IPSJ Transactions on Computer Vision and Applications Vol.6 98–102 (Aug. 2014)

[DOI: 10.2197/ipsjtcva.6.98]

Express Paper

Video Completion via Spatio-temporally Consistent Motion Inpainting

Menandro Roxas[1,a)] Takaaki Shiratori[2,b)] Katsushi Ikeuchi[1,c)]

Received: April 25, 2014, Accepted: May 19, 2014, Released: August 4, 2014

Abstract: Given an image sequence with corrupted pixels, usually big holes that span over several frames, we propose to complete the missing parts using an iterative optimization approach which minimizes an optical flow functional and propagates the color information simultaneously. Inside one iteration of the optical flow estimation, we use the solved motion field to propagate the color and then use the newly inpainted color back to the brightness constraint of the optical flow functional. We then introduce a spatially dependent blending factor, called the mask function, to control the effect of the newly propagated color. We also add a trajectory constraint by solving the forward and backward flow simultaneously using three frames. Finally, we minimize the functional by using alternating direction method of multipliers.

Keywords: video completion, optical flow

1. Introduction

Video inpainting or completion is the process of recovering missing parts of videos either by interpolation or duplication of the known parts. Missing parts or holes could come from damage such as in old vintage videos, missing edges due to frame alignment, intermediate frames in frame interpolation, and object removal. The goal of inpainting is a visually pleasing output video that is both spatially and temporally consistent. Spatial consistency requires objects to maintain their geometry while temporal consistency requires parts of the same object to move in the same manner.

Numerous methods have been formulated to solve the video inpainting problem (Refs. [7], [8], [9], [16], [17], [18] and [19] among others). Some work directly extended image inpainting methods to videos. With the addition of a third dimension (time), these methods result in poorly inpainted sequence especially when both background and holes are moving. Non-parametric sampling by Wexler et al. [23] use global spatio-temporal optimization and 3D patches to fill in holes. Jia et al. [9] use large fragments based on color similarity instead of using fixed size patches and use tracking to complete the video. Some methods use frame alignment using features (low-rank [25], SURF [5], etc.) with variants such as separately inpainting background and foreground using layers [8].

One particular approach is the use of optical flow to propagate pixels with known colors toward the hole. This approach is straight-forward, as long as the optical flow is available in the immediately succeeding or preceding frame. However in most practical cases, such as removing pedestrians in street videos, the holes extend several frames which makes immediate copying of colors using motion information insufficient. To solve this problem, one approach is to also estimate the motion inside the hole using similarity measures. Shiratori et al. [17] utilize fixed size patch motion fields instead of colors. Tang et al. [18] also inpaint motion but use weighted priority of patches to select the best-match patch.

Video inpainting can benefit from frame interpolation methods that use motion inpainting [2], [22]. The difference between the two problems is the unavailability of spatial information in the latter. Instead of exclusively interpolating the trajectory, color and motion consistency assumption at the boundary of the hole could be used to improve the inpainting results. Werlberger et al. [22] used optical flow to estimate the velocity of pixels between two consecutive frames and applied a TV-L1 denoising algorithm to inpaint holes. However, in their method, the solution for optical flow and inpainting are separately done.

In our method, we use an iterative optimization approach that uses optical flow to complete the color frames. Our method relies heavily on the accurate estimation of the motion fields at the boundary of and inside the hole, therefore we improve the standard optical flow by adding a trajectory constraint and introducing a spatially dependent mask function. Compared to existing methods, we solve the inpainting problem in a unified approach by simultaneously removing an object, inpainting the background motion, and completing the color frames, all within our iteration method.

2. Optimization Function

Given an image sequence with brightness values $\mathbf{I}$, n frames $\{n|n \in \mathbf{Z}, 0 \leq n \leq N\}$ and a hole $\mathbf{H} \subseteq \mathbf{I}$, we define $\mathbf{u_{n,n+k}}$ as

[1] The University of Tokyo, Meguro, Tokyo 153–8505, Japan
[2] Microsoft Research Asia, Haidian District, Beijing 100080, China
[a)] roxas@cvl.iis.u-tokyo.ac.jp
[b)] takaakis@microsoft.com
[c)] ki@cvl.iis.u-tokyo.ac.jp

the optical flow between frames I_n and I_{n+k} where $k = \{-1, 1\}$. Our objective is to find 1) $\mathbf{u_{H,n,n+k}} = (u_{H,n,n+k}, v_{H,n,n+k})$ between $\mathbf{H_n}$ and $\mathbf{H_{n+k}}$, where u and v are the horizontal and vertical motion vectors, and; 2) the brightness $\mathbf{H_n}$ inside the hole for all n. The method that will be presented here can be easily extended to colored frames, hence from now on, we will use the term color instead of brightness. For simplicity reasons, we will explain only one segment of the sequence $\{n-1, n, n+1\}$ therefore, we will use the subscript $\{b, 0, f\}$ or the backward, reference, and forward (flow) for those frames. We will also let $\mathbf{x}$ as the 2D position (x, y).

Our main contribution is the use of an iterative method to solve the optical flow and color. We do this by alternately solving an optimization function Eq. (1) and propagating the color information using the solved optical flow. We generalize the optimization function as:

$$\min_{\mathbf{u_f}, \mathbf{u_b}} (E_{data} + E_{spatial} + E_{trajectory})(\mathbf{u_f}, \mathbf{u_b}) \tag{1}$$

where E_{data}, $E_{spatial}$ and $E_{trajectory}$ are the energy terms which we will elaborate in the following sections. It is important to note that we solve the forward and backward flows simultaneously among three frames. From this point on, we will tackle the details of each of the terms that are essential in our method.

2.1 Data Term

The data term consists of two parts: the mask function $m(\mathbf{x})$ and the optical flow brightness constancy terms [6] shown in Eq. (2).

$$E_{data} = m(\mathbf{x}) \left[\varphi \left(I_f(\mathbf{x} + \mathbf{u_f}) - I_0 \right) + \varphi \left(I_b(\mathbf{x} + \mathbf{u_b}) - I_0 \right) \right] \tag{2}$$

We use the differentiable approximation of the L_1 data penalty $\varphi(s) = \sqrt{s^2 + \epsilon^2}$ where ϵ is an arbitrary small constant. Our data term is special due to the mask function which separates the effect of the color values into two ways. At the boundary of the hole, we give the mask a positive value (highest), which allows the known color to have more influence in the minimization of Eq. (1). Inside the hole, we give the mask function a lower value to control the effect of the newly inpainted color values. It is essential to elaborate first the two remaining energy terms in order to fully explain how the mask function works, therefore, we will suspend this discussion until Section 2.4.

2.2 Spatial Smoothness Constraint

To allow the flows to be spatially smooth, we implement the spatial energy term as the widely used Total Variation regularizer shown in Eq. (3).

$$E_{spatial} = \lambda_1 (|\nabla \mathbf{u_f}|_{TV} + |\nabla \mathbf{u_b}|_{TV}) \tag{3}$$

The TV term is discretized as

$$|\nabla \mathbf{u}|_{TV} = \sum_i \sqrt{(\nabla_x u)^2 + (\nabla_y u)^2} \tag{4}$$

where ∇_x and ∇_y is the gradient in the x and y direction, respectively, and λ_1 is the blending constant.

2.3 Smooth Trajectory Constraint

In practical applications, smooth trajectory constraint applies more appropriately compared to temporal smoothness. Temporal smoothness [11], [26] assumes that the motion in one pixel position of two frames is the same (or smooth if more than one frame). This constraint holds only if the pixels in two different frames belong to the same object (hence, similar motion). On the other hand, smooth trajectory constraint ensures that real-world points register smooth motion in the image frame.

Several works use smooth trajectory as an additional constraint to the optical flow functional. Werlberger et al. [21] solve the optical flow using three frames by imposing a hard constraint between the forward and the backward flow. Since they assume that both motion is equal which is not always true, the authors reported a degradation in the result on some of their data. Salgado et al. [15] impose a soft constraint between the flows however, their method require warping of the flows to each other, which makes it difficult to solve because flow fields refer to many different coordinate frames. Volz et al. [20] on the other hand, solve the flow fields with respect to one reference frame thus removing the need to warp them. In their method, the authors used multiple frames and only one direction which makes solving the trajectory simpler. We reformulate Volz method, but instead we use only three frames and solve both the backward and forward flow. We then impose the trajectory smoothness as a soft constraint as shown below:

$$E_{trajectory} = \lambda_2 \phi(\mathbf{u_f}, \mathbf{u_b}) \tag{5}$$

where λ_2 is a positive constant and $\phi(\mathbf{u_f}, \mathbf{u_b}) = |\mathbf{u_f} - \mathbf{u_b}|_2^2$ is the quadratic penalty function.

The term $\mathbf{u_f} - \mathbf{u_b}$ refers to the similarity measure between $\mathbf{u_f}$ and $\mathbf{u_b}$. The measure could be angular difference or end-point error [1]. To simplify the computation, we implemented the constraint using constant velocity assumption as used in Ref. [15].

The usage of only three frames in our method results in the implicit propagation of information among succeeding frames with the use of the mask function. Similarly, using three frames allows for the strong dependency of the hole on immediate frames and the diffused effect of color information from distant frames. While it is true that using more frames will improve the estimation of the trajectory, it does not always hold in our case since we are dealing with holes that span several frames. In other words, the solved trajectory will have less effect on the accuracy of the inpainted motion compared the propagated color from an immediate neighboring frame.

2.4 Mask Function

The mask function $m(x)$ allows us to control the effect of inpainted color pixels on the estimation of the motion inside the hole. We generalize $m(x) = [0, 1]$. Outside the boundary of the hole, the color values should have maximum influence on the solution of Eq. (1), therefore we set $m(x) = 1.0$ for all x in $I \setminus H$. Inside the hole, we set $m(x)$ as the function of the distance of the pixel's source wrt. the reference frame.

We give more weight on inpainted pixels that come from immediate frames. It follows that as the source of the color comes

from more distant frames, the motion relies more on the solved trajectory and spatial smoothness. The mask function needs to be controlled efficiently, because adding it to the optimization function makes the latter unstable. Instead of explicitly solving $m(x)$, we define it as a constant during one iteration and updates it after that.

3. Solution to Optimization Problem

To make this paper self-sufficient, we describe the solution to Eq. (1) including the color propagation and iteration stages in detail.

3.1 Linear Warping and Coarse-to-fine Approach

We linearize or warp the image frame I_f using the first order Taylor approximation near $x + u_0$, where u_0 is the solved optical flow after an iteration (or zero at initialization). We use linear interpolation to solve the value of $I_f(x + u_0)$. Assuming that u_0 is a good approximation of u_f, we can say that the brightness constancy term $I_f(x + u_0) - I_0) \approx 0$. The same holds for I_b.

To further impose the above assumptions, we implement a coarse-to-fine strategy by building image pyramids. We do this by repeatedly down-sampling the image frames by a factor of α. Using this approach, we can compensate for large pixel motions that is usually present our videos. We use $\alpha > 0.5$ so that each of the succeeding pyramid is a blurred version of the lower level.

3.2 Alternating Direction Method of Multipliers

To solve for the backward and forward flow simultaneously, we implement the split Bregman method [4] which is a variant of ADMM. We define an iteration variable $b^{k+1} = b^k + \phi(\mathbf{u_f}, \mathbf{u_b})$ to decouple $\mathbf{u_f}$ and $\mathbf{u_b}$. Combining Eqs. (2), (3) and (5), we get the resulting minimization function:

$$\min_{\mathbf{u_f},\mathbf{u_b}} m(x)\left[\varphi\left(I_f\left(\mathbf{x}+\mathbf{u_f}\right)-I_0\right)+\varphi\left(I_b(\mathbf{x}+\mathbf{u_b})-I_0\right)\right] + \lambda_1[|\nabla \mathbf{u_f}|_{TV} + |\nabla \mathbf{u_b}|_{TV}] + \lambda_2 \left|\phi(\mathbf{u_f}, \mathbf{u_b}) - b_k\right|_2^2 \quad (6)$$

We first solve $\mathbf{u_f}$ by holding $\mathbf{u_b}$ constant and vice versa. We then update the iteration variable b^k. For each variable, we minimize the following function.

$$\min_{\mathbf{u_f}} m(x)\varphi\left(I_f(x+\mathbf{u_f})-\mathbf{I_0}\right)+\lambda_1\left|\nabla \mathbf{u_f}\right|_{TV} + \lambda_2\left|\phi(\mathbf{u_f},\mathbf{u_b})-\mathbf{b_k}\right|_2^2 \quad (7)$$

To minimize Eq. (7), we use the iterative method described by Papenberg et al. [13]. We call this step as the inner iteration and elaborate it in Algorithm 1.

3.3 Iterative Stage

The algorithm of the iterative stage is shown in Algorithm 2 and is detailed in the following sections.

3.3.1 Initialization of u

During the first iteration, the hole does not have any information. Therefore the mask function inside the hole is set to zero. This initialization method is similar to Ref. [14] instead we added a trajectory smoothness constraint. In some cases, the initial value is very close to the final output especially when the surface is flat and there are no obvious occlusion boundaries.

Algorithm 1 Inner Iteration

Require: $\mathbf{u_f}, \mathbf{u_b}$
  initialize $\mathbf{u_f}, \mathbf{u_b}, \mathbf{b^0}, \mathbf{k} \leftarrow 0$
  while convergence≠TRUE **do**
    linearize I_f, I_b
    Hole Warping
    $u_b = constant$, solve u_f
    $u_f = constant$, solve u_b
    update b^{k+1}
    $k \leftarrow k + 1$
  end while

Algorithm 2 Our Proposed Method

Require: color of H
  solve $image\_pyramids$
  initialize $m(x \in H) = 0$
  for $level <$ max $\_level$ **do**
    while $error > thresh$ **do**
      Inner Iteration
      Color Propagation
      update $m(x)$
    end while
    upsample u_f, u_b
  end for

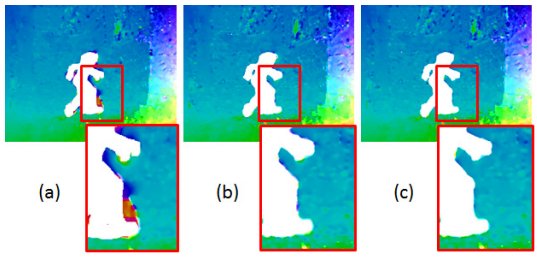

Fig. 1 Optical flow at the boundary of the hole (a) without removing the object prior to estimation; (b) without warping the hole of the next frame to the reference frame; (c) with hole warping.

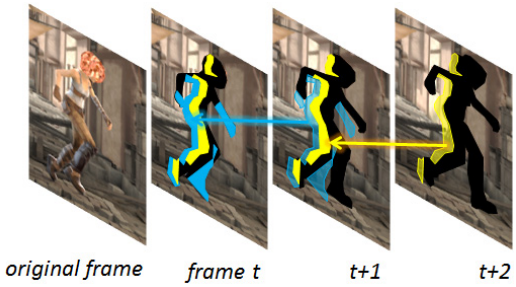

Fig. 2 Color propagation.

3.3.2 Hole Warping

One problem that is apparent in estimating optical flow is the effect of occluding objects. There are two cases where occluding objects will degrade the result of motion inpainting and therefore the overall result. The first case occurs when the removed object is at the foreground and the inpainted region is the background. If we solve the optical flow before removing the object, the error at the boundary will remain in the final solution (**Fig. 1** (a)). Therefore, motion inpainting methods that rely on spatial smoothness [14] as well as non-parametric sampling methods that uses motion fields [17], [18] will fail.

The second case occurs when the region to be estimated is also

occluded on the succeeding frame by the same foreground object (Fig. 1 (b)). The only case where this does not happen is when the background moves and the foreground object remains in the same position with respect to the camera frame (i.e., raindrop removal [24]).

The first problem is solved implicitly in our method because we remove the object prior to the motion estimation stage. The second problem is solved by warping the hole from the succeeding frame to the reference frame using the estimated flow in one iteration. We use the same interpolation technique we used for the color frames but with nearest neighbor approach. The improvement in optical flow estimation using this technique is shown in Fig. 1 (c).

3.3.3 Color Propagation

Given $\mathbf{u_f}$ and $\mathbf{u_b}$ of all the frames, we propagate the colors by following the motion to another frame. This is a straight-forward method except when the color is not available on the succeeding frame. We illustrate our approach to this problem in **Fig. 2**. Given the hole at frame t, we follow the forward flow of the hole to the next frame $t + 1$. In this frame, the pixels in blue are available, therefore we directly copy them to frame t using the same warping technique in Section 3.1. We then follow the flow of the remaining hole to frame $t + 2$ and this time, the pixels in yellow become available. We copy this to frame $t + 1$ and on to frame t. At this point, we give the a weight μ to each inpainted pixels in frame t proportional to the distance of their source to the reference frame ($\mu = 1$ for blue pixels, $\mu = 2$ for yellow pixels). We do this approach in both directions until all the pixels are inpainted, under the assumption that all pixels are visible at any of the available frames.

By giving the inpainted pixels some weight μ, we can then control how much effect it has relative to the spatial and trajectory smoothness. We use this weight directly as the new value of the mask and perform the iterative step of our method. The mask inside the hole is updated as $m(x) = \gamma^{-\mu}$ similar to the weighting parameter used in Refs. [3] and [23].

4. Experimental Results

In **Fig. 3**, we show first the improvement in the inpainted motion by using the newly inpainted pixels compared to a fixed mask function (zero inside the hole). The result suggests that the smoothing effect of the spatial and temporal constraint is regulated resulting in a more detailed optical flow especially at the object boundaries.

Next, we show the effectiveness of our method using an image sequence (**Fig. 4**) taken using a hand-held camera with minimum motion (only camera shaking). We introduce a moving box as noise and inpaint the damaged part to recover the original video. We would like to demonstrate that under track-able foreground motions (real world motion) and static background, temporal aliasing (ghosts) can be easily noticed. In our result, we successfully inpainted the video sequence with little noticeable motion. We compare the inpainted frames to the original ones and solved the Mean Absolute Error $MAE = \frac{1}{i}\sum_i \left|I_{output} - I_{orginal}\right|$ to

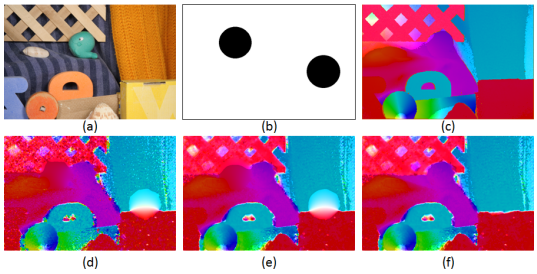

Fig. 3 Effect of using newly inpainted colors in motion inpainting of 'Rubberwhale' sequence [12]. (a) Color frame (b) Initial mask (c) Ground optical flow (d) Inpainted without trajectory constraint and fix mask (m = 0) (e) Inpainted with trajectory constraint (f) Inpainted with our mask function.

Fig. 4 Results of inpainting on 'room' sequence. Top row: masked input. Bottom row: inpainted output.

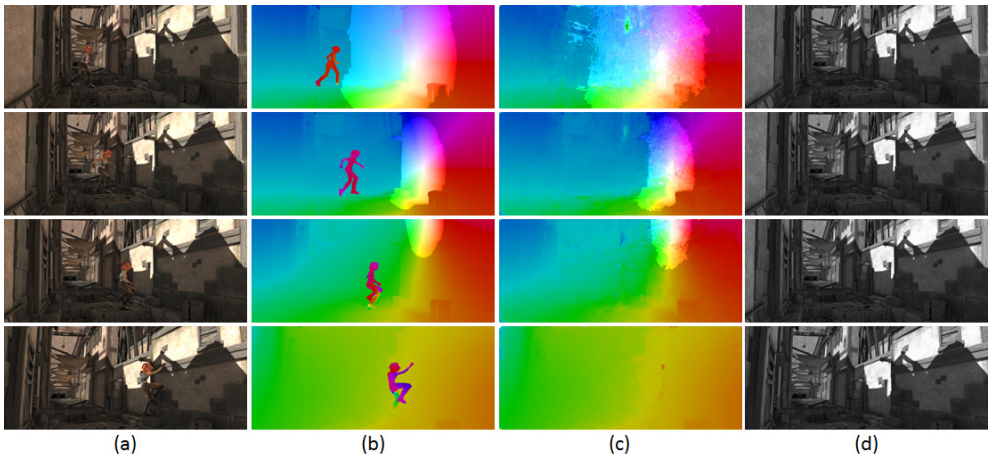

Fig. 5 Inpainting result on 'alley' sequence. (a) Input frames (b) Ground-truth optical flow (c) Solved and inpainted optical flow using our method (d) Inpainted gray image sequence.

Fig. 6 Inpainting result of street video. Top row: input sequence. Bottom row: removed pedestrian and inpainted sequence.

be 1.08 pixel intensity units.

We also tested our method on synthetic image sequences [10]. We use the first 30 frames of the 'alley' sequence in Ref. [10] and remove the running girl. We show the representative frames in **Fig. 5**.

Finally, we tested our method on a rectified and stabilized street video (**Fig. 6**) taken from one side of an omnidirectional camera (the reason for the rotated appearance). We used 62 frames for this sequence and remove the walking pedestrians. We successfully inpainted the video including the parts that are represented only on few frames due to occlusion (ground).

5. Conclusion and Future Work

In this paper we have demonstrated the efficiency of our method in inpainting videos. Several improvements can be done. First, using spatio-temporal pyramid could increase the efficiency of motion estimation especially during cases where severe occlusion and plane homography occurs. Second, since we used split Bregman method, the penalty functions could be improved to make each iteration faster. Finally, the initialization of **u** could be improved so that the solution can converge faster.

Acknowledgments This work is, in part, supported by Monbukagakusho (MEXT) project.

References

[1] Barron, J. et al.: Performance of Optical flow Techniques, *IJCV*, Vol.12, No.1, pp.43–77 (1994).
[2] Chen, K. and Lorenz, D.: Image Sequence Interpolation using Optimal Control, *Journal of Mathematical Imaging and Vision*, Vol.41, pp.222–238 (2011).
[3] Criminisi, A., Perez, P. and Toyama, K.: Object Removal by Exemplar-Based Inpainting, *Proc. IEEE CVPR '08*, Vol.2, pp.721–728 (2008).
[4] Goldstein, T. and Osher, S.: The Split Bregman Method for L1 Regularized Problems, *Journal of Scientific Computing*, Vol.45, No.1-3, pp.272–293 (2010).
[5] Granados, M. et al.: Background Inpainting for Videos with Dynamic Objects and a Free-moving Camera, *ECCV '12 Proceedings* (2012).
[6] Horn, B. et al.: Determining Optical Flow, *Artificial Intelligence*, Vol.17, pp.185–203 (1981).
[7] Jia, J., Tai, Y.W., Wu, T.P. and Tang, C.K.: Video Repairing under Variable Illumination using cyclic motions, *IEEE TPAMI*, Vol.28, No.5, pp.832–836 (2006).
[8] Jia, J. et al.: Video Repairing: Inference of Foreground and Background Under Severe Occlusion, *CVPR '04 Proceedings* (2004).
[9] Jia, Y.T. et al.: Video Completion using Tracking and Fragment Merging, *The Visual Computer*, Vol.21, pp.601–611, Springer-Verlag (2005).
[10] MPI Sintel Dataset, available from ⟨http://sintel.is.tue.mpg.de/⟩.
[11] Nagel, H.: Extending the 'Oriented Smoothness Constraint' into the Temporal Domain and the Estimation of Derivatives of Optical Flow, *ECCV '90 Proceedings* (1990).
[12] Optical Flow - The Middlebury Computer Vision Pages, available from ⟨http://vision.middlebury.edu/flow/data/⟩.
[13] Papenberg, N. et al.: Highly Accurate Optic Flow Computation with Theoretically Justified Warping, *Intl. Journal of Comp. Vision*, Vol.67, pp.141–158, Springer Science (2006).
[14] Roxas, M. et al.: Video Completion via Maintaining Spatially Consistent Motion, *Meeting on Image Recognition and Understanding '13* (2013).
[15] Salgado, A. et al.: Temporal constraints in large optical flow estimation, *Computer Aided Systems Theory-EUROCAST '07*, Vol.4739, pp.709–716, Springer, Berlin (2007).
[16] Shen, Y., Lu, F., Cao, X. and Foroosh, H.: Video Completion for Perspective Camera under Constrained Motion, *IEEE ICPR '06 Proceedings* (2006).
[17] Shiratori, T. et al.: Video Completion by Motion Field Transfer, *IEEE CVPR'06 Proceedings* (2006).
[18] Tang, N.C. et al.: Video Inpainting on Digitized Vintage Films via Maintaining Spatio-temporal Continuity, *IEEE Trans. Multimedia*, Vol.13, No.4, pp.602–614 (2011).
[19] Tsai, T.H. et al.: Text-Video Completion using Structure Repair and Texture Propagation, *IEEE Trans. Multimedia*, Vol.13, No.1, pp.29–39 (2011).
[20] Volz, S. et al.: Modeling Temporal Coherence for Optical Flow, *ICCV '11* (2011).
[21] Werlberger, M. et al.: Motion estimation with non-local total variation regularization, *IEEE CVPR '10 Proceedings*, San Francisco, CA (2010).
[22] Werlberger, M. et al.: Optical flow guided TV-L1 video interpolation and restoration, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, Springer Berlin Heidelberg (2011).
[23] Wexler, Y. et al.: Space-time Completion of Video, *IEEE TPAMI*, Vol.29, No.3, pp.463–476 (2007).
[24] You, S. et al.: Adherent Raindrop Detection and Removal in Video, *CVPR '13*, Portland, Oregon (2013).
[25] Zhang, Z. et al.: TILT: Transform Invariant Low-rank Textures, *ACCV '10 Proceedings* (2010).
[26] Zimmer, H. et al.: Optic Flow in Harmony, *IJCV*, Vol.93, No.3, pp.368–388 (2011).

(Communicated by *Keiji Yanai*)