VIDEO INTERPOLATION THROUGH GREEN'S FUNCTIONS OF MATCHING EQUATION

Perfilino Eugênio Ferreira Jr[1], *José R. A. Torreão*[2], *Paulo Cezar P. Carvalho*[1] *and Luiz Velho*[1]

[1]IMPA–Instituto Nacional de Matemática Pura e Aplicada - Estrada Dona Castorina, 110
22460-320 Rio de Janeiro, RJ, Brasil
{perfeuge,pcezar,lvelho}@visgraf.impa.br
[2]Instituto de Computação - Universidade Federal Fluminense - Rua Passo da Pátria, 156
24210-240 Niterói, RJ, Brasil
jrat@ic.uff.br

ABSTRACT

We propose a novel approach for generating video interpolating frames from optical flow information. Our approach is based on solving an affine image-matching equation via the Green's function method. This method provides a pair of filters that, when applied on each frame of an input video signal, allows the generation of a virtual sequence that creates extra frames. Our technique provides smooth reconstruction of video, especifically of sequences created through camera translation or translation of objects in a otherwise static scene.

1. INTRODUCTION

Video interpolation is an important task for smooth reconstruction of 3D spatiotemporal signals. Some approachs to interpolation between two or more images, involving optical flow, are in [1]-[2]. Here the strategy employed was to consider solutions of matching (irradiance-conservation) equations of the form

$$I_2(x+u, y+v) = I_1(x, y), \tag{1}$$

where I_1 is the input image, (u, v) is the optical flow field, and I_2 is a matching image, to be found. Together with I_1, I_2 conveys the motion information essential to video interpolation.

Expanding the left-hand side in a Taylor series up to second order in u and v, and performing a suitable change of variables, it is easy to see that equation (1), for matching along a general direction θ, reduces to the one-dimensional form [3]

$$\frac{u^2}{2}{I_2}'' + u{I_2}' + I_2 = I_1, \tag{2}$$

Thanks to CAPES agency for funding.

where $I_i = I_i(x, y + \gamma x)$, for $\gamma = \tan\theta$, and where the primes denote differentiation with respect to x.

Solutions to equation (2), for uniform u, have already been considered in [3], in the context of 3D shape estimation. These can be obtained as the convolution of I_1 with a linear, shift-invariant filter which is the Green's function for equation (2). An extension to this approach, in the context of motion simulation and modeling u as an affine function $U = u + u'x$, where both u and u' are constant, has been shown to yield equation (2) [4]. Here we show an application to this last technique. Considering the solution to the matching equation under such model, we obtain a Green's function filter which, when applied to a frame of a given input video, is able to generate extra frames, for video sequences obtained through camera translation or translation of objects in a otherwise static scene.

The organization of this paper is as follows: in Section 2, we describe the affine motion model; in Section 3 we derive the Green's function filter, and consider its practical implementation; in Section 4, our method for video interpolation is described; in Section 5, we present and discuss some experimental results; finally, in Section 6, we make our concluding remarks and propose directions for future work.

2. AFFINE OPTICAL FLOW MODEL

The general affine matching model allows us to represent much more varied and complex motions in comparison with the constant-flow model. The affine motion model can be expressed by a first order approximation of the optical flow $(U, V)^T$ as

$$\begin{bmatrix} U \\ V \end{bmatrix} = \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} + \begin{bmatrix} D + S_1 & S_2 - R \\ R + S_2 & D - S_1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}, \tag{3}$$

where $(x, y)^T$ are the pixel coordinates in the image, $(u_0, v_0)^T$ are the uniform, zero-order components of the flow, D is a dilation(contraction or expansion) component, R is the rotation component, and S_1 and S_2 are shear components [4, 5, 6, 7]. We consider the special case in which u can be written as $U = u + u'x$ in Equation (2). This is equivalent to assuming that $S_2 - R = 0$, $u = u_0$, $u' = D + S_1$, and that the flow is constant along the V direction. Manipulation of parameters u and u', together with the choice of the direction u, allows simulating a broad class of movements in images [4].
In this work, we obtain u and u' from the affine optical flow computed between two frames of the video sequence to be interpolated. We use the implementation of an algorithm to compute affine optical flow given in [8].

3. AFFINE MATCHING THROUGH GREEN'S FUNCTION

Let us go back to the matching equation (2), now considered under an affine optical flow model, *viz.*,

$$\frac{U^2}{2} {I_2}'' + U{I_2}' + I_2 = I_1, \tag{4}$$

where $U = u + u'x$ is in the above form, and both u and u' are assumed to be constant and represent, respectively, the optical flow and its spatial rate of change. Similarly as in (2), I_1 and I_R are taken as functions of $(x, y + \gamma x)$, in order to accomodate matching along a general direction $\theta = \tan^{-1} \gamma$.

Equation (4) has the Cauchy-Euler form [9], and its solution, over a domain D, can be expressed as

$$I_2(x, y + \gamma x) = \int_D G(x, x_0) I_1(x_0, y + \gamma x_0) dx_0, k \tag{5}$$

where $G(x, x_0)$ is the Green's function, which solves

$$\frac{U^2}{2} G'' + UG' + G = \delta(x - x_0), \tag{6}$$

given the same boundary conditions as assumed for I_R.

Under the sole condition that $G(x, x_0)$ remains finite as x goes to infinity, a solution to (6) can be found as

$$G(x, x_0) = \frac{2}{u'^2 \beta (x_0 + a)} \left[\frac{x + a}{x_0 + a}\right]^{\alpha} \sin\left\{\beta \log\left[\frac{x + a}{x_0 + a}\right]\right\}, \tag{7}$$

for $x > x_0$, with $G(x, x_0) = 0$, otherwise. The parameters a, α and β are given as

$$\begin{cases} a = \frac{u}{u'} \\ \alpha = -\frac{1}{u'} + \frac{1}{2} \\ \beta = \frac{1}{u'}\sqrt{1 + u' - \frac{u'^2}{4}}, \end{cases} \tag{8}$$

leading to a bounded $G(x, x_0)$ for $D \subset (-a, \infty)$, as long as we take $0 < u' < 2$. In finite domains, this solution is valid for $2 - 2\sqrt{2} < u' < 2 + 2\sqrt{2}$.

It should be mentioned that, taking a similar form to (7), but with a cosine substituted for the sine term there, we would obtain a second filter - let us call it $H(x, x_0)$ - which, when applied over I_1, leads to a solution of the homogeneous equation associated with (4). Therefore, a general matching pair can be obtained by filtering the input image through a linear combination of G and H, which are the imaginary and real parts, respectively, of the complex function

$$\frac{2}{u'^2 \beta (x_0 + a)} \left[\frac{x + a}{x_0 + a}\right]^{\alpha + i\beta}, \tag{9}$$

for $x > x_0$. The G filter is plotted in Figure 1.

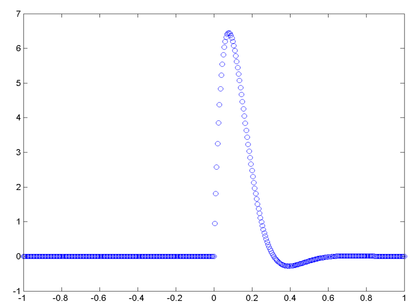

Fig. 1. Filter G plotted as functions of x, with $x_0 = 0$.

3.1. Implementation Issues

The practical implementation of the Green's function filtering process deserves some discussion. We wish to perform an integration such as (5), and thus, given a certain $\gamma = \tan\theta$, we must traverse the input image, computing weighted sums of the pixel intensities along the direction θ, where the weights will be the G or H values. Notice that, instead of trying to locate the actual pixels along the scan line, we uniformly sample points along that line, using a unit sample interval, and compute their intensities using bilinear interpolation.

Unless stated otherwise, we will assume that the filtering direction is horizontal ($\theta = 0°$), for simplicity. For the computation of $G(x, x_0)$, we will map the leftmost point in each image row to -0.5 and the rightmost point to 0.5 (thus, the midpoint in each row is mapped to zero).

4. VIDEO INTERPOLATION

In order to generate interpolating frames for a video sequence, we start by computing an affine optical flow between consecutive frames. We used the implementation provided by Black [8], based on affine regression, that computes parameters a_i, $i = 0, ..., 5$ relative to equation (3), where $a_0 = u_0$, $a_1 = D + S_1$, $a_2 = S_2 - R$, $a_3 = v_0$, $a_4 = R + S_2$ e $a_5 = D - S_1$. In order to use a one-dimensional affine flow model, we choose the direction of flow as the one given by vector (u_0, v_0) and take u as its norm and u' as the sum of the projections of the flow derivatives onto this direction. The interpolated frames are generated by applying the Green filter described in section 3 with parameters of the form ru and ru', where r is the proportion of movement desired for each frame.

5. EXPERIMENTAL RESULTS

Interpolated video sequences illustrating the application of our approach can be found at the web site http://www.impa.br/~perfeuge/green/video/index.html. Here we briefly discuss some general features of such experiments.

In some cases, the Green function filtering causes the images to be darkened. It also causes some amount of blur. In order to alleviate these undesired effects, we replace G in equation (5) by a linear combination of the form $\alpha_1 G(x, x_0) + \alpha_2 H(x, x_0)$. This works due to the fact that the filtering by H produces a homogeneous solution for the affine matching equation. As a consequence, the intensities of pixels at locations x near to x_0 tend to be preserved, since H peaks at that point and the blurring and/or darkening effect of G is attenuated.

Next, we will explain in more detail how the interpolated video sequences were generated. In order to evaluate the quality of the interpolation, we employed a measure of dissimilarity between a interpolating frame I_S and a frame of reference I_S, defined as

$$Error = \frac{\sum_{j=0}^{M-1} \sum_{k=0}^{N-1} \| I_R(j,k) - I_S(j,k) \|}{M.N.P}, \qquad (10)$$

where M is the image height, N is the image width and P is the number of gray levels in the image.

Camera Translation: The original video sequence was generated by translating the camera and corresponds to example 1 in our website. Eight intermediate frames were generated, based on the initial and final frames, by following the steps below: **(a)** The affine optical flow between the reference frames was computed, as explained in section 4. **(b)** The first four intermediate frames were generated by applying the Green function filter to the initial frame, with parameters of the form ru and ru', for r equal to $\frac{1}{7}$, $\frac{1}{6}$,$\frac{1}{5}$ e $\frac{1}{4}$. The last four frames were obtained by applying the filters of parameters $-ru$ and $-ru'$ to the final frame of the sequence. Figure 2 shows the two reference frames and a intermediary frame, generated by the Green filter with parameters $\theta = 0°$, $u = 0.011$, and $u' = -0.0000037$, that correspond to $r = \frac{1}{4}$ of movement. It should be noted that the amount of optical flow to be applied to the reference images must be kept small. For larger values, considerable blur can be noted. That is the case, for instance, of example 2 in our website,

Fig. 2. (Top) First frame from original sequence; (Middle) Interpolated frame with G-filter for $u = 0.011$, $u' = -0.0000037$, $\theta = 0°$, $\alpha_1 = 1.0$, and $\alpha_2 = 0.0$; (Bottom) Last frame from original sequence.

A second test was performed, with the purpose of evaluating if a image generated by a Green filter was a good approximation to an existing image. We compared a computed intermediate frame with a captured intermediate frame, and obtained a mean error equal to 0.035, that corresponds to a similarity of 97% between the computed and the captured frames.

Object Translation and Rotation: The original video sequence consists of an object translating and rotating in a otherwise static scene and corresponds to example 5 in our website. First, the image was manually segmented in order to establish the region of interest containing the moving object. Then, four intermediate frames were generated as in the previous example. The first two were obtained by applying the Green function filter to the initial frame, with parameters u and u' corresponding to $\frac{2}{7}$ and $\frac{3}{7}$ of the computed optical flow. For the last two, the same process was applied to the final frame. Figure 3 shows a sequence of three frames, where the initial and final frames are shown on the top and on the bottom, respectively. The middle frame is an intermediate frame generated using this process, with parameters $\theta = 0°$, $u = 0.00033$, and $u' = 0.0000034$ ($\alpha_1 = -0.1$ and $\alpha_2 = -0.3$), that correspond to $r = \frac{1}{3}$ of movement.

Fig. 3. (Top) First frame from original sequence; (Middle) Interpolated frame with G-filter with $u = 0.00033$, $u' = 0.0000034$, $\theta = 0°$, $\alpha_1 = -0.1$, and $\alpha_2 = -0.3$; (Bottom) Last frame from original sequence.

6. CONCLUDING REMARKS

We have presented a video interpolation approach based on the Green's function solution of an affine matching equation. By filtering a pair of reference frames through our matching kernels, we have been able to smooth a video sequence, by creating extra frames. Augmented videos produced this way have shown good results for sequences generated by camera translation, and by object translation or rotation in static scene.

The results obtained for the situation where the camera rotates around the object of interest are not as satisfatory, as shown in example 6 in our website. We conjecture that this is due to the limitations of the one-dimensional flow model that we employed. We plan to investigate whether better results can be obtained by using a two-dimensional Green filter. As another application of Green's function filters we are also envisioning the generation of virtual stereoscopic pairs.

Acknowledgements

The first author would like to thank Professor Michael Black for permit us to use his affine optical flow code [8]. We use the video sequences *pepsi* and 7*up* in Michael Black's web page. The first author also would like to thank Anderson Mayrink for the creation of some videos used here. This research has been developed in the VISGRAF Laboratory at IMPA.

7. REFERENCES

[1] W. Hinterberger and O. Scherzer, *Models of Image Interpolation Based on the Optical Flow*, Computing 66, 231-247, Springer-Verlag, 2001.

[2] S. E. Chen and L. Williams, *View Interpolation for Image Synthesis*, ACM Computer Graphics (Proceedings of SIGGRAPH'93) 27, 279-288, 1993.

[3] J. R. A. Torreão, *A Green's Function Approach to Shape from Shading*, Pattern Recognition 34, 2367-2382, 2001.

[4] P. E. Ferreira Jr., J. R. A. Torreão and P. C. P. Carvalho, *Data-Based Motion Simulation Through a Green's Function Approach*, Proceedings of XVII Brazilian Symposium on Computer Graphics and Image Processing, 193-199, 2004.

[5] C. Capurro, F. Panerai and G. Sandini, *Dynamic Vergence Using Log-Polar Images*, International Journal of Computer Vision 24(1), 79-94, 1997.

[6] B. Jahne, H. Hauβecker and P. Geiβler, *Handbook of Computer Vision and Applications*, Academic Press, 1999.

[7] M. J. Black and P. Anandan, *The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields*, Computer Vision and Image Understanding, CVIU, 63(1), 75-104, 1996.

[8] M. J. Black, Area-Based Optical Flow: Robust Affine Regression (Software available on-line at http://www.cs.brown.edu/ people/ black/).

[9] D.G. Zill and M.R. Cullen, Differential Equations with Boundary-Value Problems, PWS Publishing Company, 1993.