \leftarrow \mathbf{u}^{s-1}$;
  end if
end for

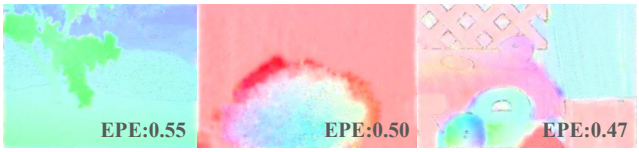

Fig. 4. Example optical flow fields achieved by trained TV-LSTMs on MiddleBurry.

structure-texture image decomposition refers is to split I into a structure part $\mathbf{u}$ and a textural part $\mathbf{v}$. Ideally, the structure part $\mathbf{u}$ contains sharp edges, and the texture part $\mathbf{v}$ contains oscillatory patterns in I. A typical choice to penalize $\mathbf{u}$ and $\mathbf{v}$ is defined as following TV-L^1 problem:

$$\text{Find} \quad \{\hat{\mathbf{u}} \in \underset{\mathbf{u}}{\operatorname{argmin}} \, \text{TV}_i(\mathbf{u}) + \lambda \|\mathbf{v}\|_{L^1(\Omega)} : I = \mathbf{u} + \mathbf{v}\}. \tag{20}$$

Here, the total variation penalizes oscillations, and allows piecewise smooth in $\mathbf{u}$. L^1-norm characterize the texture $\mathbf{v}$ components. Thus, we can rewrite reaction term in equation (10) as $f(\mathbf{u}) = -\mathbf{u} - I$. The problem can now be formulated by the proposed TV-LSTMs, as illustrated in Figure 3.

6. EXPERIMENTS

6.1. Optical Flow

Recently, Fan, L. et al. [7] proposed a trainable neural network for optical flow estimation, namely TVNet. In this section, we mainly compare our method with TVNet. We followed the same experimental settings as TVNet, performed on the MiddleBurry dataset [14]. All image pairs with ground true optical flows were used as training data. The estimation errors on MiddleBurry are measured by EPE. TVNet-3-1-10 achieves an average EPE of 2.00 without training and 0.52 with training. Our work achieves an average EPE sore of 0.36 with 20 iterations. Table 1 presents the qualitative results achieved by TV-LSTMs and TVNet. From this table, it can be seen that the proposed method outperforms the original TVNet.

Table 1. The average EPEs on MiddleBurry

Methods*	No Training	Training
TVNet(1-1-10)	3.47	1.24
TVNet(3-1-10)	2.00	0.52
TV-LSTMs(1-1-30)	2.55	1.30
TV-LSTMs(3-1-10)	1.77	0.51
TV-LSTMs(3-1-30)	1.67	0.36

*TVNet / TV-LSTMs($N_{warps} - N_{scales} - N_{iters}$)

6.2. Image Decomposition

In this section, we present the qualitative results achieved by the TV-LSTMs based image decomposition algorithm. Experiments were performed on 8 color images selected from the dataset [14]. All pixel values were normalized to [0,1]. Parameter λ in Equation(20) is a weight inevitable in regularized optimization. It is set to 0.15 empirically in all experiments. Meanwhile, we set the number of units in TV-LSTMs to 20. For the training of TV-LSTMs, we used the state-of-the-art image decomposition method [16] to generate the groundtrue structural images $\mathbf{u}_{gt}$, and define the loss function as $\|\mathbf{u} - \mathbf{u}_g\|_{\ell_1}$.

Figure 5 shows several example results achieved by different decomposition algorithms. It can be seen that, our TV-LSTMs can produces well-decomposed textures (e.g., spines in the cactus scene) and piecewise-constant structures (e.g., floor in the Barbara scene), even though we do not use a training step. Moreover, our TV-LSTMs can achieve a speed of 7 fps on images with a resolution of 410×620 pixels.

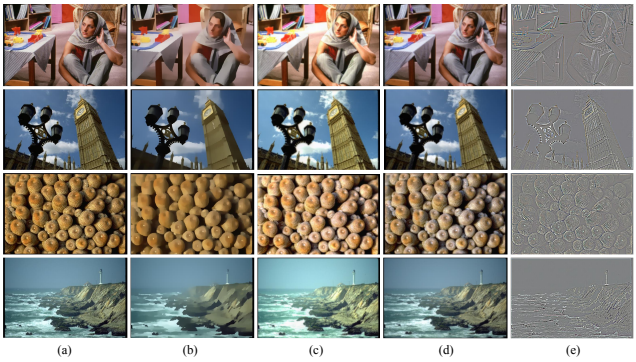

Fig. 5. Examples of decomposition results achieved by different methods. (a) Input. (b) Structures ([16]). (c) Structures (TV-LSTMs without training). (d) Structures (trained TV-LSTMs). (e) Textures (trained TV-LSTMs).

7. CONCLUSION

This work proposes a neural network, namely TV-LSTMs, to achieve the TV-L^1 model in an end-to-end manner. We show that the optimization of TV-L^1 can be unfolded as a LSTM-like network with a novel computational unit. Furthermore, our TV-LSTMs can be naturally extended to a task-specific network by using a specific reaction term. Extensive experimental results show the effectiveness of our method.

8. REFERENCES

[1] Rudin, L.I., Osher, S., Fatemi, E., "Nonlinear total variation based noise removal algorithms", Phys. D: Nonlinear Phenom, 60(1):259–268, 1992.

[2] Gregor, K., Lecun, Y., "Learning Fast Approximations of Sparse Coding", International Conference on Machine Learning, pp. 399-406. Omnipress, 2010.

[3] Diamond, S., Sitzmann, V., Heide, F., Wetzstein, G., "Unrolled Optimization with Deep Priors",arXiv:1705.08041, 2017.

[4] Yang, Y., Sun, J., Li, H., Xu, Z., "Deep Admm-net for Compressive Sensing MRI",NIPS2016.

[5] Hao, H., Huang, W., Gan, C., Ermon, S., Gong, B., "From Bayesian Sparsity to Gated Recurrent Nets", Advances in Neural Information Processing System, NIPS2017.

[6] Long, Z., Lu, Y., Ma, X., Dong, B., "PDE-Net: Learning PDEs from Data", ICML2018.

[7] Fan, L., Huang, W., Gan, C., Ermon, S., Gong, B., "End-to-End Learning of Motion Representation for Video Understanding", IEEE Conference on Computer Vision and Pattern Recognition, CVPR2018.

[8] Kobler, E., Klatzer, T., Hammernik, K., Pock, T., "Variational Networks: Connecting Variational Methods and Deep Learning", German Conference on Pattern Recognition, pp. 281–293. Springer, 2017.

[9] Chen, Y., Pock, T., "Trainable Nonlinear Reaction Diffusion: a Flexible Framework for Fast and Effective Image Restoration", IEEE Transactions on Pattern Analysis Machine Intelligence", 39(6):1256-1272, 2017.

[10] Vogel, C., Pock, T., "A Primal Dual Network for Low-Level Vision Problems", Pattern Recognition: 39th German Conference, pp. 189-202. Basel, Springer–Verlag, 2017.

[11] Chambolle, A., "An Algorithm for Total Variation Minimization and Applications", Kluwer Academic Publishers, 2004.

[12] Perez, J. S., "TV-L1 Optical Flow Estimation", Image Processing on Line, 2(4):137-150, 2013.

[13] Chambolle, A., Levine, S. E., Lucier, B. J., "An Upwind Finite-Difference Method for Total Variation-based Image Smoothing", SIAM Journal on Imaging Sciences, 4(1):277-299, 2011.

[14] Baker, S., Roth, S., Scharstein, D., Black, M. J., Lewis, J. P., Szeliski, R., "A Database and Evaluation Methodology for Optical Flow", IEEE International Conference on Computer Vision, pp. 1-31, 2007.

[15] Buades, A., Lisani, J. L., "Directional filters for cartoon + texture image decomposition", Image Processing on Line, pp. 75-88, 2016.

[16] Xu, L., Yan, Q., Xia, Y., Jia, J., "Structure Extraction from Texture via Relative Total Variation", ACM Transactions on Graphics, 31(6):139, 2012.