Spatio-Temporal Optimization-Based Motion Inpainting for Video Completion

Menandro Roxas
The University of Tokyo
Tokyo, Japan
roxas@cvl.iis.u-tokyo.ac.jp

Takaaki Shiratori
Microsoft Research Asia
Beijing, China
takaakis@microsoft.com

Katsushi Ikeuchi
The University of Tokyo
Tokyo, Japan
ki@cvl.iis.u-tokyo.ac.jp

Abstract

In this thesis, we propose a method to complete damaged videos using motion inpainting (MI) and color propagation (CP). We first constraint the interpolated motion of the regions to be completed, called holes, to be spatio-temporally smooth. To achieve this, we simultaneously solve and interpolate the motion of the known regions and the hole by minimizing an optical flow estimation function with the spatio-temporal smoothness constraints. Then, we use the solved optical flow to propagate the color of the known pixels to the hole using bicubic warping. We embed this two-step (MI+CP) method in an iterative optimization framework where we propose to use the newly inpainted color to further improve the optical flow estimation. This is done by introducing a spatially varying mask function that is dependent on the frame distance of the source of the inpainted color. We also propose a method to accurately impose the temporal smoothness constraint by solving a trajectory prior based on the camera's egomotion. We propose a fast estimation of the translation parameters through points correspondence among only three frames. Finally, we test our method in synthetic and real videos as well as urban street videos taken from a moving vehicle.

1 Introduction

Recent advances in the field of transportation, navigation, and virtual reality have caused the emergence of video cameras that are used to capture an urban environment. These cameras are mounted in different places around a vehicle such that they can view the scene of particular interest. For example, cameras that are mounted on top of cars are used to document a cityscape for applications such as virtual tours, 3D modeling, digital archiving, and driving simulation.

Several issues arise from the use on-vehicle video cameras that could cause problems in certain applications. For example, the presence of pedestrians in videos that are used for virtual tours and digital archiving pose privacy issues especially when the faces are clear enough to be recognized. This issue is very common in crowded places such as tourist spots or streets. A simple and common solution to address this problem is to blur or blackout the people's faces. However, in some applications, simple blurring is not enough especially because it removes the visual appeal of the video. Oftentimes, a complete view of the facade of a building is also necessary and a complete removal of pedestrians is needed.

Another issue is the presence of artifacts such as dead or corrupt pixels that are inherent to the camera. Some are adherent water or smudges and occluding objects on the lens of the camera or on the windshield of a car. All of these artifacts degrade the quality of the video and are problematic when used in applications that require a clear view of the surrounding.

We argue that the best solution to these issues is to completely delete the information (color pixels) containing the unwanted artifacts and redraw or replace them with the desired pixels. Although this could be done manually frame-by-frame using any image/video editing software, the process requires accuracy and time.

In this thesis, we call this process as video completion. In the succeeding section, we will define the video completion problem and its objectives and we will follow with the main contribution of this work in addressing this problem.

1.1 Video Completion Problem

Given an image sequence $\mathbf{S}$, we define the deleted region (removed pedestrians/artifacts) as the hole $\mathbf{H}$. The completion process fills in $\mathbf{H}$ with information from the known parts $\mathbf{\Omega} = \mathbf{S} \setminus \mathbf{H}$ of the sequence. The main objective of video completion is to find an $\mathbf{H}$ that makes $\mathbf{S}$ visually pleasing.

In our criteria, a visually pleasing video should have spatial and temporal consistency in both color and motion domains.

- **Spatially Consistent Color.** An object that appears to be geometrically improbable (floating objects, curved building walls) is undesired. In video completion, a recognized object must satisfy its geometrical definition (i.e. a building must have doors, and windows, and its walls must be smooth.), therefore the completed parts must adhere to the original structure. Any divergence from its preconceived appearance is easily recognized by the viewer as an inconsistency.

- **Temporally Consistent Color.** If an object appears in one frame, then it should appear in all frames unless it is occluded by another object. Violation of this objective results in flickering of objects where it appears and disappears abruptly.

- **Spatially Consistent Motion.** This criterion constraints the motion of the points belonging to a same object to be smooth. Ideally, we want the motion of the hole and the boundary to be smooth such that the edge of the hole will not be apparent in the resulting video. It also suggests that the motion of the points inside the hole must be smooth.

- **Temporally Consistent Motion.** If we track a point among several frames, the motion of that point must be smooth. Although this criterion is violated in shaky videos, it still constraints the motion of a point to the general motion of the shaking that comes from the camera motion.

Using these criteria, we review some of the most recent techniques in video completion and propose our method in the succeeding sections.

1.2 Related Work

Video completion methods can be divided into two categories: using color or brightness frames and using optical flow or motion frames. Methods that use color or brightness frames rely on a similarity measure between pixels while methods that use optical flow frames rely on the similarity of motion of neighboring pixels.

1.2.1 Using Color Frames

Numerous methods have been formulated to solve the video completion problem. Some work directly extended image inpainting methods [14] to videos. With the addition of the time dimension, these methods result in poorly inpainted sequence especially when the background and holes are both moving.

Non-parametric sampling is the most famous video inpainting method. A global spatio-temporal optimization is proposed by Wexler et al [13]. They use 3D patches including RGB channel and intensity gradient in the horizontal and vertical directions. With the use of 3D patches the authors claim a solution in the temporal discontinuity that result from extended image inpainting techniques. However, this method suffers in both accuracy and efficiency when the hole becomes very big and the inpainted background become large.

Jia et al [15] propose an extension of the previous method and use large fragments based on color similarity instead of using fixed size patches and use tracking to complete the video. A large improvement in time was achieved but the quality of the inpainting is still the same. Another extension that solve the time complexity of patch matching is [16] which allows a person to indicate locations in the video that the source of the hole might come from. In this way, the search space was dramatically reduced and the completion time was improved.

Some methods use frame alignment using features (low-rank [18], SURF [17], etc.) with variants such as separately inpainting background and foreground using layers [19]. However, frame alignment only works with planar scenes and with very few visible planes. These methods also require that objects have the same size throughout the video and therefore is not applicable in most applications. Moreover, detecting planes become problematic when there are multiple planes that affect the desired value of the hole.

The most common issue with these methods is the absence of an explicit motion constraint. The success of the inpainting results depends solely on the effective comparison between neighboring pixels and the source patches. Consistent motion are somehow achieved by using 3D patches (including neighboring frames), however this approach relies too much on the presence of a periodic motion. Another problem with this approach is that patches tend to diverge from consistency when the hole is too big. In order to solve this problem, image pyramids are used which greatly improves the inpainting results.

1.2.2 Using Optical Flow Frames

Another approach in solving the video completion problem is the use of optical flow to propagate the pixels with known colors toward the hole. The methods that falls in this category uses two steps in completing the video. The first step is to estimate the optical flow (motion inpainting, MI) inside the hole and then propagate the information (color propagtaion, CP) from known parts of the video into the hole using the optical flow values. In [20], the motion is completed by using motion patches similar to the approach used in color frames. Tang et al. [21] also inpaint the motion but use weighted priority of patches to select the best matching patch.

Video completion can benefit from frame interpolation methods that use motion inpainting [22]. The difference between the two problems is the unavailability of the 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. [23] 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.

After motion inpainting, color propagation is trivial. The success of these methods rely heavily on the accurate estimation of optical flow at the boundary of the hole. Moreover, working on the optical flow field does not ensure consistent motion between the hole and the boundary. Since most of the motion inpainting methods use only two frames, the consistency is also limited within two frames. Aside from that, when the hole becomes very large, the motion information at the boundary will have difficulty in reaching the center of the hole.

Most of the effort in video completion have been focused on solving the spatial and temporal consistency in color. However, the motion characteristics of the completed video have not been sufficiently addressed. With methods that uses only color frames, the motion consistency was not addressed because there is no explicit motion constraints applied during the inpainting process. Even with the use of 3-dimensional patches for patch matching, the color channels are still not enough to sufficiently address the motion inconsistencies.

Although methods that uses motion frames implicitly address the motion consistencies, the method still lacks in several aspects. First, spatial consistency is hard to achieve if the holes become very big because the information outside the boundary of the hole could not reach the center. The convergence of global similarity measures takes longer time to a point of nonconvergence. The temporal motion consistency on the other hand is hard to solve if the size of patches are small. However, increasing the patch size will result in including unnecessary information hence, a control becomes necessary.

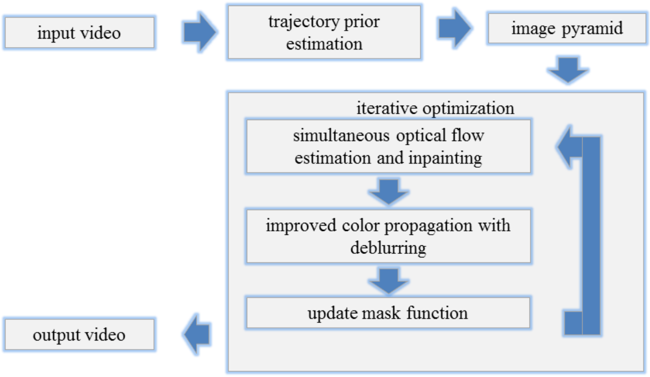
input video
trajectory prior estimation
image pyramid
iterative optimization
simultaneous optical flow estimation and inpainting
improved color propagation with deblurring
update mask function
output video

Figure 1: Overview of our proposed method.

1.3 Thesis Contributions

This thesis addresses the problem of video completion through motion inpainting and color propagation. We summarize our contributions as follows.

- Our first contribution is the simultaneous optical flow estimation and inpainting. By incorporating a spatially varying mask function in the data term of the optimization function, we were able to estimate the motion inside the hole by basing it to the motion at its boundary. We use two smoothness constraints to achieve coherent motion among pixels. One is the spatial smoothness constraint that is enforced among neighboring motions which allows us to keep the motion of the hole and the boundary close as well as the total motion inside the hole itself. The second constraint is the trajectory smoothness measure which relates the forward and backward flow computed among three frames. The trajectory measure or prior is calculated as the ratio of the forward and backward optical flow which is estimated by averaging many point values under the assumption that the motion is purely translational or the rotational motion is very low.
- Another contribution of this work is the iterative optimization framework that allows us to continuously compute the optical flow and propagate the color from known parts of the video into the hole. In order to do this, we modify the mask function used in data term of the optical flow estimation function by inferring the distance of the frames of the source pixel and the hole. This distance increases as the frame of the source pixel moves farther away from the frame of the current inpainted hole. In our method, we use a negative exponential which decreases in value as the distance increases. This means that colors that come from distant frames will have less effect on the value of the optical flow.

We show the overview of our contribution in Figure 1.

2 Simultaneous Optical Flow Estimation and Inpainting

It is necessary to jump into discussing the most critical part of our work in this section, where most of the extensions, initializations and preprocessing is derived. We will first introduce our multi-frame optical flow estimation and inpainting method. For simplicity reasons, we will generalize everything into three frames.

2.1 Optical Flow Estimation

The general two-frame optical flow estimation function (Horn-Schunck [1]), consists of a brightness constancy term and a spatial smoothness term. We define the vector $\mathbf{u} = (u, v)$ as the optical flow between frames, say, I_0 and I_1, where u and v are in the x and y direction, respectively. We represent the brightness constancy term as the data term E_D:

$$E_D(\mathbf{u}) = \lambda_D(\mathbf{x})\psi\left(I_1(\mathbf{x}+\mathbf{u}) - I_0(\mathbf{x})\right) \qquad (1)$$

where ψ can be any convex penalty function (*L1* [4], *Lorentzian* [6]). In this work, we use the differentiable L1-approximation - *Charbonnier* [7] function. We also define the spatially varying $\lambda_D(\mathbf{x})$ which we call as the *mask function.*

We will diverge from [1] and use a non-local (weighted median filter-based [2]) regularizing term. We define this as the spatial energy term, E_S:

$$E_S(\mathbf{u}) = \frac{\lambda_S}{2}\|\mathbf{u} - \hat{\mathbf{u}}\|^2 + \sum_{n}^{N} w_n \left|\hat{\mathbf{u_i}} - \hat{\mathbf{u_n}}\right|_2^2 \qquad (2)$$

Using three or more frames, it is possible to compute two optical flows, namely *forward* and *backward* flows, that are both based on a single reference frame. In prior work [3], these are estimated using only two frames. This is done by designating the forward flow as the mapping of pixels from I_0 to I_1 and the backward from I_1 to I_0. As a result, the two flows are not necessarily spatially coherent because they are based on two different reference frames.

In our work, the forward flow is defined as the mapping of pixels from frame I_j to I_{j+1} and the backward is from I_j to I_{j-1}. To simplify the succeeding definitions, we will use the subscript $\{b, 0, f\}$ instead of $\{j-1, j, j+1\}$ to represent $\{backward, reference, forward\}$ terms. We then represent the forward and backward flow as $\mathbf{u_f}$ and $\mathbf{u_b}$, respectively.

Since both $\mathbf{u_f}$ and $\mathbf{u_b}$ are based on a single reference frame, under certain assumptions, it is possible to derive a relationship between the two. For example, we can say that they have the same magnitude but opposite in direction (constant velocity), $\mathbf{u_f} + \mathbf{u_b} = 0$. Other relationship can be derived which will be left for discussion in the succeeding sections.

In the meantime, let us assume that the relationship is described by a general function $\phi(\mathbf{u_f}, \mathbf{u_b})$.We can then subject the optical flow function to the following strict constraint:

$$\phi(\mathbf{u_f}, \mathbf{u_b}) = 0 \qquad (3)$$

We will call Equation (3) as the *trajectory constraint* and assume that ϕ is convex and differentiable. We then impose this as a soft constraint to our total optimization function by defining a third energy term E_T.

Using augmented Lagrangian method, we will introduce a dual update variable $\mathbf{b^k} = (b_u^k, b_v^k)$ and define E_T as:

$$E_T(\mathbf{u_f}, \mathbf{u_b}) = \frac{\lambda_T}{2} \left|\phi(\mathbf{u_f}, \mathbf{u_b}) - \mathbf{b^k}\right|_2^2 \tag{4}$$

For the sake of clarity, we will rewrite E_D and E_S as functions of $\mathbf{u_f}$ and $\mathbf{u_b}$:

$$\begin{aligned} E_D(\mathbf{u_f}, \mathbf{u_b}) = \lambda_D(\mathbf{x})[&\psi\left(I_f(\mathbf{x} + \mathbf{u_f}) - I_0(\mathbf{x})\right) \\ &+\psi\left(I_b(\mathbf{x} + \mathbf{u_b}) - I_0(\mathbf{x})\right)] \end{aligned} \tag{5}$$

$$\begin{aligned} E_S(\mathbf{u_f}, \mathbf{u_b}) &= \frac{\lambda_s}{2}[\|\mathbf{u_f} - \hat{\mathbf{u_f}}\|^2 + \|\mathbf{u_b} - \hat{\mathbf{u_b}}\|^2] \\ + \sum_n^N w_n &\left|\hat{\mathbf{u_{fi}}} - \hat{\mathbf{u_{fn}}}\right|_2^2 + \sum_k^N w_n \left|\hat{\mathbf{u_{bi}}} - \hat{\mathbf{u_{bn}}}\right|_2^2 \end{aligned} \tag{6}$$

$$\tag{7}$$

Combining all the energy terms, our optical flow estimation function becomes:

$$\min_{\mathbf{u_f}, \mathbf{u_b}} (E_D + E_S + E_T)(\mathbf{u_f}, \mathbf{u_b}) + const. \tag{8}$$

2.2 Joint Estimation and Inpainting

The inpainting process is embedded in the minimization of the function in (8). By assigning a spatially varying value for λ_D, we can control the effect of the brightness constancy on the total solution. Inside the hole, we initially remove the effect of the data term by assigning a *binary mask* to λ_D and rely solely on the spatial and trajectory smoothness terms. This means that the optical flow inside the hole is dependent only on the motion values along the boundary. In the succeeding sections, we will modify λ_D to allow for intermediately inpainted color to affect our solved optical flow.

We use variational approach in minimizing the function in (8). In order to do so, we linearize the data energy term using the first order Taylor approximation. This yields:

$$\begin{aligned} E_D(u_f, u_b) = \lambda_D[&\psi\left(u_f I_{f_x} + v_f I_{f_y} + I_{f_t}\right) \\ &\psi\left(u_b I_{b_x} + v_b I_{b_y} + I_{b_t}\right)] \end{aligned} \tag{9}$$

where the I_{fx} and I_{fy} terms are the partial derivatives of I_f which is approximated by convolving the image with a kernel filter. On the other hand, $I_{ft} = I_f(\mathbf{x}) - I_0(\mathbf{x})$.

In practice, the linearization will not be satisfied because the image gradients will have large variations between the two frames due to large motions. This problem is addressed using continuous refinement [11]. The goal is to first find an initial guess for the optical flow, $\mathbf{u_{f0}}$ and then continuously minimize the function along the differential $\Delta\mathbf{u_f} = \mathbf{u_f} - \mathbf{u_{f0}}$. Famous methods to find a good initial guess are image pyramids [8], patch matching [9], or point correspondences [10].

Assuming that $\mathbf{u_{f0}}$ is already a close approximation of the desired value, we warp I_f towards I_0 using bicubic interpolation and then solve the differential $\Delta\mathbf{u_f} = (\delta u_f, \delta v_f)$. The new warped image is given by $\bar{I_f}(\Delta\mathbf{u_f}) = I_f(\mathbf{x} + \mathbf{u_{f0}} + \Delta\mathbf{u_f})$. We then rewrite the data term again using the warped image for the forward flow as:

$$E_D(\Delta\mathbf{u_f}) = \lambda_d\left[\psi\left(\delta u_f \bar{I_{f_x}} + \delta v_f \bar{I_{f_y}} + \bar{I_{ft}}\right)\right] \tag{10}$$

We do the same step with I_b and then minimize (8) in terms of $\Delta\mathbf{u_f}$ and $\Delta\mathbf{u_b}$.

To minimize (8), we perform the following double alternating direction method (ADMM). First we hold $\mathbf{u_b}$ and $\mathbf{b^k}$ constant and the resulting function will be dependent only on $\mathbf{u_f}$ and $\hat{\mathbf{u_f}}$.

We perform another ADMM and hold $\hat{\mathbf{u_f}}$ constant to find $\mathbf{u_f}$. The resulting function can be minimized by solving the Euler-Lagrange equations and performing a simple point-wise algebraic manipulation. After finding an initial value for $\mathbf{u_f}$, we then solve for $\hat{\mathbf{u_f}}$ as proposed in [2].

The same step is done for the backward direction and then $\mathbf{b^k}$ is updated as $\mathbf{b^{k+1}} = \phi(\mathbf{u_f}, \mathbf{u_b}) - \mathbf{b^k}$. We call this step as the inner iteration and summarize it in Algorithm 1.

Algorithm 1 Inner iteration for simultaneous optical flow estimation and inpainting.

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

3 Trajectory Prior Estimation

First, we will justify why a constant velocity assumption in the trajectory constraint of the optical flow functional could be problematic by giving an example. The trajectory constraint we use describes the relationship between the forward and the backward flow. We could say that by assuming a constant velocity motion $\mathbf{u_f} + \mathbf{u_b} = \mathbf{0}$ and assigning a small value to λ_T, we can accommodate small changes in the camera motion. However, there are two cases where this assumption pose a problem.

The general case is when the velocity change abruptly that the difference between two consecutive optical flow frames is large. The other case has to do with changing perspectives due to camera motion. Say for example we are inpainting a wall. On one side of the hole, the wall might be seen as planar and runs at a diagonal with the camera plane. When the wall reappears on the other side of the hole in the sequence, it might suddenly appear as a straight line. If the camera velocity is constant, it is possible to trace the motion of all the points in the wall and they will converge as a line on the other side. The motion can then be estimated when the wall is still visible and simply follow that motion to the other side of the hole.

The problem occurs when the camera's velocity change abruptly while the wall is still hidden. Take for example the case in Figure 2. The wall that is being inpainted was visible on both sides of the hole but

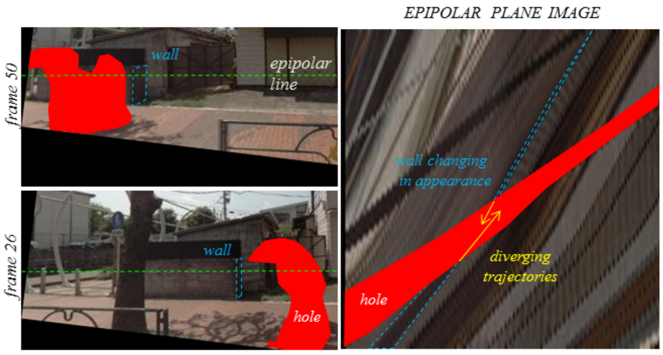

Figure 2: Changing camera velocity poses difficulties in video completion especially when the object being inpainted changes its appearance. Here we show how a wall maps to an almost straight line when passing through a hole.

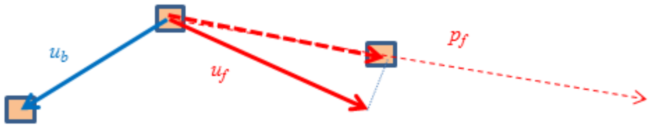

Figure 3: Trajectory prior

with different perspective (one side is more perpendicular to the camera plane). During the time the wall is still behind the hole, the camera changes speed. If we follow the same idea as in the first case, we will run into a problem where tracing the motion of the points in the wall will result in a different reconstructed perspective of the wall.

Another way to view this problem is when we try to interpolate the motion from both sides of the hole. Take for example the right-side image in Figure 2. A constant velocity corresponds to a straight line in the non-strict epipolar plane. If we trace the movement of two corresponding points in the wall from both ends of the hole, we will find that they will not converge at the center, unless of course we deliberately track them.

To address the discussed problem, we propose to estimate the relative motion of all the points in the sequence by reasoning on the motion of the known ones. We designate this relative motion as the trajectory prior and can be described in the trajectory energy of the optical flow estimation as:

$$\phi(\mathbf{u_f}, \mathbf{u_b}) = \mathbf{u_f} + \rho\mathbf{u_b} = \mathbf{u_f} + \mathbf{p_f} \qquad (11)$$

where the p_f term is the trajectory prior. The expression can be illustrated as in Figure 3.

We allow the forward flow u_f to be around the same value as the trajectory prior by minimizing their difference. If in case u_b is known (such as the case along the boundary of the hole), it is possible to solve for the trajectory of u_f by using this technique. In most cases however, the two optical flows are both unknown, therefore the difference serves as a weak constraint on their estimated values.

Assuming that the camera motion is dominantly translational, with the parameters $T = (T_x, T_y, T_z)$, the trajectory prior can be defined as:

$$p_{fx} = \frac{-T_{xf} + xT_{zf}}{-T_{xb} + xT_{zb}} u_b = \rho u_b \qquad (12)$$

We call ρ as the transition ratio which defines the transition of the trajectory from the backward to the forward direction.

Now, we need to solve for ρ from the camera motion parameters. We will show two methods to solve this. The first is a direct solution through structure-from-motion technique. Then we will propose a faster and simpler method through only point correspondences between three frames.

3.1 Method using Structure from Motion (SFM)

Assuming a predominantly translational egomotion, the transformation matrix of a projective camera system can be simplified as:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{T} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \qquad (13)$$

We first find the matching points among all the necessary frames using scale invariant feature transform (SIFT). Then using the matched features, we solve for the camera translation. A standard process is well-described in [28]. Using the translation parameters, we then derive the transition ratio from Equation 12.

3.2 Method using Points Correspondences (PC)

We propose a faster method in solving the trajectory prior. Instead of directly estimating camera parameters, we make use of the definition of transition ratio. Assuming that the camera does not change its depth much compared to its motion along the x or y-axis ($T_z << T_x$), we can ignore the T_z parameter in the equation and redefine the approximate transition ratio as:

$$\rho \approx \frac{T_{xf}}{T_{xb}} \qquad (14)$$

We can find ρ instead by taking the average ratio of the known optical flows u_{fi} and u_{bi} of all points i outside the hole. To find these optical flows, we solve again the SIFT features, this time only between three frames. The flows are defined as $u_{fi} = x_i - x_j$, where x is the position of point i in the reference frame and point j in the forward frame (same with backward frame). The transition ratio is then given by:

$$\rho = \frac{1}{N} \sum_{i}^{N} \frac{u_{fi}}{u_{bi}} \qquad (15)$$

3.3 Comparison

We compare the two methods on a synthetic video with sudden change in camera velocity about the 11th frame. The SFM method achieves a 0.031 RMSE while the PC method is at 0.064.

Although the error is slightly larger, there are several benefits to using the PC method. First, the SFM requires all the frames to estimate the camera parameters while the PC only requires three. Second, even our naive implementation of the SIFT detector (0.06s) proves folds faster than the SFM (baseline VisualSFM [26] at 4.3s). Finally, in cases where it is impossible

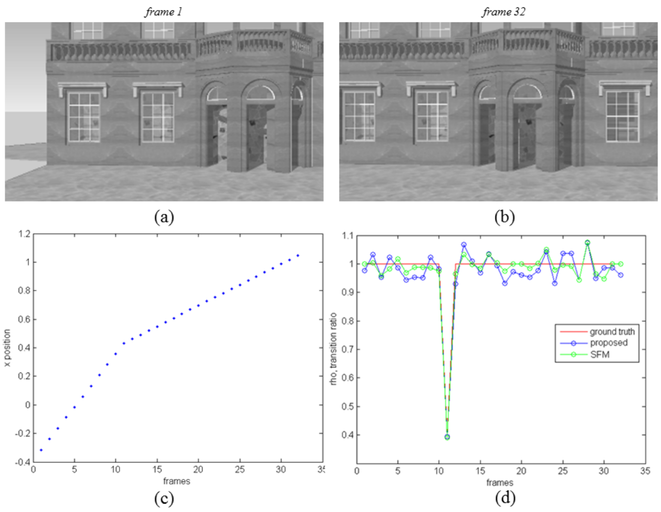

Figure 4: (a,b) Representative frames; (c) Camera position along the x-axis. Note the change in velocity about the 11th frame; (d) Comparison on the solved ρ between the SFM and the proposed method.

to estimate the camera motion due to lack of features (i.e. planar scenes, insufficient depth variance), the PC method will still work.

4 Iterative Optimization

Generally speaking, the inpainted motion can be intuitively used to propagate the color from known regions of the video towards the hole. However, we argue that it is possible to use the newly inpainted color of the hole to the brightness constraint to further improve the motion estimation. This time, instead of using a binary label for the mask, we use a probabilistic function that is dependent on the distance between the inpainted frame and the source color frame.

Using this assumption, we combine the motion estimation and inpainting and the color propagation into an iterative optimization framework.

We first initialize all the optical flows and dual update variable to 0 and then solve for the initial $\mathbf{u_f}$ and $\mathbf{u_b}$ for all the frames. Using these initial values, we then perform our color propagation method.

4.1 Color Propagation

We first present a simple color propagation technique based on linear warping. The method starts with an inpainted motion inside the hole. The values of the optical flow in each frames forms a graph that maps the pixels between neighboring frames (see Figure 5). By following the map, the known color is then propagated to the hole.

Ideally, we want the optical flow to point to an exact location in another frame. However, this is seldom the case. To solve this problem, we *warp* the known pixels to the hole via bicubic interpolation (see Figure 6). The four pixels in the vicinity of the hole and their neighboring pixels are used as initial values of the interpolation method. The image derivatives are solved about these four points and the value of $(\bar{x}, \bar{y})$ is determined using the bicubic polynomial [12].

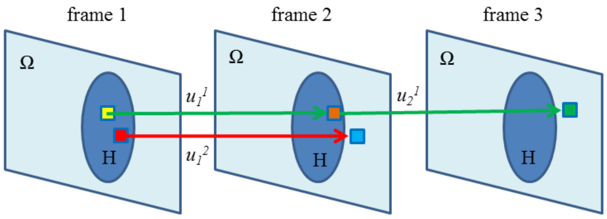

Figure 5: Color propagation as graph. The optical flow is treated as a graph that maps the correspondence of pixels among the frames.

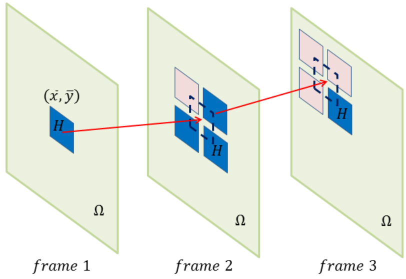

Figure 6: Inexaxt optical flow points to a vicinity of four neighboring pixels.

We perform the propagation for each of the forward and backward flow directions. As an additional step, we record the distance between the current frame and the source frame. Say we are completing the n^{th} frame I_n of the image sequence. For each pixel x_i, we are propagating the color $\bar{x}_i$ of the known region in $(n + s_i)^{th}$ frame I_{n+s_i}. We define a distance $\mu(x)$ as:

$$\mu(x) = (n + s_i) - n = s_i \tag{16}$$

After color propagation, we will have two differently inpainted frames I_{Hf} and I_{Hb}. We combine the two images by first, in the regions close to the hole boundary, choosing the direction which has the lower μ value. As we move deeper in the hole, we blend the color from both direction using:

$$I_H = \frac{\mu_b^2}{\mu_b^2 + \mu_f^2} I_{H_f} + \frac{\mu_f^2}{\mu_b^2 + \mu_f^2} I_{H_b} \tag{17}$$

4.1.1 Effect of Consecutive Warping

Since the color of the pixel in the hole is only an interpolated value of the surrounding pixels, subsequent interpolation will result in an averaging effect. This effect is more apparent with large holes which inner parts can only be inpainted using sources from very distant frames.

To reduce the blurring effect, we use the following technique as illustrated in Figure 7. As of now, we are inpainting by interpolating the values in the source on a frame-by-frame basis. Say for example, we are completing the hole in frame 1. We do this by propagating the color following $u_{(1,2)}$ to frame 2 and if this points to the hole, we follow to $u_{(2,3)}$ and so on. Then, we warp the colors from frame 3 to frame 2 using $u_{(2,3)}$ and then the colors from frame 2 to frame 1 using $u_{(1,2)}$.

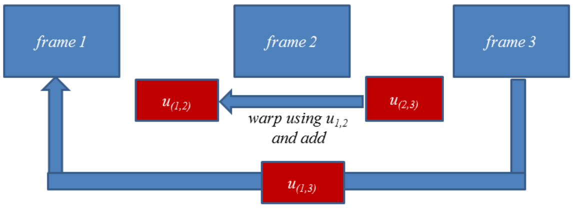

Figure 7: Warping of optical flow.

We propose to directly interpolate the values in frame 3 to frame 1 by solving for the optical flow $u_{(1,3)}$. Fortunately, we do not need to solve for this separately. In our method, we first solve for $\mu(x)$ of all the frames using the method in the previous seciton. The $\mu(x)$ contains the frame location of the source pixels in all x in the hole. To illustrate, say, we are completing the n^{th} frame I_n. For a pixel x in the hole in I_n, we have the location of the source frame $n + \mu(x)$. We then linearly warp the optical flow of that frame $u_{n+\mu(x)}$ using the flow of the preceding frame $u_{n+\mu(x)-1}$. We iterate this linear warping k times until we get to the current frame $n + \mu(x) - k = n$. As a result, we will get the optical flow $u_{n+\mu(x)}$ of point x that maps it to frame $n + \mu(x)$. We repeat this process for all pixels in parallel.

With the updated motion, we also update I_{Hb} and I_{Hf} and perform the weighted combination to get I_H as in Equation (17).

4.2 Modifying the Mask Function

After one color propagation step, we will get an initially inpainted image sequence. We now use this initially completed hole to further improve the result of the motion inpainting.

To do this, we modify the binary label mask function to have a spatially varying value based on the reliability of the inpainted pixel. We define the reliability of the pixel as:

$$m(x) = \gamma^{-\mu(x)} \qquad (18)$$

where γ is a positive real number. The value of gamma controls how much the inpainted pixel affects the overall error.

Choosing an arbitrary γ value will result in unstable global minimization. In theory, we want the total error inside the hole to be less than that of its boundary [13] [14]. This will help in the convergence and allows the information to gradually propagate towards the hole. Choosing a small value for γ, however will let the newly inpainted color at the center of the hole more effect on the minimization rather than the spatial and trajectory smoothness which may prematurely converge to a wrong local minimum. On the other hand, a very large value will result in the information not reaching the center of the hole, especially if it is too big. In our experiments, we choose $\gamma = 1.3$ and find this value suitable in the videos that we used.

4.3 Note on Convergence

Before performing the iterative motion inpainting and color propagation, we first solve for the image pyramids through a coarse-to-fine strategy. We do this by repeatedly down-sampling the image by a factor α. The higher level $l+1$ image (or the coarser scale) given the current level l image G_l is solved as:

$$G_{l+1}(x, y) = \sum_{m=-2}^{2} \sum_{n=-2}^{2} 0.25 G_l(2x + m, 2y + n) \qquad (19)$$

Using this approach, we compensate for large pixel motions that is usually present in our videos. We use $\alpha > 0.5$ so that each of the succeeding level is a blurred version of the lower level.

We start the iteration from the coarsest level and use an initial value of the mask to be zero inside the hole. We then orderly choose a frame, I_0 and its two neighboring frames I_f and I_b. The unknowns $(\mathbf{u_f}, \mathbf{u_b}, \mathbf{b^k})$ are initialized to zero. We then iterate the joint motion estimation and inpainting, color propagation and the mask update.

In each of the iteration we are presented with an entirely different optimization function. It is possible that the final output is not a desirable result, which means that the hole could contain any value that a human viewer will see as visually unpleasant. However, it can be proven that at each of the iteration, since the mask function is held at a constant value, we are optimizing a function that is convex and therefore will converge at a point. The point definitely optimizes the function, but does not mean that it is an optimum value.

We summarize the steps of this method in Algorithm 2.

Algorithm 2 Iterative motion inpainting and color propagation.

```
Require: color of $H$
  solve $trajectory_prior$
  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
```

5 Results and Discussion

Optical Flow Estimation

We test our proposed method on synthetic and real videos. We first compare our optical flow estimation method with [24]. We used the database in [25]. To constraint the comparison on the effectiveness of the regularizers, we limit the non-local implementation of [24] to the weghted median filtering similar to our approach. In this case, the only difference between their method and ours is the TV smoothness constraint.

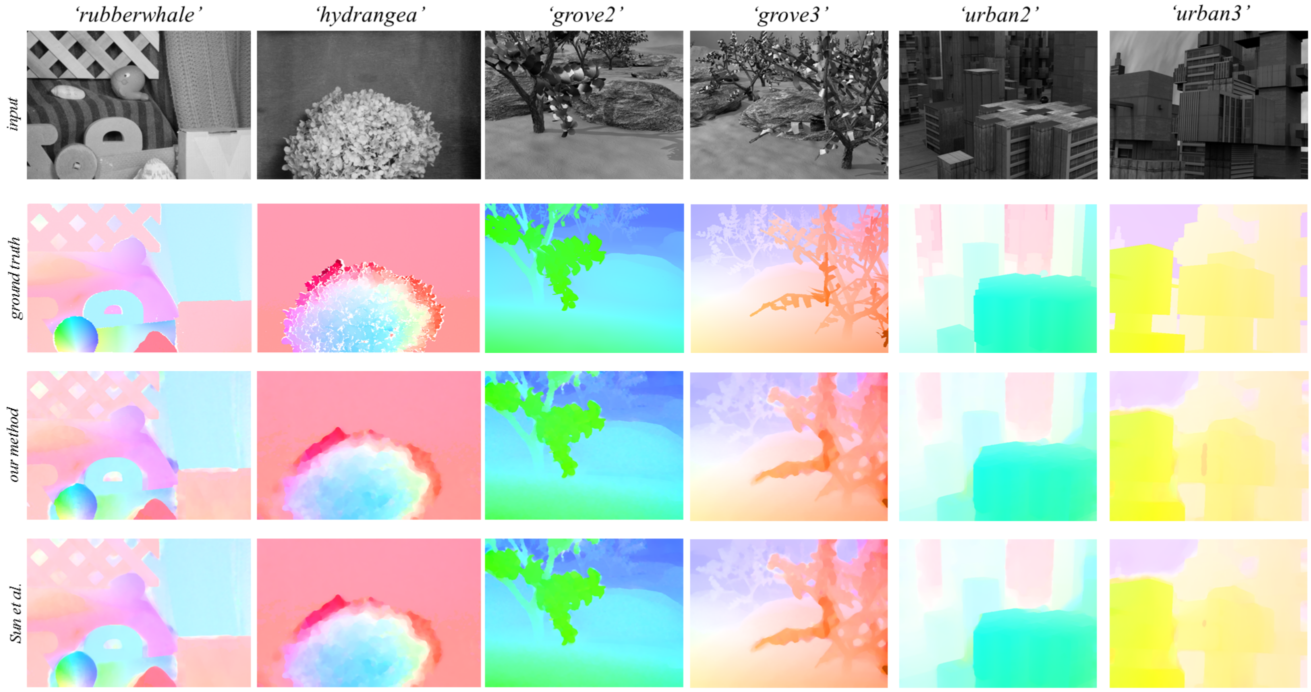

Figure 8: TV+median [24](Sun et.al) vs. median only regularizer (our method).

Sequence	End-Point Error		Runtime (s)	
	Sun et al.	Our method	Sun et al.	Our method
rubberwhale	0.180	0.187	135.9	120.1
hydrangea	0.339	0.350	138.4	123.2
grove2	0.175	0.179	186.5	180.5
grove3	0.747	0.744	186.7	126.3
urban2	0.854	0.909	183.3	177.4
urban3	0.874	0.831	209.2	123.6

Table 1: TV+median [24] vs. median only regularizer. The small increase in the end-point error, which is almost negligible, is a better trade-off for a faster and more efficient solution of the optical flow estimation.

The decrease in runtime is due to the removal of the TV smoothness constraint in the solution. Without it, we were able to remove one iteration required to solve a Gauss-Seidel step and the solution for $\mathbf{u_f}$ and $\mathbf{u_b}$ becomes a point-wise algebraic manipulation. The improvement in runtime is very important because we are solving the optical flow of many frames and the improvement in time accumulates as the number of frames increases. The representative frames are shown in Figure 8 and the quantitative comparison is shown in Table 1.

Trajectory Prior

We test the effectiveness of the trajectory prior estimation method with different videos. We first used a video where the camera suddenly change its velocity. We compare the results between the trajectory prior estimation using SFM and point correspondence. We also compare them from the result of a constant velocity assumption. We show the representative frames in Figure 9 and plot the error (difference between the ground truth video and the inpainted video) in Figure 10.

We also use a shaking video (in x-axis only) to demonstrate the effectiveness of point correspondence method in solving the trajectory prior. We show an improvement in the inpainting result and show them in Figures 9 and 10.

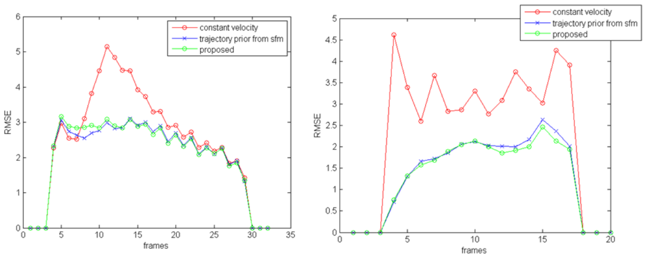

Figure 10: Plot of the error in completion of video with (left) changing velocity and (right) shaking. In both cases, the trajectory prior solution shows a significant reduction in error.

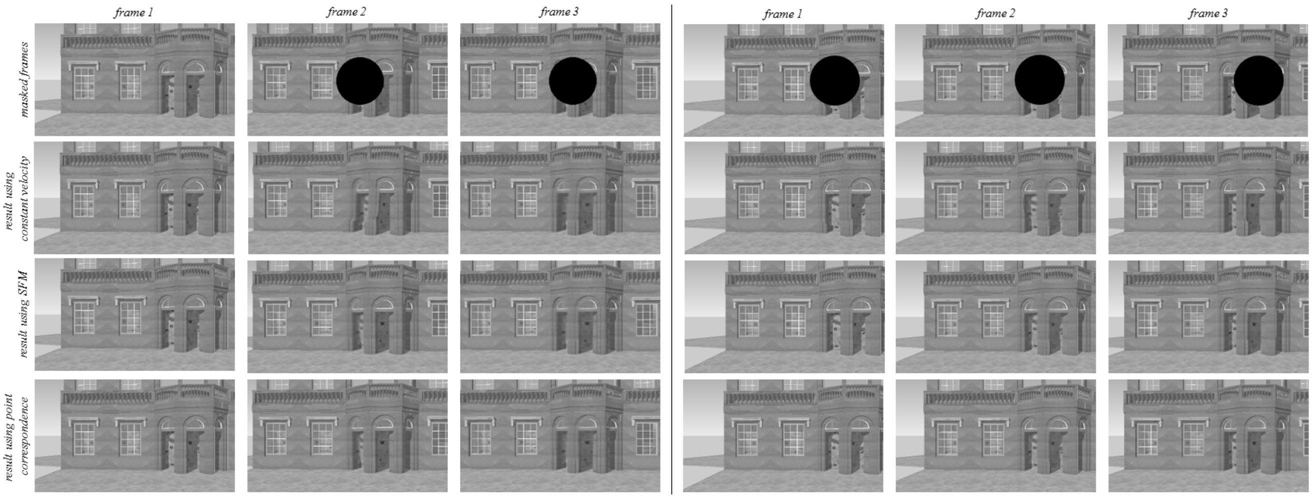

Figure 9: Representative frames of the result of the video completion method on a synthetic video with (left sequence) changing velocity and (right sequence) shaking. We compare the results using a constant velocity trajectory assumption and with the trajectory prior solutions using SFM and point correspondences.

Blur Reduction Tests

We quantify the improvement in the results by measuring the gradient-based sharpness measure (Tenengrad function [27]:

$$f_{sharp}(I) = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\left(\frac{dI_i}{dx}\right)^2 + \left(\frac{dI_i}{dy}\right)^2} \quad (20)$$

A higher value in the measure suggests a less blurred image. We confine the computation inside the hole and normalize the output. We tested the technique using a real street video sequence with removed pedestrian ('human' sequence) and show the improvement in Figure 11. The shaprness range from 5.44 to 7.20 without the deblurring technique. By applying our proposed optical flow warping method, we were able to improve the range from 7.34 to 9.14. To further test the effectiveness, we also show the comparison between the sharpness measure of the ground truth sequence and the inpainted ones in Figure 12 using the Middlebury database. We introduced a hole as shown in the mask row. We were able to improve the sharpness from 6.62-7.69 to 7.79-9.26.

Test on Street Videos

We tested the whole video completion process on real street videos where we remove the walking pedestrians. We show the representative frames of two image sequences 'human' and 'person' in Figure 13.

6 Conclusion and Future Work

In this thesis we proposed to solve the video completion problem by using a spatio-temporally consistent motion inpainting. First, we proposed a framework in inpainting motion using multiple frames by imposing a smooth spatial and trajectory constraint on the motion among the frames. We did this using a joint motion estimation and inpainting algorithm that utilizes a binary label mask to eliminate the effect of the color information inside the hole. The smoothness constraints proved to be effective in propagating the known motion from the boundaries towards the hole.

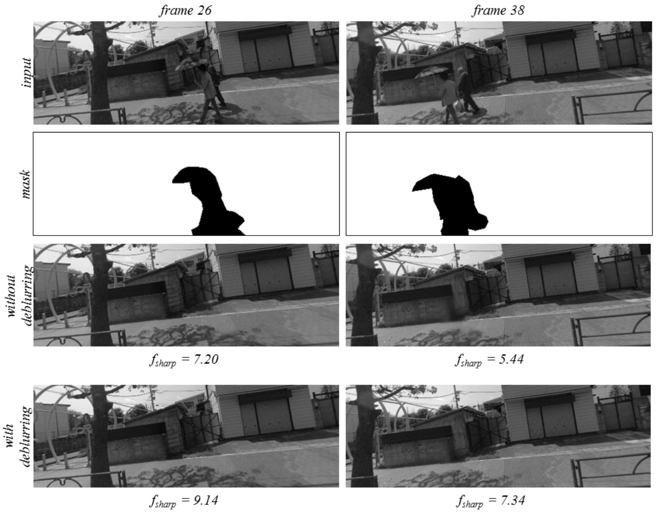

Figure 11: Result of reducing the blurring effect on the 'humanc' sequence for representative frames 26 and 38.

Secondly, we proposed a simultaneous motion inpainting and color propagation method by using an iterative optimization method. We obtained better results when we used the newly inpainted pixels inside the hole to refine the optical flow estimation inside it. We control the effect of the newly inpainted pixels using our proposed mask function that relates the frame distance of the source pixel to the reference pixel in the hole. We also introduced a trajectory prior estimation method to handle the trajectory constraint during non-smooth motion. Our method comprised of only three frames and therefore was implemented really fast. We also improved the standard color propagation method to include a technique in combining the result of two directions, namely the forward and the backwards. We combined the propagated color from both directions using our proposed blending technique. We then showed in our result that this method can accom-

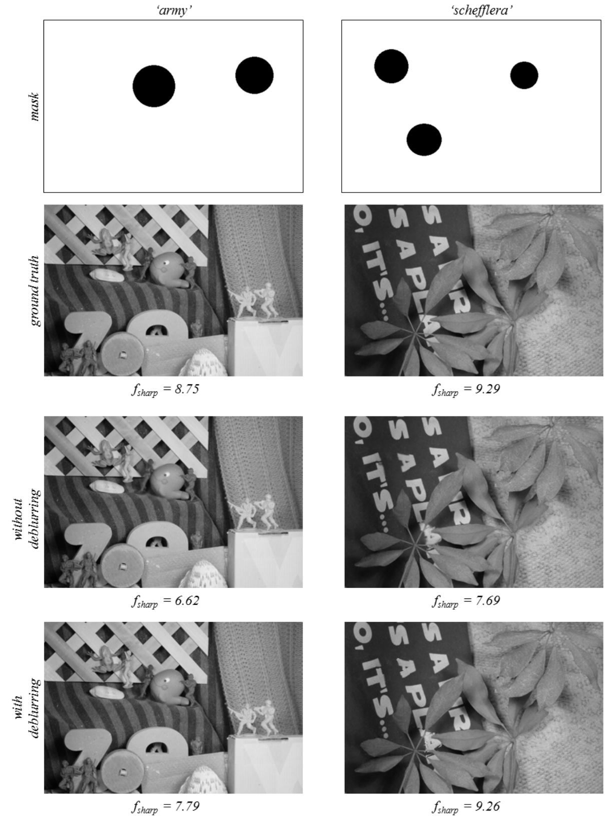

Figure 12: Result of reducing the blurring effect on the ’army’ and ’schefflera’ sequence.

plish video completion results accurately.

Our optimization framework is designed to be extended to virtually any optical flow estimation method. A choice of a good functional will result in faster approximation of the motion inside the hole and eventually faster video completion results. A real-time implementation is also desired since a lot of application, such as mixed reality, require that an object is removed and inpainted in real-time. Since the motion can be estimated by using only three frames, a real-time implementation is very possible. The only limitation is when the hole extends several frames that the first available source pixel is very far from the current frame.

A desired extension is to combine the masking of the hole and the inpainting method proposed in this thesis into one automatic framework. The burden is put on the detection and tracking of the unwanted object on all the frames in real-time.

Another possible improvement is to modify the constraints to handle more dynamic motion such as those of non-rigid objects and to consider more complex scenes such as places with clutters.

We conclude this thesis by saying that video completion is a very hard task and requires a lot of engineering in order to be useful in most applications. However, with the emergence of fast computers and algorithms, this problem is not really far from being perfectly solved.

References

[1] B.K.P. Horn, B.G. Schunck: ”Determining Optical Flow,” *Artificial Intelligence*, vol. 17, pp. 185-203, 1981.

[2] Y.Li, S.Osher: ”A new median formula with applications to PDE based denoising,” *Communication Mathematics Science*, vol. 7, no. 3, pp. 741-753, 2007.

[3] A. Randriantsoa, Y. Berthoumieu: ”Optical flow estimation using forward-backward constraint equation,” *In Proc. ICIP*, 2000.

[4] N. Papenberg, A. Bruhn, T. Brox, S. Didas, J. Weickert: ”Highly Accurate Optic Flow Computation with Theoretically Justified Warping,” *IJCV*, vol. 67, pp. 141-158, 2006.

[5] A. Wedel, T. Pock: ”An improved algorithm for TV-L1 optical flow,” *In Proc. Dagstuhl Motion Workshop*, 2008.

[6] M. Black, P. Anandan: ”Robust dynamic motion estimation over time,” *In Proc. CVPR*, 1991.

[7] A. Bruhn, J. Weickert, C. Schnorr: ”Lucas/Kanade meets Horn/Schunck - combining local and global optic flow methods,” *IJCV*, vol. 61, no. 3, pp 211-231, 2005.

[8] P. Burt: ”Fast filter transforms for image processing,” *Computer Graphics and Image Processing*, 1981.

[9] L. Bao, Q. Yang, H. Jin: ”Fast edge-preserving patch match for large displacement optical flow,” *In Proc. CVPR*, 2014

[10] X. Li, J. Jia, Y. Matsushita: ”Motion detail preserving optical flow estimation,” *TPAMI*, vol. 34, no. 9, pp. 1744-1757, 2012.

[11] J. Weickert, A. Bruhn, N. Papenberg, T. Brox: ”Variational optic flow computation: from continuous models to algorithms,” *IWCVIA*, 2003.

[12] R. Keys: ”Cubic interpolation for digital image processing,” *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 29, no. 6, pp. 1153-1160, 1981.

[13] Y. Wexler, M. Irani: ”Space-time Completion of Video,” *IEEE TPAMI*, vol. 29, no. 3, pp 463-476, 2007.

[14] A. Criminisi, P. Perez, K. Toyama: ”Object removal by exemplar-based inpainting,” *In Proc. CVPR*, 2003.

[15] Y. Jia, et al.: ”Video Completion Using Tracking and Fragment Merging,” *The Visual Computer*, pp. 601-610, 2005.

[16] M. Granados, J. Tompkin, K. Kim, O. Grau, J. Kautz, C. Theobalt: ”How not to be seen - object removal from videos of crowded scenes,” *Computer Graphics Forum*, vol. 31, no. 2.1, pp. 219-228, 2012.

[17] M. Granados, et al.: ”Background inpainting for videos with dynamic objects and a free moving camera,” *In Proc. ECCV*, 2012.

[18] Z. Zhang, A. Ganesh, X. Liang, Y. Ma: ”TILT: Transform invariant low-rank textures,” *In Proc. ACCV*, 2010.

[19] J. Jia, et al.: ”Video Repairing: inference of foreground and background under severe occlusion,” *In Proc. CVPR*, 2004.

[20] T. Shiratori, et al: ”Video Completion by Motion Field Transfer,” *In Proc. CVPR*, 2006.

[21] N. Tang, C. Hsu, C. Su, T. Shih: ”Video inpainting on digitized vintage films via maintaining spatio-temporal continuity,” *IEEE Trans. on MM*, vol. 13, no. 4, pp. 602-614, 2011.

[22] K. Chen, D. Lorenx: ”Image sequence interpolation using optimal control,” *Journal of Math. Imaging and Vision*, vol. 41, pp. 222-238, 2011.

[23] M. Werlberger, T. Pock, M. Unger, H. Bischof, ”Optical flow guided TV-L1 video interpolation and restoration,” *In Proc. EMMCVPR*, 2011.

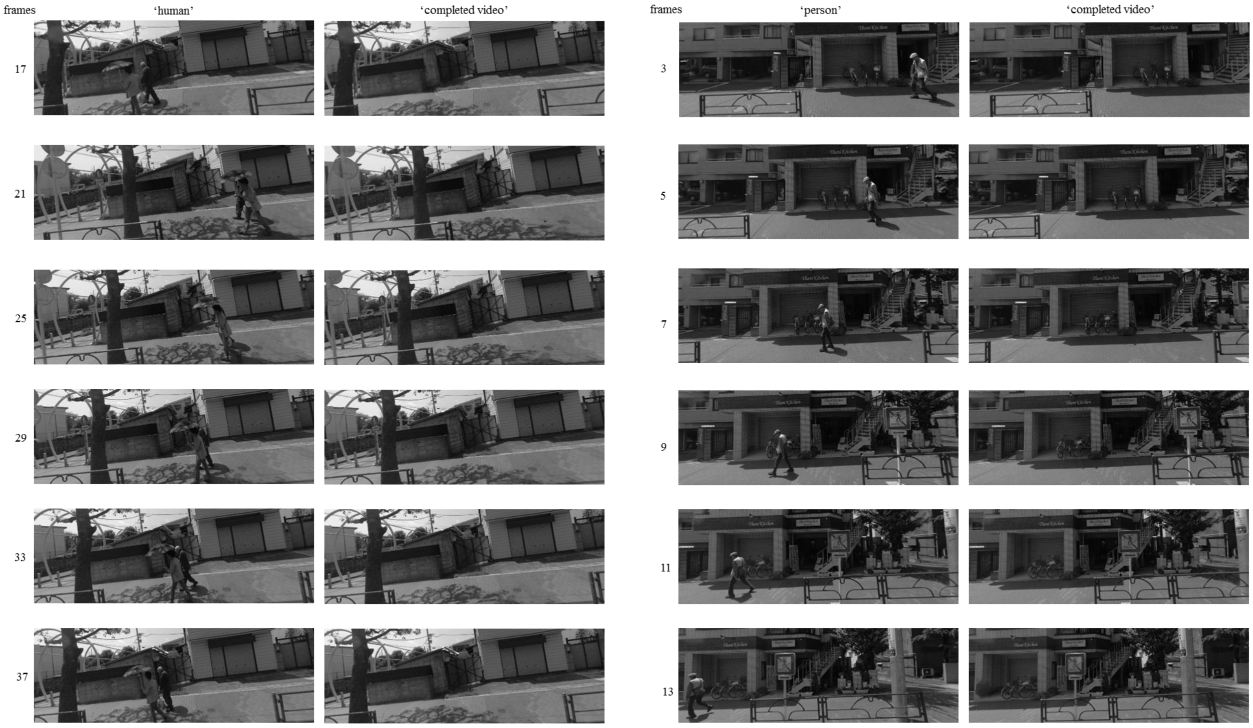

Figure 13: Representative frames of the result of completion on the "human" and "person" sequences where the pedestrians are removed.

[24] D. Sun, S. Roth, M. Black, "Secrets of optical flow estimation and their principles," *In Proc. CVPR*, 2010.

[25] Optical Flow - The Middlebury Computer Vision Pages, http://vision.middlebury.edu/flow/data/

[26] Visual SFM, http://ccwu.me/vsfm/

[27] E. Krotkov, "Active computer vision by cooperative focus and stereo," *Springer-Verlag*, 1989.

[28] R. Hartley, A. Zisserman, Multiple view Geometry in Computer Vision, Cambridge University Press, 2nd Edition, 2004.