Total variation regularization of local-global optical flow

Marius Drulea, Sergiu Nedevschi

***Abstract*— More data fidelity terms in variational optical flow methods improve the estimation's robustness. A robust and anisotropic smoother enhances the specific fill-in process. This work presents a combined local-global (CLG) approach with total variation regularization. The combination of bilateral filtering and anisotropic (image driven) regularization is used to control the propagation phenomena. The resulted method, CLG-TV, is able to compute larger displacements in a reasonable time. The numerical scheme is highly parallelizable and runs in real-time on current generation graphic processing units.**

I. Introduction

MOTION estimation plays an important role in computer vision applications. Two of the most known and simple to analyze methods were developed by K.P. Horn, B.G. Schunck in [1] and B. D. Lucas, T. Kanade in [2]. The Horn-Schunck (HS) model proposes a variational approach to optical flow estimation:

$$E_{HS} = \int_{\Omega} \left[\lambda \cdot r(u,v)^2 + (\|\nabla u\|^2 + \|\nabla v\|^2) \right] (1).$$

E_{HS} is the functional error to be minimized and $r(u,v)$ is the residual between the images, defined below:

$$I_1(\vec{x}) = I_2(\vec{x}+\vec{u}) = I_2(\vec{x}+\vec{u}_0+\overbrace{\vec{u}-\vec{u}_0}) \cong \overbrace{I_2(\vec{x}+\vec{u}_0)}^{I_{2w}} + (\vec{u}-\vec{u}_0)^T \cdot \nabla I_2(\vec{x}+\vec{u}_0) \;(2),$$

$$r(u,v) \overset{def.}{:=} \overbrace{I_{2w}-I_1}^{I_t} + (\vec{u}-\vec{u}_0)^T \cdot \nabla I_{2w} \cong 0$$

where $\vec{x}=(x,y)$, $\vec{u}=(u,v)$ and $\vec{u}_0$ is an initial estimate.

The terms of order higher than two are ignored. The above approximation is valid only for small values of the displacements u and v. The optimal solution of E_{HS} is calculated using the Euler-Lagrange equations and a Jacobi (not mandatory) iterative scheme.

Lucas-Kanade (LK) is a local method and assumes that small regions of pixels have the same flow. It is natural to consider a minimization procedure for the total error of that region: $E_{LK}(u,v) \approx \sum_{region} w \cdot r(u,v)^2$, which is a least-square problem. Here, w represents the weighting factor. In order to find the minimizers of E_{LK}, the derivatives are set to zero. The resulting system is 2x2 with the following determinant:

$$\begin{pmatrix} \sum w \cdot \left(I_{2w}^x\right)^2 & \sum w \cdot I_{2w}^x \cdot I_{2w}^y \\ \sum w \cdot I_{2w}^x \cdot I_{2w}^y & \sum w \cdot \left(I_{2w}^y\right)^2 \end{pmatrix} (3)$$

The determinant can be singular or nearly singular if the image derivatives for the corresponding region are close to zero. Where the determinant is non-singular, i.e. the pixel is a relatively strong feature, the method performs well.

A new variational approach results after combining LK and HS:

$$E_{CLG-HS} = \int_{\Omega} \left[\lambda \cdot \left(\sum_{region} w \cdot r(u,v)^2 \right) + (\|\nabla u\|^2 + \|\nabla v\|^2) \right] (4).$$

The combination aims to improve the robustness of the HS model [3]. The main reason to do this is to enhance the flow accuracy for very large displacements. Such situations occur in traffic scenarios where the cars' speeds can be high. On the other hand, the system camera(s) may move in the opposite direction and so the cars displacement in the images becomes even higher. The integration of LK into HS model increases the number of the data fidelity terms and the effect is an improved behavior for large homogenous areas with large flows.

The HS and CLG-HS models use the squared L2 norm and therefore they have some disadvantages like sensitivity to noise, over-propagation and the facts that they are not directionally selective and do not preserve motion boundaries. In order to significantly reduce these problems, a new variational model is proposed and a parallel numerical scheme is presented. The next section presents the new model and its minimization procedure.

II. The Proposed Model

L1-norm is a better choice for modeling real problems involving discrete signals. Rudin, Osher and Fatemi originally suggested this and introduced the total variation in image processing domain [9]. L2-norm is easy to analyze mathematically, but is sensitive to outliers. L1-norm is robust in the presence of outliers, but is very difficult to analyze, especially in the case of functionals ([4],[9]). The Huber penalty function is also a good choice for a robust estimation.

In [6] an alternative to the HS model is proposed: $E_{TV-L1} = \int_{\Omega} \left[\lambda \cdot |r(u,v)| + (\|\nabla u\| + \|\nabla v\|) \right] (5)$. (TV-L1 = total variation, L1). As an observation, the data term is in L1 and the smoothness term represents the isotropic total variation. Since the functions composing the functional are not differentiable, the minimization is not trivial. The experiments reveal clearly more accurate results than HS.

Manuscript received April 10, 2011. This work was partly supported by the research project INTERSAFE-2 and partly by CNCSIS – UEFISCSU, project number PNII – IDEI 1522/2008. INTERSAFE-2 is part of the 7th Framework Programme, funded by the European Commission. The partners of INTERSAFE-2 thank the European Commission for supporting the work of this project.

M. Drulea is with the Computer Science Department, Technical University of Cluj-Napoca, Memorandumului Str. 28, 400114 Cluj-Napoca (e-mail: marius.drulea@ cs.utcluj.ro).

S. Nedevschi is with the Computer Science Department, Technical University of Cluj-Napoca, Memorandumului Str. 28, 400114 Cluj-Napoca (phone: +40-264-401219; e-mail: Sergiu.Nedevschi@cs.utcluj.ro).

The above functional is decomposed into three parts, in order to match the minimization scheme proposed in [4]:

$$E_{TV-1} = \int_{\Omega} \left[\lambda \cdot | r(\hat{u},\hat{v}) | + \frac{1}{2\cdot\theta}(u-\hat{u})^2 + \frac{1}{2\cdot\theta}(v-\hat{v})^2 \right] (6),$$

$$E_{TV-u} = \int_{\Omega} \left[\frac{1}{2\cdot\theta}(u-\hat{u})^2 + \|\nabla u\| \right] (7),$$

$$E_{TV-v} = \int_{\Omega} \left[\frac{1}{2\cdot\theta}(v-\hat{v})^2 + \|\nabla v\| \right] (8).$$

While TV-1 is easily solved point-wise, the minimization of TV-u and TV-v is a difficult task. The last two functionals are under the Euler-Lagrange conditions, but the resulting system is highly nonlinear. Their solution can be computed using the numerical scheme proposed in [4]. The solution was originally developed to solve a denoising problem and it is subject of convex optimization ([4],[10]).

In order to increase the robustness of this model a local-global combination is proposed:

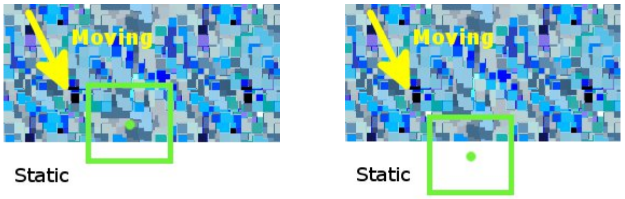
$$E_{CLG-TV} = \int_{\Omega} \left[\lambda \cdot \left(\sum_{region} w \cdot r(u,v)^2 \right) + (\|\nabla u\| + \|\nabla v\|) \right] (9).$$

As in CLG-HS, the idea is to use more data fidelity terms in order to increase the robustness. The data terms are squared and the smoothness terms are not. Therefore, the model uses a robust regularization. The main reason to keep the data fidelity terms squared is to keep the minimization scheme as simple as possible and to achieve real-time performance.

The CLG-TV model developed so far is isotropic, it propagates the flow in all directions, regardless of local properties. To enhance the propagation, an anisotropic diffusion filter is considered. The effect of this filter is to reduce the propagation of the flow for image features (corners, edges) and to allow a higher "fill-in" effect for untextured areas. Obviously, the diffusion coefficient, called diffusion tensor, should depend on image derivatives. Perona and Malik [11] pioneered the idea of anisotropic diffusion and proposed two functions for the diffusion coefficient: $D(\|\nabla I\|) = \frac{1}{1+(\|\nabla I\|/K)^2}$ and $D(\|\nabla I\|) = e^{-(\|\nabla I\|/K)^2}$, where the constant K controls the sensitivity to edges and is usually chosen experimentally. For a better control, the following diffusion tensor is used to alter the CLG-TV model: $D(\|\nabla I\|) = e^{-\alpha \cdot \|\nabla I\|^{\beta}}$. The altered model now reads:

$$E_{CLG-TV} = \int_{\Omega} \left[\lambda \cdot \left(\sum_{region} w \cdot r(u,v)^2 \right) + (\|D \cdot \nabla u\| + \|D \cdot \nabla v\|) \right] (10).$$

The local window representing the data fidelity is another source of over-propagation. The pictures below illustrate this phenomenon. In Fig.1 a), the local window is centered into the moving region. In this case, the Lucas-Kanade method correctly computes the pixel's flow. In Fig.1 b), the center of the window is placed into the static region, so the pixel has no motion. The top of the window is in the moving region. Since the method tries to match the moving part (beside the static one) of the window, the pixel appears to have a motion, which is not true. Traditional methods use a Gaussian filter to weight the importance of the pixels in the local window, but this is not satisfactory for all cases, including the scenario shown in Fig. 1. The most elegant method to avoid such a behavior is to use a bilateral filter.



a)The window is centered in the moving region; the pixel's flow is correctly determined

b)The window is centered in the static region; the pixel's flow is incorrectly determined

Fig. 1. Two effects of the local window

The bilateral filter ([13], [14]) combines two types of filters: a Gaussian (space) filter and a range (color, intensity) filter. The weight of a pixel decreases as the distance from the center increases. In addition, the weight decreases if the intensity of the pixel is different from the center's intensity.

$$BF[I]_p = \underbrace{\overbrace{\frac{1}{W_p}}^{\substack{normalization\\factor}} \cdot \sum_{q \in region} \overbrace{G_{\sigma_s}\left(\|p-q\|\right)}^{space} \cdot \overbrace{G_{\sigma_r}\left(|I_p - I_q|\right)}^{range}}_{the\,weight\,of\,I_q} \cdot I_q \; (11),$$

where G_{σ_s} denotes the spatial Gaussian filter and G_{σ_r} denotes the range (intensity) Gaussian filter. The product of these coefficients represents the weight for the pixel q of the region. The expression is then normalized in order to have a "fair" filter: the coefficients should sum to 1. Fig. 2 shows the effect of the bilateral filtering. Fig.2 c) shows the range filter for a pixel situated at the border of two regions with different intensities. Fig.2 d) shows the combined kernel. The pixels with the same intensity as the center weight more than the others. The output in Fig.2 e) keeps the edge of the initial image Fig.2 a).

The proposed CLG-TV model includes the bilateral weighting. Since the weights does not depend on the unknowns and for simplicity, the theoretical CLG-TV model is kept in the form (10). The CLG-TV functional error is decomposed into three parts:

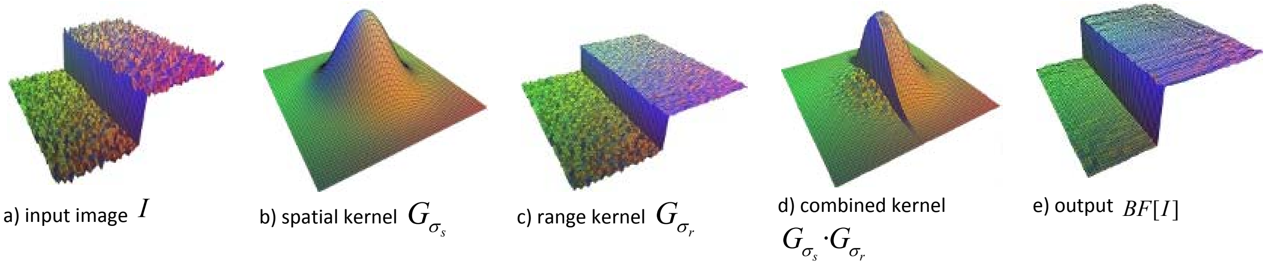
$$E_{CLG-TV-1} = \int_{\Omega} \left[\lambda \left(\sum_{region} bfwr(\hat{u},\hat{v})^2 \right) + \frac{1}{2\cdot\theta}(u-\hat{u})^2 + \frac{1}{2\cdot\theta}(v-\hat{v})^2 \right] (12),$$

$$E_{TV-u} = \int_{\Omega} \left[\frac{1}{2\cdot\theta}(u-\hat{u})^2 + \|D \cdot \nabla u\| \right] (13),$$

$$E_{TV-v} = \int_{\Omega} \left[\frac{1}{2\cdot\theta}(v-\hat{v})^2 + \|D \cdot \nabla v\| \right] (14),$$

where bfw denotes the bilateral filter weight.

For CLG-TV-1, u and v are considered fixed and $\hat{u},\hat{v}$ are the unknowns. Since the functional does not depend on image derivatives, a point-wise minimization is employed.



a) input image I b) spatial kernel G_{σ_s} c) range kernel G_{σ_r} d) combined kernel $G_{\sigma_s} \cdot G_{\sigma_r}$ e) output $BF[I]$

Fig. 2. The bilateral filter (the images are taken from [13])

Setting the derivatives to zero, the following linear system of equations results:

$$\begin{pmatrix} 1+2\lambda\theta\sum bfw\left(I_{2w}^x\right)^2 & 2\lambda\theta\sum bfw\cdot I_{2w}^x\cdot I_{2w}^y \\ 2\lambda\theta\sum bfw\cdot I_{2w}^x\cdot I_{2w}^y & 1+2\lambda\theta\sum bfw\left(I_{2w}^y\right)^2 \end{pmatrix}\cdot\begin{pmatrix}\hat{u}\\ \hat{v}\end{pmatrix}=\begin{pmatrix} u-2\lambda\theta\sum bfw\cdot I_{2w}^x\cdot r_0 \\ v-2\lambda\theta\sum bfw\cdot I_{2w}^y\cdot r_0\end{pmatrix}$$

(15), where $r_0 := I_t - u_0\cdot I_{2w}^x - v_0\cdot I_{2w}^y$. The main determinant of (15) is always non-zero.

For TV-u and TV-v, u and v are the unknowns. Their solutions are found adapting the algorithm presented in [4]. Since the problems are similar, the numerical schemes are expected to be similar. The following steps leads to the desired numerical scheme:

The Euler-Lagrange equation for TV-u is :

$$-div\left(D\cdot\frac{\nabla u}{\|\nabla u\|}\right)+\frac{1}{\lambda}\cdot(u-\hat{u})=0 \text{ (16).}$$

Let $\overrightarrow{p_u}$ be the dual variable defined as:

$$\overrightarrow{p_u}=\frac{\nabla u}{\|\nabla u\|}\Rightarrow \overrightarrow{p_u}\cdot\|\nabla u\|-\nabla u=0, \left\|\overrightarrow{p_u}\right\|\le 1 \text{(17).}$$

The relation (16) is rewritten in the form:

$$-div\left(D\cdot\overrightarrow{p_u}\right)+\frac{1}{\lambda}\cdot(u-\hat{u})=0\Rightarrow u=\lambda\cdot div\left(D\cdot\overrightarrow{p_u}\right)+\hat{u} \text{ (18).}$$

Substituting u in (17) and dividing by λ results to:

$$\overrightarrow{p_u}\cdot\left\|\nabla\left(div\left(D\cdot\overrightarrow{p_u}\right)+\hat{u}/\lambda\right)\right\|-\nabla\left(\lambda\cdot div\left(D\cdot\overrightarrow{p_u}\right)+\hat{u}/\lambda\right)=0 \text{ (19),}$$

which can be solved using a fixed-point iteration scheme:

$$\overrightarrow{p_u^{n+1}}=\frac{\overrightarrow{p_u^n}+\tau\cdot\nabla\left(div\left(D\cdot\overrightarrow{p_u^n}\right)+\hat{u}/\lambda\right)}{1+\tau\cdot\left\|\nabla\left(div\left(D\cdot\overrightarrow{p_u^n}\right)+\hat{u}/\lambda\right)\right\|} \text{ (20), where } \tau=\frac{1}{4}$$

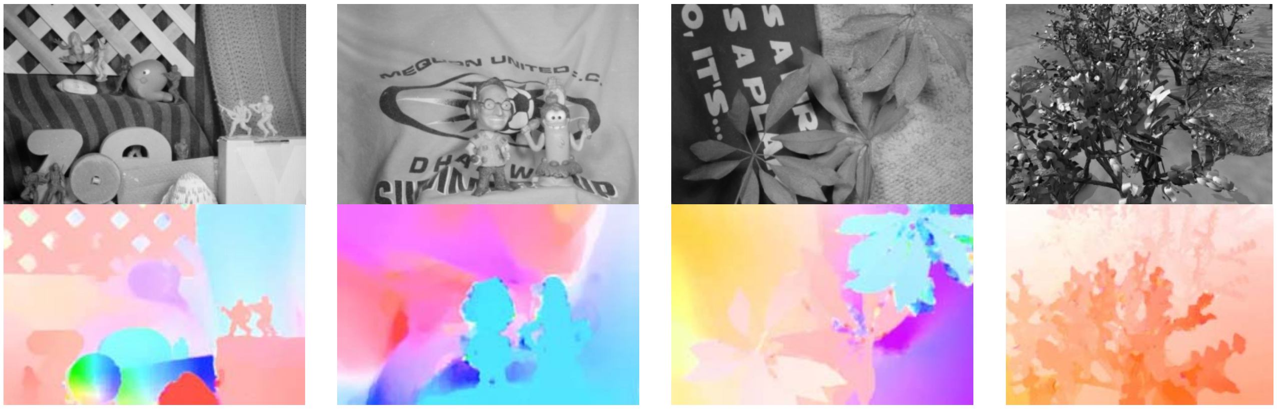
is the step width. Setting $D=1$, the numerical procedure match the one developed in [4]. The resulted numerical scheme is highly parallelizable.

III. Implementation

Since the approximation (2) is valid only for small displacements, a coarse-to-fine approach is employed. Beside large optical flow estimation, the coarse-to-fine approach improves the propagation from textured areas into weakly textured ones. For each level of the pyramid, a warping technique is used. For each warp the optical flow is computed using the equations (15), (18) and (20) in an iterative manner. The matrices u, v, $\overrightarrow{p_u}, \overrightarrow{p_v}$ are prolongated from each coarser level to the finer one. They are set to zero at the coarsest level. The next sequence depicts the pseudo-code of the method:

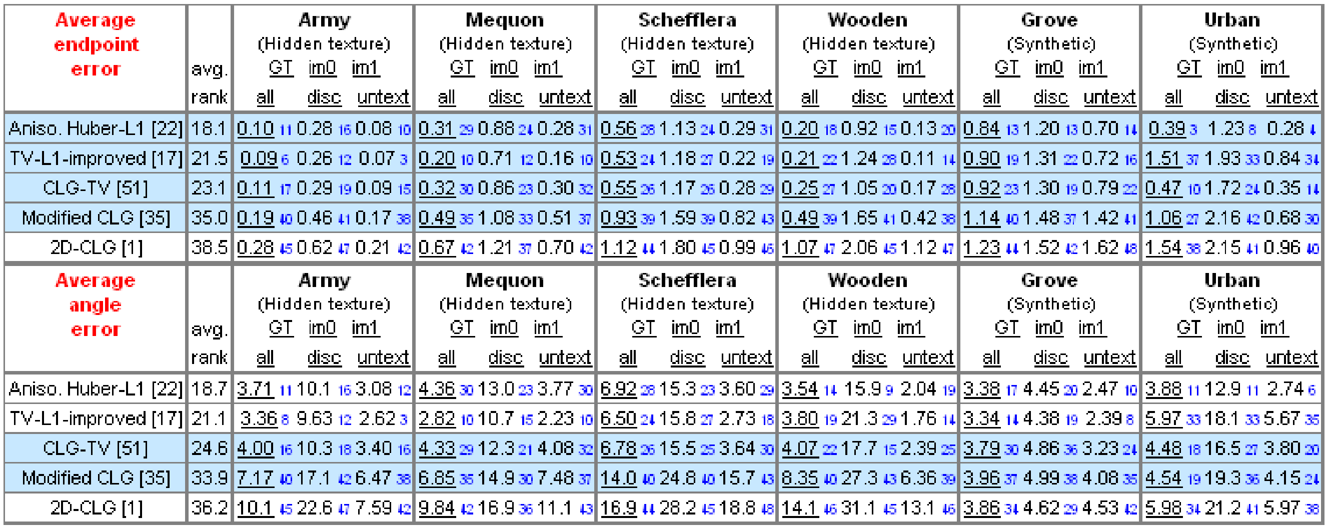
```
1) Set up pyramids $I_1, I_2$ and the their derivatives at each level
2) Computes the diffusion tensor $D$ for each level of the pyr. $I_1$.
3) For each level starting with the coarsest (below, each referred variable is at the current level)
   a) If the coarsest level, then initialize $u, v$, $\overrightarrow{p_u}, \overrightarrow{p_v}$ to zero.
   b) If not, upsample $u, v$, $\overrightarrow{p_u}, \overrightarrow{p_v}$ from the previous level to the current level.
   c) for i = 1 to warps (Apply the warping technique)
      i) Apply a median filter to $u, v$ in order to avoid strong outliers.
      ii) Warp $I_1, I_2, I_2^x, I_2^y$ using bilinear interpolation
      iii) For k=1 to eq_iterations
         A. Computes $\hat{u}, \hat{v}$ using (15). This step includes the computations of the coefficients using the bilateral filter. $I_1$ is used for the range gaussian.
         B. Update $u=\lambda\cdot div\left(D\cdot\overrightarrow{p_u}\right)+\hat{u}$)(18)
         C. Update $\overrightarrow{p_u}, \overrightarrow{p_v}$ using (20)
```

Typical values for the parameters are *warps = 5*, *eq_iterations = 10*. The pyramid factor is 0.5 and the number of levels depends on the image size. For 512x383 images, the number of levels can be six. For the diffusion filter $D(\|\nabla I\|)=e^{-\alpha\cdot\|\nabla I\|^\beta}$, $\alpha=5, \beta=1/2$ are very good values. The bilateral filter uses the first image I_1 to compute the range filter. In order to cope with large displacements, the implementation uses a 5x5 bilateral filter with $\sigma_s=5/6$ for the space gaussian and $\sigma_r=0.1$ (image intensities are in $[0,1]$) for the range gaussian. A 3x3 median filter is enough for the step 3)c)i). The data fidelity factor can be $\lambda=1000$ and $\theta=0.5$. The derivative operator (∇) is computed using forward differences, while the divergence operator (div) uses backward differences. The derivatives of the images are computed using central differences. The five points mask $[1,-8,0,8,-1]/12$ gives very good results.



a) Army b) Mequon c) Schefflera d) Grove

Fig. 3. Optical flow results of the CLG-TV method, on a part of the semi-synthetic Middlebury images



Average endpoint error	avg.	Army			Mequon			Schefflera			Wooden			Grove			Urban		
		(Hidden texture)			(Hidden texture)			(Hidden texture)			(Hidden texture)			(Synthetic)			(Synthetic)		
		GT	im0	im1	GT	im0	im1	GT	im0	im1	GT	im0	im1	GT	im0	im1	GT	im0	im1
	rank	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
Aniso. Huber-L1 [22]	18.1	0.10 11	0.28 16	0.08 10	0.31 29	0.88 24	0.28 31	0.56 28	1.13 24	0.29 31	0.20 18	0.92 15	0.13 20	0.84 13	1.20 13	0.70 14	0.39 3	1.23 8	0.28 4
TV-L1-improved [17]	21.5	0.09 6	0.26 12	0.07 3	0.20 10	0.71 12	0.16 10	0.53 24	1.18 27	0.22 19	0.21 22	1.24 28	0.11 14	0.90 19	1.31 22	0.72 16	1.51 37	1.93 33	0.84 34
CLG-TV [51]	23.1	0.11 17	0.29 19	0.09 15	0.32 30	0.86 23	0.30 32	0.55 26	1.17 26	0.28 29	0.25 27	1.05 20	0.17 28	0.92 23	1.30 19	0.79 22	0.47 10	1.72 24	0.35 14
Modified CLG [35]	35.0	0.19 40	0.46 41	0.17 38	0.49 35	1.08 33	0.51 37	0.93 39	1.59 39	0.82 43	0.49 39	1.65 41	0.42 38	1.14 40	1.48 37	1.42 41	1.06 27	2.16 42	0.68 30
2D-CLG [1]	38.5	0.28 45	0.62 47	0.21 42	0.67 42	1.21 37	0.70 42	1.12 44	1.80 45	0.99 46	1.07 47	2.06 45	1.12 47	1.23 44	1.52 42	1.62 48	1.54 38	2.15 41	0.96 40

Average angle error	avg.	Army			Mequon			Schefflera			Wooden			Grove			Urban		
		(Hidden texture)			(Hidden texture)			(Hidden texture)			(Hidden texture)			(Synthetic)			(Synthetic)		
		GT	im0	im1	GT	im0	im1	GT	im0	im1	GT	im0	im1	GT	im0	im1	GT	im0	im1
	rank	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
Aniso. Huber-L1 [22]	18.7	3.71 11	10.1 16	3.08 12	4.36 30	13.0 23	3.77 30	6.92 28	15.3 23	3.60 29	3.54 14	15.9 9	2.04 19	3.38 17	4.45 20	2.47 10	3.88 11	12.9 11	2.74 6
TV-L1-improved [17]	21.1	3.36 8	9.63 12	2.62 3	2.82 10	10.7 15	2.23 10	6.50 24	15.8 27	2.73 18	3.80 19	21.3 29	1.76 14	3.34 14	4.38 19	2.39 8	5.97 33	18.1 33	5.67 35
CLG-TV [51]	24.6	4.00 16	10.3 18	3.40 16	4.33 29	12.3 21	4.08 32	6.78 26	15.5 25	3.64 30	4.07 22	17.7 15	2.39 25	3.79 30	4.86 36	3.23 24	4.48 18	16.5 27	3.80 20
Modified CLG [35]	33.9	7.17 40	17.1 42	6.47 38	6.85 35	14.9 30	7.48 37	14.0 40	24.8 40	15.7 43	8.35 40	27.3 43	6.36 39	3.96 37	4.99 38	4.08 35	4.54 19	19.3 36	4.15 24
2D-CLG [1]	36.2	10.1 45	22.6 47	7.59 42	9.84 42	16.9 36	11.1 43	16.9 44	28.2 45	18.8 48	14.1 46	31.1 45	13.1 46	3.86 34	4.62 29	4.53 42	5.98 34	21.2 41	5.97 38

Fig. 4. Average end-point and angular errors of the selected methods

The proposed numerical solver of the CLG-TV model is highly parallelizable. The implementation of the current method using CUDA technology is straightforward. However, the GPU implementation uses a separable median filtering for a better performance. Generally, the median filter is not separable, but the separate version is enough for the step goal: to remove the strong outliers from the flow.

IV. RESULTS AND COMPARISONS

The CLG-TV method has two evaluations: one for synthetic and semi-synthetic images and the other for real images. The distinction is necessary because the CLG (combined local-global) methods are more suitable for real scenarios ([3],[15]). CLG-TV has been ranked on the Middlebury o. flow site (http://vision.middlebury.edu/flow/) and outperforms its CLG competitors both for endpoint error and for angular error. The proposed method is also compared with TV-L1 ([6]) and its improved version, Aniso. Huber-L1([7]), because all of them use [4] and [5] to derive a solver, are highly parallelizable and have approximately the same running time.

Fig. 4. shows the average end-point and angular errors. In order to reach high accuracy for Middlebury images, many warps and equation iterations should be used. The values *warps=35* and *eq_iterations=5* were used for this evaluation. The bilateral filter uses a window size of 3x3 instead of 5x5, since the images do not present very large displacements. The pyramid factor was 0.8. Also, the median filtering step is performed after each equation iteration (as a step D. inside the last "for" cycle).

The Middlebury benchmark [12] provides four real sequences for evaluation (Fig. 4). The interpolation quality measures the quality of the estimation. At the time of submission, CLG-TV was ranked 4th for interpolation error and 2nd for normalized interpolation error (Fig. 5). This result confirms a high quality estimation of the CLG-TV method for real scenarios. As seen in Fig. 5, Aniso. Huber-L1 and CLG-TV have nearly the same interpolation quality. However, these sequences do not have large displacements.

Fig. 6 presents common traffic scenarios. The cars have high speeds and they cause large displacements and large occlusions. In such situations, CLG-TV is clearly superior; Aniso. Huber-L1 fails to compute the correct flow. For Fig. 6 c) the flow might look correct for both methods, but an analysis of the horizontal flow for Aniso. Huber-L1 (in Fig. 6 d)2)) shows the contrary. The car moves about 50 pixels, but the method shows a motion of 30 pixels for its center. The backside flow is correctly determined. On the other hand, CLG-TV shows a motion of about 50 pixels for the

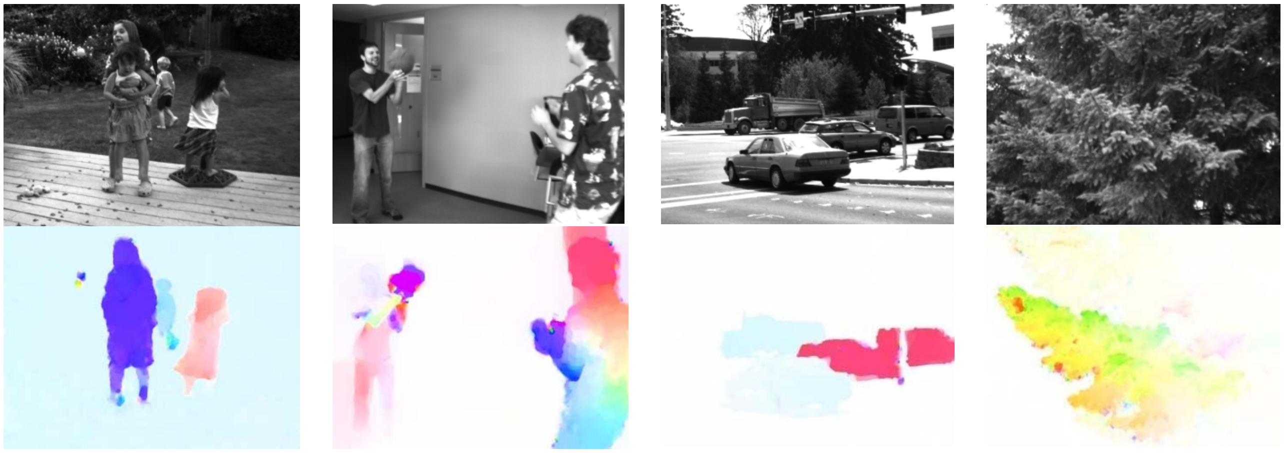
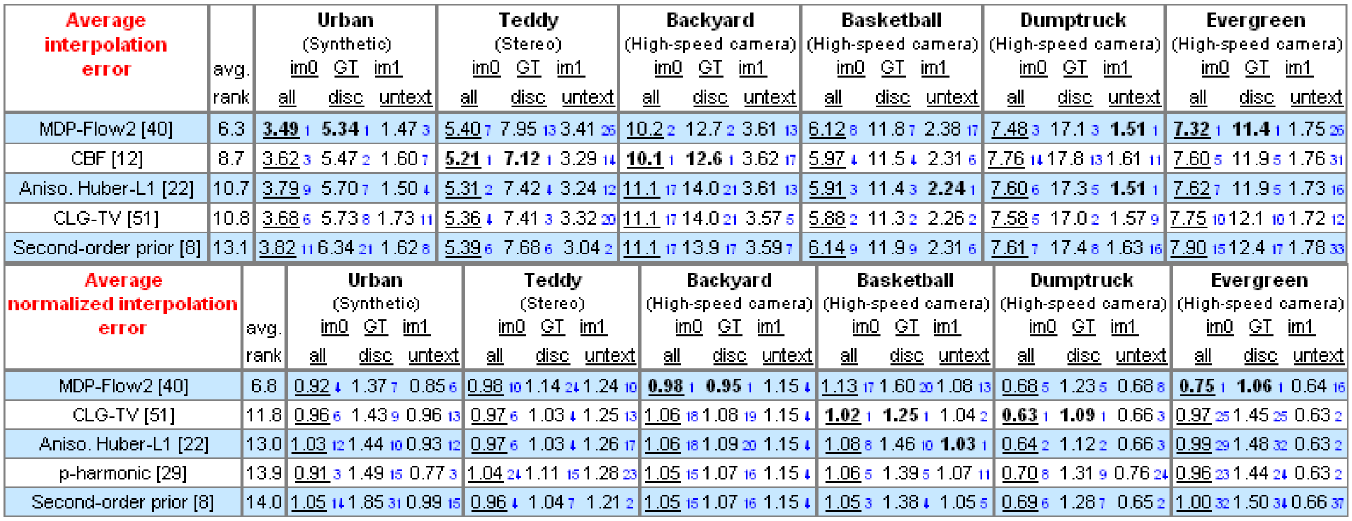


Fig. 4. Optical flow results of the CLG-TV method, on the Middlebury real images



Average interpolation error	avg.	Urban			Teddy			Backyard			Basketball			Dumptruck			Evergreen		
		(Synthetic)			(Stereo)			(High-speed camera)			(High-speed camera)			(High-speed camera)			(High-speed camera)		
		im0 GT im1			im0 GT im1			im0 GT im1			im0 GT im1			im0 GT im1			im0 GT im1		
	rank	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
MDP-Flow2 [40]	6.3	**3.49** 1	**5.34** 1	1.47 3	5.40 7	7.95 13	3.41 26	10.2 2	12.7 2	3.61 13	6.12 8	11.8 7	2.38 17	7.48 3	17.1 3	**1.51** 1	**7.32** 1	**11.4** 1	1.75 26
CBF [12]	8.7	3.62 3	5.47 2	1.60 7	**5.21** 1	**7.12** 1	3.29 14	**10.1** 1	**12.6** 1	3.62 17	5.97 4	11.5 4	2.31 6	7.76 14	17.8 13	1.61 11	7.60 5	11.9 5	1.76 31
Aniso. Huber-L1 [22]	10.7	3.79 9	5.70 7	1.50 4	5.31 2	7.42 4	3.24 12	11.1 17	14.0 21	3.61 13	5.91 3	11.4 3	**2.24** 1	7.60 6	17.3 5	**1.51** 1	7.62 7	11.9 5	1.73 16
CLG-TV [51]	10.8	3.68 6	5.73 8	1.73 11	5.36 4	7.41 3	3.32 20	11.1 17	14.0 21	3.57 5	5.88 2	11.3 2	2.26 2	7.58 5	17.0 2	1.57 9	7.75 10	12.1 10	1.72 12
Second-order prior [8]	13.1	3.82 11	6.34 21	1.62 8	5.39 6	7.68 6	3.04 2	11.1 17	13.9 17	3.59 7	6.14 9	11.9 9	2.31 6	7.61 7	17.4 8	1.63 16	7.90 15	12.4 17	1.78 33

Average normalized interpolation error	avg.	Urban			Teddy			Backyard			Basketball			Dumptruck			Evergreen		
		(Synthetic)			(Stereo)			(High-speed camera)			(High-speed camera)			(High-speed camera)			(High-speed camera)		
		im0 GT im1			im0 GT im1			im0 GT im1			im0 GT im1			im0 GT im1			im0 GT im1		
	rank	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
MDP-Flow2 [40]	6.8	0.92 4	1.37 7	0.85 6	0.98 10	1.14 24	1.24 10	**0.98** 1	**0.95** 1	1.15 4	1.13 17	1.60 20	1.08 13	0.68 5	1.23 5	0.68 8	**0.75** 1	**1.06** 1	0.64 16
CLG-TV [51]	11.8	0.96 6	1.43 9	0.96 13	0.97 6	1.03 4	1.25 13	1.06 18	1.08 19	1.15 4	**1.02** 1	**1.25** 1	1.04 2	**0.63** 1	**1.09** 1	0.66 3	0.97 25	1.45 25	0.63 2
Aniso. Huber-L1 [22]	13.0	1.03 12	1.44 10	0.93 12	0.97 6	1.03 4	1.26 17	1.06 18	1.09 20	1.15 4	1.08 8	1.46 10	**1.03** 1	0.64 2	1.12 2	0.66 3	0.99 29	1.48 32	0.63 2
p-harmonic [29]	13.9	0.91 3	1.49 15	0.77 3	1.04 24	1.11 15	1.28 23	1.05 15	1.07 16	1.15 4	1.06 5	1.39 5	1.07 11	0.70 8	1.31 9	0.76 24	0.96 23	1.44 24	0.63 2
Second-order prior [8]	14.0	1.05 14	1.85 31	0.99 15	0.96 4	1.04 7	1.21 2	1.05 15	1.07 16	1.15 4	1.05 3	1.38 4	1.05 5	0.69 6	1.28 7	0.65 2	1.00 32	1.50 34	0.66 37

Fig. 5. Top five methods for the interpolation and normalized interpolation errors

complete car surface. The explanation of this outcome is simple: the local window of the CLG-TV model spans better over the untextured zones and links them to the textured ones. Thus, beside representing the data fidelity, the local window contributes to the flow propagation (enhances the regulatization). For images in Fig. 6, the default settings were used (enabling real-time processing on the GPU). Aniso. Huber-L1 can solve some of these problems using different settings (like high accuracy), but these affects the running time.

The CUDA GPU implementation of the proposed numerical solver runs in 29 ms using the default settings. The images are 512x383 grayscale. The implementation was tested on an Intel Core 2 Duo machine running at 2.6 GHz, equipped with a NVidia GeForce GTX 285 graphics card.

V. CONCLUSIONS

The CLG (combined local-global) methods increase the robustness for scenes with large displacements, often encountered in traffic scenarios. In order to increase the accuracy, the CLG-TV method use a robust regularization based on total variation minimization. In addition, the proposed model is anisotropic and uses a bilateral filtering technique, combination that significantly improves the flow propagation. It reduces the propagation on the background and improves the fill-in process from textured areas to untextured ones. Overall, CLG-TV is a fast, accurate and robust optical flow estimator.

REFERENCES

[1] B.K.P. Horn and B.G. Schunck, “Determining optical flow,” Artificial Intelligence, 17:185–203, 1981.

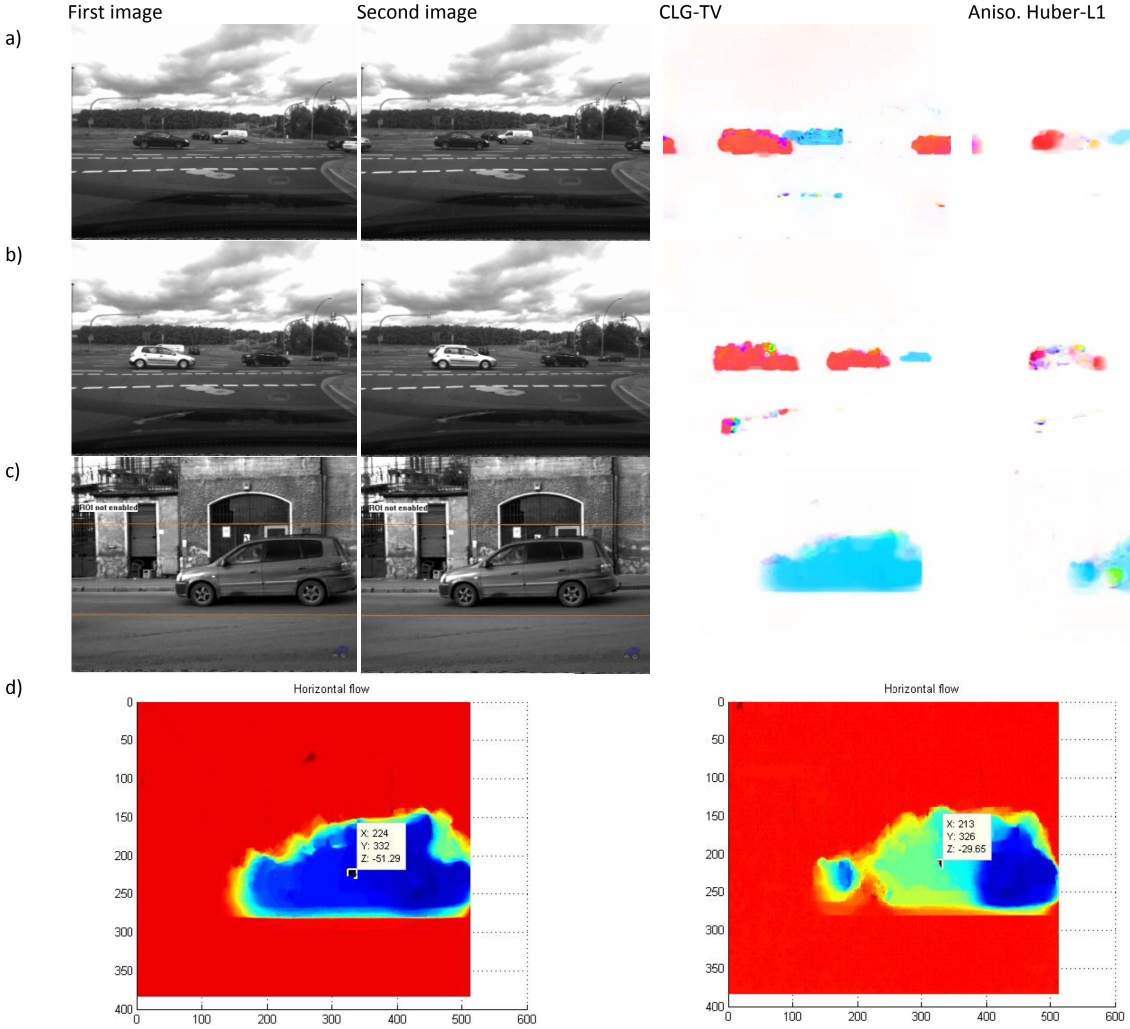
[2] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” In Proceedings of the 7th International Joint Conference on Artificial Intelligence, pages 674–679, April 1981.

[3] A. Bruhn, J. Weickert, and C. Schnorr, “Lucas/Kanade Meets Horn/Schunck: Combining Local and Global Optic Flow Methods,” In International Journal of Computer Vision 61(3), 211–231, 2005.

[4] A. Chambolle, “An algorithm for total variation minimization and applications”, Journal of Mathematical Imaging and Vision, 20(1–2):89–97, 2004.

[5] A. Chambolle, T. Pock, “A first-order primal-dual algorithm for convex problems with applications to imaging”, Journal of Mathematical Imaging and Vision (21 December 2010), pp. 1-26.

[6] C. Zach, T. Pock, and H. Bischof, “A duality based approach for realtime TV-L1 optical flow,” In Pattern Recognition, volume 4713 of LNCS, pages 214–223, 2007.



d)1) Horizontal flow for CLG-TV, c)

d)2) Horizontal flow for Aniso. Huber-L1, c)

Fig. 6. CLG-TV vs Aniso. Huber-L1 on traffic scenes with large displacements. The default setting were used (real-time)

[7] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof, “Anisotropic Huber-L1 optical flow,” In British Machine Vision Conference (BMVC), 2009.

[8] H.-H. Nagel and W. Enkelmann, “An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 8:565–593, 1986.

[9] L. I. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” Physica D, 60:259–268, 1992.

[10] R. T. Rockafellar. Convex analysis. Princeton Landmarks in Mathematics. Princeton University Press, Princeton, NJ, 1997. ISBN 0-691-01586-4. Reprint of the 1970 original, Princeton Paperbacks.

[11] P.Perona, J. Malik, “Scale space and edge detection using anisotropic diffusion,” Proc. IEEE Computer Soc. Workshop on Computer Vision (1987), 16–22.

[12] S. Baker, S. Roth, D. Scharstein, M. Black, J. Lewis, and R. Szeliski, “A database and evaluation methodology for optical flow,” in International Conference Computer Vision, 2007, pp. 1–8.

[13] Frédo Durand and Julie Dorsey, “Fast bilateral filtering for the display of high-dynamic-range images”, In SIGGRAPH ’02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques, pages 257–266, New York, NY, USA, 2002. ACM.

[14] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images”, In Proceedings of the Sixth International Conference on Computer Vision, ICCV ’98, Washington, DC, USA, 1998. IEEE Computer Society.

[15] Marius Drulea, Ioan Radu Peter, Sergiu Nedevschi, "Optical flow. A combined local-global approach using L1 norm," ICCP, pp.217-222, Proceedings of the 2010 IEEE 6th International Conference on Intelligent Computer Communication and Processing, 2010.