10

Motion and Motion Blur Through Green's Matrices

Perfilino E. Ferreira Júnior[1] and José R. A. Torreão[2]
[1]*Universidade Federal da Bahia*
[2]*Universidade Federal Fluminense*
Brazil

1. Introduction

Green's functions are powerful tools for solving differential equations. They arise in the resolution of boundary value problems (Stakgold, 1998) and heat diffusion equations (Folland, 1995), being also applied to economic modeling (Oppenheim & Willsky, 1996), and for deriving numerical approximations to integral equations (Jerri, 1999).

In the computer vision context, Green's functions of image matching equations have been introduced for solving the shape-from-shading problem (Torreão, 2001), and later used for edge detection (Torreão & Amaral, 2002; 2006), disparity estimation (Torreão, 2007), motion synthesis (Ferreira Júnior et al., 2008), and video interpolation (Ferreira Júnior et al., 2005).

If we consider a dynamic $3D$ scene imaged by a single camera, a pair of captured images, f_1 and f_2, can be related through the image matching equation (Jahne et al., 1999),

$$f_2(x+U, y+V) = f_1(x, y), \tag{1}$$

where U and V denote, respectively, the horizontal and vertical components of the optical flow. Several models have been used for describing the pair (U, V), among them the uniform model (which allows only for translational motions), the affine model (which incorporates planar rotation, shear, and dilation), and the projective model (which also incorporates perspective distortions).

Here, we will initially consider one-dimensional matching equations of the form

$$f_2(x+U) = f_1(x), \tag{2}$$

where $U \equiv U(x)$ follows the affine model $U(x) = u_0 + u_1 x$, for u_0 and u_1 constants. The $2D$ case will be tackled later. Expanding the left-hand side of Eq. (2) into a second-order Taylor series, we obtain the approximation

$$\frac{U^2(x)}{2} f_2^{''}(x) + U(x) f_2^{'}(x) + f_2(x) = f_1(x) \tag{3}$$

The solution to Eq. (3), via the Green's function method, can be obtained as

$$f_2(x) = f_1(x) \star G(x) = \int_{\mathcal{D}} G(x,\xi) f_1(\xi) d\xi \tag{4}$$

where $\mathcal{D}$ is the domain of interest, and where $G(x,\xi)$ is the Green's function, whose form, when considering an unbounded domain, has been obtained in (Ferreira Júnior et al., 2005), and employed for video interpolation. There, the possible existence of singular points of Eq. (3) – which are points where U(x) vanishes (Stakgold, 1998) – has not been considered, an issue that will be taken up here.

Motion synthesis with the simultaneous introduction of motion blur is another task which has been approached through the $1D$ affine Green's function model, proving it superior to competing techniques (Ferreira Júnior et al., 2008). Also, in (Ferreira Júnior et al., 2009), the second-order matching model of Eq. (3) has been compared to first-order variants, having been shown to generally yield more realistic motion effects.

In the present chapter, we propose to consider the following issues: i) that of improving our motion synthesis approach by using a matrix formulation, instead of a filtering one, to obtain the solutions to the affine matching equation; ii) that of solving the related problem of motion reversal. The chapter is organized as follows: in Section 2, we model the Green's function image matching problem, considering its discretization and its solution via the matrix approach. In Section 3, we apply the Green's function method to the problems of video interpolation and motion reversal. Finally, in Section 4, we present our concluding remarks.

2. Green's matrices

Let us start by considering some properties of the general matching equation (Eq. (2)), in the affine flow case. That equation can be formally rewritten as

$$f_2(x) = M_U [f_1](x), \tag{5}$$

where $U(x) = u_0 + u_1 x$, for u_0 and u_1 constants, with $u_1 \neq 0$ (the case $u_1 = 0$ reduces to the uniform flow, and will not be considered here), and where M_U denotes the linear transformation

$$M_U [f](x) = f\left(\frac{x - u_0}{1 + u_1}\right) \tag{6}$$

The affine vector field $U(x)$ vanishes at the point $x_U = \frac{-u_0}{u_1}$, for which we have $f_2(x) = f_1(x)$, i.e., a fixed point. Fig. 1 illustrates the behavior of the matching operator in the neighborhood of x_U. Its effect consists in the combination of a translation by $\frac{u_0}{1+u_1}$, and a scaling by $\frac{1}{1+u_1}$. When $u_1 > 0$, we obtain an expansion, and when $u_1 < 0$, a contraction results. Fig. 2 depicts the overall effect of applying M_U to the function $f_1(x) = \frac{sin(x^2)}{x}$. If we consider a second-order Taylor series expansion of the left-hand side of Eq. (2), we obtain the approximation in Eq. (3), whose solution can be expressed in terms of the Green's function, as in Eq. (4). This will be treated next.

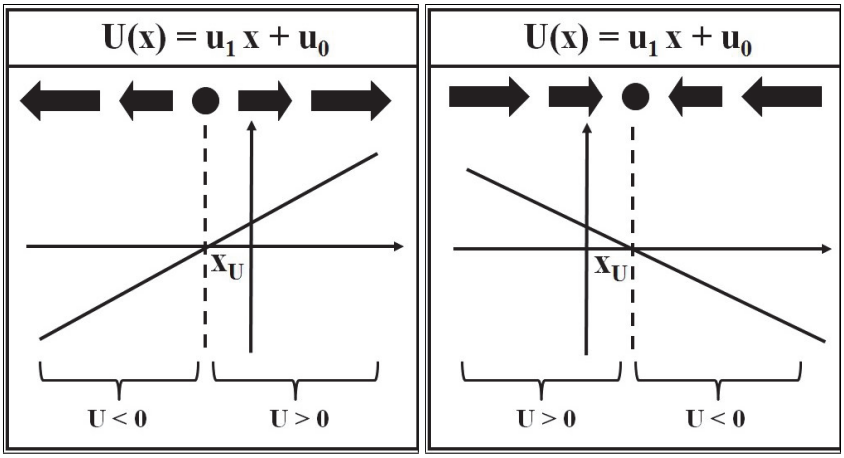


Fig. 1. Illustrating the behavior of the affine matching operator. Left: Expansion ($u_1 > 0$). Right: Contraction ($u_1 < 0$).

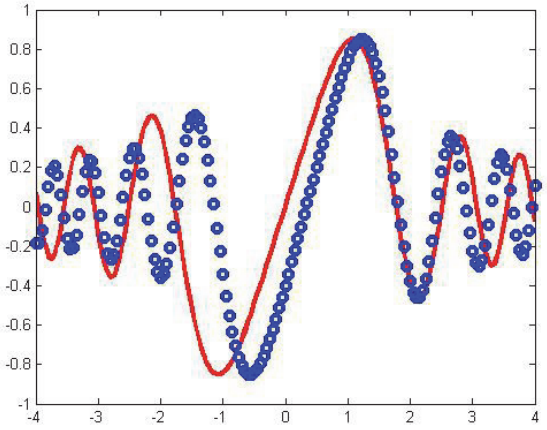


Fig. 2. Application of the operator M_U, with $(u_0, u_1) = (0.31, -0.155)$, to the function $f_1(x) = \frac{sin(x^2)}{x}$ (solid line). The circles represent the resulting signal, $f_2(x) = M_U[f_1](x) = \frac{sin\left(\frac{x-u_0}{1+u_1}\right)^2}{\left(\frac{x-u_0}{1+u_1}\right)}$. The fixed point, in this case, is at $x = 2$.

2.1 Continuous Green's functions

Assuming an unbounded domain $\mathcal{D}$, a limited Green's function can be obtained under two different guises. The form

$$G_+(x,\xi) = \frac{2}{u_1{}^2\beta(\xi - x_U)}\left[\frac{x-x_U}{\xi-x_U}\right]^{\alpha} \sin\left\{\beta \log\left[\frac{x-x_U}{\xi-x_U}\right]\right\}, \tag{7}$$

for $x > \xi$, with $G_+(x,\xi) = 0$, otherwise, will be bounded over the domain $\mathcal{D} \subset (x_U, +\infty)$, as long as we take $u_1 > 0$. On the other hand, a bounded Green's function over the domain $\mathcal{D} \subset (-\infty, x_U)$ can be obtained, assuming $u_1 < 0$, as

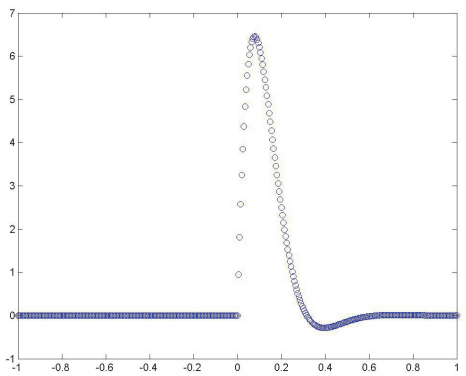
$$G_-(x,\xi) = \frac{2}{{u_1}^2 \beta (x_U - \xi)} \left[\frac{x_U - x}{x_U - \xi}\right]^{\alpha} \sin\left\{\beta \log\left[\frac{x_U - x}{x_U - \xi}\right]\right\}, \tag{8}$$

for $x < \xi$, with $G_-(x,\xi) = 0$, otherwise. In either case, the parameters α and β are given as

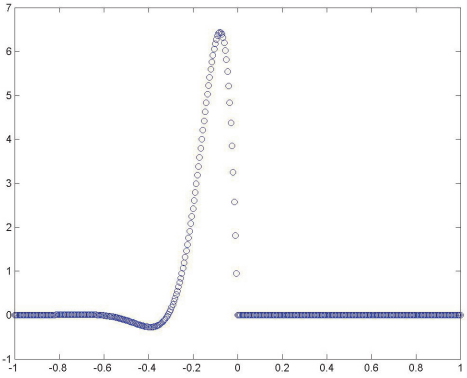
$$\begin{cases} \alpha = -\frac{1}{u_1} + \frac{1}{2} \\ \beta = \frac{1}{u_1}\sqrt{1 + u_1 - \frac{u_1^2}{4}} \end{cases} \tag{9}$$

Over finite domains, both forms will remain valid for $2(1-\sqrt{2}) < u_1 < 2(1+\sqrt{2})$. Fig. 3 shows plots of G_+ and G_-, and it can be noted that the Green's function forms are mirror reflections of one another about the axis $x = \xi$.

Fig. 4 illustrates the roles of the filters G_+ and G_- as approximations of the affine matching operator. For instance, let us consider the expansion case (left-hand panel). When considering a point x on the interval $(x_U, +\infty)$, we see that the value of the matching function $f_2(x)$ will depend on the values of $f_1(\xi)$ for all $\xi < x$. Each of these values will be weighted by the corresponding Green's function, $G_+(x,\xi)$, to yield $f_2(x)$ by the linear combination in Eq. (4). Similarly, the values of $f_1(\xi)$ for all $\xi > x$ will be weighted by $G_-(x,\xi)$, in order to yield $f_2(x)$ whenever $x \in (-\infty, x_U)$. The contraction case (right-hand panel in the figure) can be similarly treated. Fig. 5 illustrates the effects of the exact matching operator, M_U, and of its



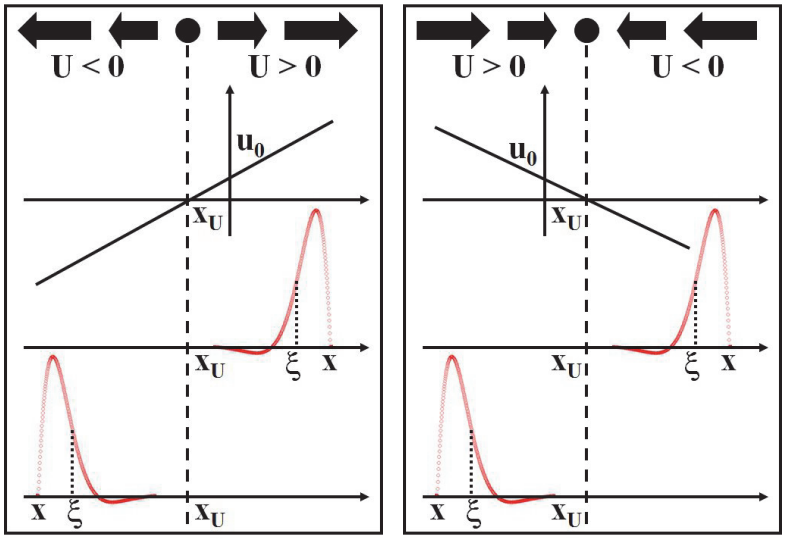
(a) G_+ filter plotted as function of x, for $\xi = 0$. Parameter values: $u_0 = 0.1$ and $u_1 = 0.005$.



(b) G_- filter plotted as function of x, for $\xi = 0$. Parameter values: $u_0 = -0.1$ and $u_1 = 0.005$.

Fig. 3. Plots of the G_+ and G_- filters.

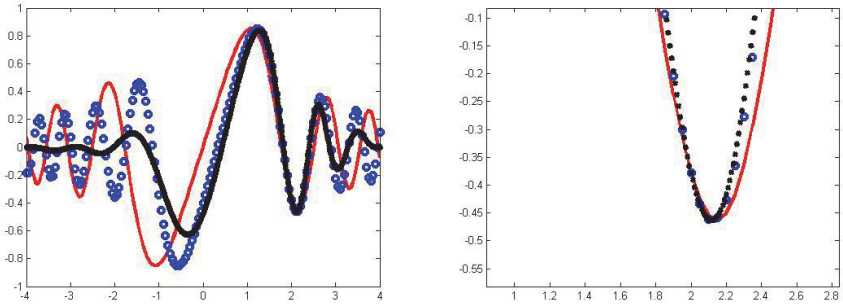
approximation by the Green's filters, $G_\pm$, when applied to the same test signal as considered in Fig. 2. The discretization of both operators will be discussed in the following subsection.



(a) Expansion, with $u_0 > 0$ and $u_1 > 0$: The central plot shows $G_+(x,\xi)$, and the lower plot shows $G_-(x,\xi)$, as functions of ξ, for a fixed x.

(b) Contraction, with $u_0 > 0$ and $u_1 < 0$: Similarly as in (a), but with G_+ in the lower plot and G_- in the central one.

Fig. 4. Illustrating the roles of the G_+ and G_- filters.



(a) Curves f_1, $M_U[f_1]$ and $f_1 \star G_\pm$

(b) The property of the fixed point

Fig. 5. Illustrating the action of the operator M_U, and of its approximations $G_\pm$. Plots of $f_1(x) = \frac{\sin(x^2)}{x}$ (solid line), $f_2(x) = M_U[f_1](x) = \frac{\sin\left(\frac{x-u_0}{1+u_1}\right)^2}{\left(\frac{x-u_0}{1+u_1}\right)}$ (circles), and $f_1 \star G_\pm$ (crosses). The parameter values were $u_0 = 0.31$ and $u_1 = -0.155$, leading to a fixed point at $x_U = 2$.

2.2 Discrete Green's functions

In the discrete case, Eq. (4) will reduce to the form

$$f_2(x) = \sum_k G(x,k) f_1(k) \tag{10}$$

where $G = G_{\pm}$. Since $x = x_U$ is a singular point of the Green's operator, we found it necessary to use stratified sampling for its discretization, with the sampling frequency increasing as we approach x_U (Ferreira Júnior et al., 2009). We consider this below.

2.2.1 Samples of the Green's function

We have adopted a non-weighted area sampling (Foley et al., 1995), for computing G at an image point. At each scanline, the area of a given pixel is partitioned along the horizontal direction, and the value of G is computed there. The discretized value $G(x,k)$ is then obtained by considering a Haar basis $\{\varphi(x)\}$, such that

$$G(x,k) = \sum_j G_j \varphi(k-j), \tag{11}$$

where k plays the role of an index inside the pixel at coordinate x, and where the coefficients G_j are computed as

$$G_j = \int_j^{j+1} G(x,\bar{\xi}) d\bar{\xi}. \tag{12}$$

Thus, the normalization condition

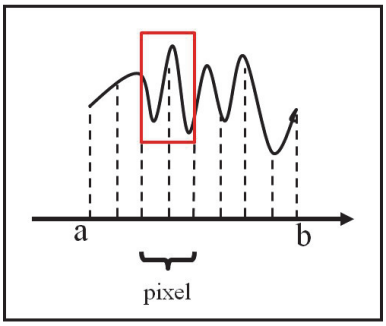
$$\int_{\mathcal{D}} G(x,\xi) d\xi = 1, \tag{13}$$

will also hold in the discrete case, where $\mathcal{D}$ is the considered domain. Fig. 6 illustrates the sampling process.

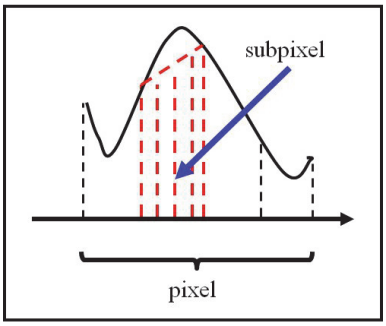
The stages for the computation of $G(x,k)$ can be summarized as follows:

1. At each scanline, an image pixel is partitioned into n subpixels (cells).
2. At the subpixel level, numerical integration is performed for determining the value of G_j (see Eq. (12)).
3. Each G_j value will then contribute to the expansion of $G(x,k)$, in terms of the Haar basis (see Eq. (11)).
4. A larger sampling frequency, $f_s^* > f_s$, is employed when $|U(x)| < T_U$.

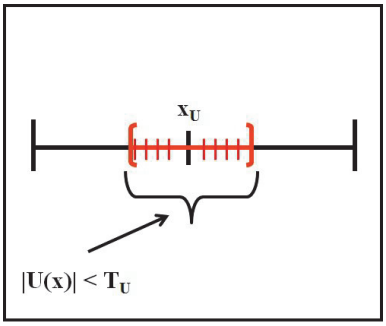
As we show next, the discretization of the Green's function becomes more efficient if a matrix representation is used. In this case, we need to compute the Green's function samples only once, and a single matrix product per line is required for yielding the transformed image. The complexity of the original filtering process, as described in (Ferreira Júnior et al., 2008; 2009) , is of order $O(M.N^2)$ per each image line, and since the filtering along the image columns is also required, we have a total complexity of order $O(M.N^2) + O(N.M^2)$. With the matrix approach, this will be reduced to $O(M) + O(N)$.



(a) An uniform partitioning is performed along a scanline.



(b) Each pixel is subdivided and sampled at a frequency f_s, contributing to the value G_j.



(c) The sampling frequency increases to f_s^* ($f_s^* > f_s$), when $|U(x)| < T_U$, where T_U is a threshold value.

Fig. 6. Unweighted area sampling of the Green's function

2.2.2 Constructing the Green's matrix

Here we undertake the construction of the Green's matrix for the expansion case, as in Fig. 4(a) (the contraction case can be similarly treated). Let us consider Eq. (4), with $g = f_2$ and $f = f_1$, and let us partition the intervals $(x_U, +\infty)$ and $(-\infty, x_U)$ in n subintervals, each of length $\Delta\xi = 1$. Thus, at each pixel x_j, Eq. (4) can be rewritten as

$$g(x_j) = \Delta\xi \left\{ \frac{1}{2} G_{\pm}(x_j, \xi_0) f(\xi_0) + G_{\pm}(x_j, \xi_1) f(\xi_1) + \cdots \right.$$
$$\left. + G_{\pm}(x_j, \xi_{j-2}) f(\xi_{j-2}) + G_{\pm}(x_j, \xi_{j-1}) f(\xi_{j-1}) \right\} \quad (14)$$

assuming $x_j > \xi_i > x_U$, for $\xi_i > \xi_{i+1}$ ($j = 1, 2, \cdots, n$ and $i = 0, 1, \cdots, j-1$), with $x_j \in (x_U, +\infty)$, when considering $g = G_+ \star f$, and $x_j < \xi_i < x_U$, for $\xi_i < \xi_{i+1}$ ($j = 1, 2, \cdots, n$ and $i = 0, 1, \cdots, j-1$), with $x_j \in (-\infty, x_U)$, when considering $g = G_- \star f$.

Thus, at each domain, $(x_U, +\infty)$ and $(-\infty, x_U)$, Eq. (14) yields $2n$ equations, n of which will be given as

$$\begin{cases} g_1 = \frac{1}{2} G_{10} f_0 \\ \\ g_j = \frac{1}{2} G_{j0} f_0 + G_{j1} f_1 + ... + G_{j,i-1} f_{i-1}, \end{cases} \tag{15}$$

where $g_j = g(x_j)$ $(j = 1, 2, \cdots, n)$, $f_i = f(\xi_i)$ $(i = 0, 1, ..., j-1)$ and $G_{ji} = G_+(x_j, \xi_i)$, for $x_j \in (x_U, +\infty)$. The other n equations will be similarly given, but for $G_{ji} = G_-(x_j, \xi_i)$, with $x_j \in (-\infty, x_U)$. In matrix form, this corresponds to the product

$$\begin{bmatrix} \frac{1}{2}G_{n,0} & G_{n,1} & \cdots & G_{n,n-1} & 0 & 0 & 0 & \cdots & 0 \\ 0 & \frac{1}{2}G_{n-1,0} & \cdots & G_{n-1,1} & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \frac{1}{2}G_{1,0} & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \frac{1}{2}G_{n+1,0} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & G_{n+2,1} & \frac{1}{2}G_{n+2,0} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & G_{2n,n+1} & G_{2n,n+2} & \cdots & \frac{1}{2}G_{2n,0} \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_{n-1} \\ f_U \\ \bar{f}_{n-1} \\ \bar{f}_{n-2} \\ \vdots \\ \bar{f}_0 \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \\ g_U \\ \bar{g}_n \\ \bar{g}_{n-1} \\ \vdots \\ \bar{g}_1 \end{bmatrix} \tag{16}$$

or, equivalently,

$$\mathcal{A}\hat{\mathbf{f}} = \hat{\mathbf{g}} \tag{17}$$

with

$$\mathcal{A} = \left[\begin{array}{c|c} \mathbf{G}_- & \bigcirc \\ \hline \bigcirc & \mathbf{G}_+ \end{array}\right] \tag{18}$$

being the Green's matrix, and the vectors $\hat{\mathbf{f}} = (f_0, \ldots, f_{n-1}, f_U, \bar{f}_{n-1}, \ldots, \bar{f}_0)^T$ and $\hat{\mathbf{g}} = (g_1, \ldots, g_n, g_U, \bar{g}_n, \ldots, \bar{g}_1)^T$ denoting, respectively, the lines of the input and of the output images. Note that $(f_0, f_1, \ldots, f_{n-1})^T$ will be the input pixels actuated on by G_- and $(\bar{f}_{n-1}, \bar{f}_{n-2}, \ldots, \bar{f}_0)^T$ will be those actuated on by G_+. Also note that the value 1 appearing at the center of the matrix $\mathcal{A}$ expresses the fixed point property

$$f(x_U) = g(x_U) \tag{19}$$

When implementing consecutive horizontal and vertical filterings, as in(Ferreira Júnior et al., 2009), separate kernels can be used, such that

$$\mathcal{B}\hat{\mathbf{f}} = \hat{\mathbf{g}} \tag{20}$$

where $\mathcal{B} = \mathcal{A}_Y.\mathcal{A}_X$, with $\mathcal{A}_X$ and $\mathcal{A}_Y$ implementing the filtering along the directions x and y, respectively. Thus, the whole computational cost will be that of performing the multiplication of each image line by the Green's matrix.

In the following section, we present the application of the above strategy to the problems of video interpolation and motion reversion.

3. Experimental results

Motion synthesis with the simultaneous generation of blur is obtained here by the use of Eq. (17), where $\mathcal{A}$ denotes the Green's matrix, and where we are considering $\hat{\mathbf{f}}$ and $\hat{\mathbf{g}}$ as image rows. In the interpolation of zoom sequences, Eq. (17) is applied twice, once over rows and once over columns.

3.1 Motion estimation and reconstruction quality

For the video interpolation experiments, we take the output of a motion estimation algorithm as input parameters to the Green's function, similarly as in (Ferreira Júnior et al., 2009). The general $2D$ affine model for motion estimation can be expressed as

$$\begin{bmatrix}\tilde{U}\\ \tilde{V}\end{bmatrix} = \begin{bmatrix}\tilde{u_0}\\ -\tilde{v_0}\end{bmatrix} + \begin{bmatrix}\tilde{u_1} & -\tilde{u_2}\\ \tilde{v_1} & \tilde{v_2}\end{bmatrix} \cdot \begin{bmatrix}x - c_x\\ y - c_y\end{bmatrix}, \tag{21}$$

where $(c_x, c_y)^T$ denotes the image center. In our interpolation experiments, we considered combined horizontal and vertical motions, such that our affine model will be separable, i.e.,

$$\begin{bmatrix}U\\ V\end{bmatrix} = \begin{bmatrix}u_0\\ v_0\end{bmatrix} + \begin{bmatrix}u_1 & 0\\ 0 & v_2\end{bmatrix} \cdot \begin{bmatrix}x\\ y\end{bmatrix} \tag{22}$$

Note that we are using the tilde in Eq. (21), but not in Eq. (22), to distinguish the *estimated* motion components from *generated* ones.

The signs of the components u_1 and v_2, in Eq. (22), allow the classification of the singular (fixed) points of the generated vector field (Verri et al., 1989). Thus, the singular point (x_U, y_V) (here, $x_U = -\frac{u_0}{u_1}$ and $y_V = -\frac{v_0}{v_2}$ denote the fixed points for the horizontal and the vertical motion components, respectively) will be a focus of expansion when $u_1, v_2 > 0$, a focus of contraction, when $u_1, v_2 < 0$, and a saddle point , if u_1 and v_2 have different signs. Comparing equations (21) and (22), we obtain

$$\begin{cases} u_0 = \tilde{u_0} - \tilde{u_1}c_x + \tilde{u_2}c_y \equiv u_0^* \\ u_1 = \tilde{u_1} \\ 0 = -\tilde{u_2} \\ v_0 = -\tilde{v_0} - \tilde{v_1}c_x - \tilde{v_2}c_y \equiv v_0^* \\ 0 = \tilde{v_1} \\ v_2 = \tilde{v_2} \end{cases} \tag{23}$$

For estimating the motion components from a given image sequence, we have used an optical flow algorithm based on affine regression, kindly made available to us by Professor Michael Black (Black & Anandan, 1996). Using the estimated components in the relations above, the Green's function parameters $(u_0, u_1, u_2, v_0, v_1, v_2)$ were then obtained. Similarly as in (Ferreira Júnior et al., 2005), we here usually work with fractions of the translation components, $(r.u_0, r.v_0)$, where r is the proportion of movement desired at each frame (e.g., $1/2$, $1/3$, etc.). As for the scaling components, u_1 and v_2, they are usually chosen as fractions of u_0 and v_0, respectively.

In what follows, we present results of the experimental analysis of the Green's matrix affine matching approach.

3.2 Results

We have considered here the effects of camera zoom-in and camera zoom-out (actually, both processes can be construed as a single one, just differing on the order of presentation of the frames). The Green's function approach will give rise to the fixed point

$$(x_U, y_V) = (-\frac{u_0}{u_1}, -\frac{v_0}{v_2}) \tag{24}$$

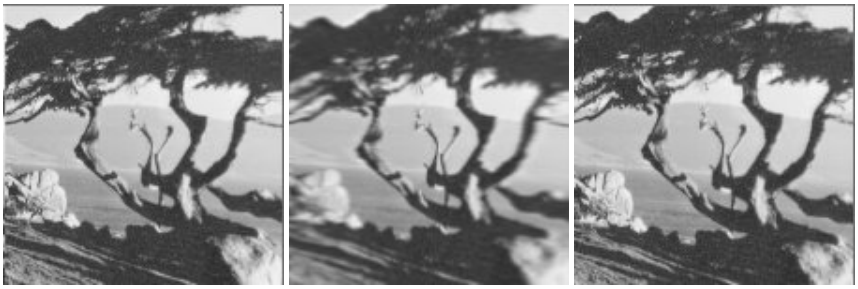
corresponding, respectively, to an expansion focus, in the zoom-in case, and to a contraction focus, in the zoom-out case.

3.2.1 Application 1: Video interpolation

Ideally, since we are considering an affine model instead of a full projective one, we should work with scenes depicting flat objects on a frontoparallel plane, such that there would be little change in depth. It is also known that depth discontinuities increase the complexity of the motion estimation process (Li, 2006).

The interpolated frames have here been obtained according to the following steps: **(a)** we estimate the affine optical flow between a pair of original frames; **(b)** from this, the corresponding Green's function components are calculated through Eq. (23); finally **(c)**, each interpolation frame is generated through Eq. (17).

The example illustrated in Fig. 7 corresponds to a synthetic motion sequence, over which we performed a zooming-in operation. It should be noted that the intermediate, interpolated frame in the figure presents appreciable blur, as expected from the Green's function approach. The full sequence appears as Example 1 in our website (`http://www.graphics.ufba.br/intechpage`), and simulates an optical zoom operation.



(a) A frame of reference - 150 × 150 pixels.

(b) Interpolated frame with our Green's Matrix from the reference image in (a).

(c) A second frame of - 150 × 150 pixels.

Fig. 7. Motion parameters were $(u_0, u_1, v_0, v_2) = (-3, 0.040, -1, 0.013)$. A focus of expansion appears at $(x_U, y_V) = (75, 75)$.

Table 1 presents the optical flow components estimated from the reference frames depicted in Fig. 7(a) and (c). Fig. 8 presents a different pair of frames from the same video considered

Motion Estimation Parameters					
u_0	u_1	u_2	v_0	v_1	v_2
6.226374	0.009514	0.005162	−1.119647	−0.014769	−0.014783

Table 1. Optical flow components obtained through Black's algorithm (Black & Anandan, 1996).

above, along with the interpolated frame now obtained through a zooming-out operation. The full generated sequence also appears, as Example 2, in our website.

(a) A frame of reference - 150 × 150 pixels.

(b) Interpolating frame, obtained from the image in (c) through the Green's Matrix approach.

(c) A second frame of reference - 150 × 150 pixels.

Fig. 8. Motion parameters were $(u_0, u_1, v_0, v_2) = (2, -0.027, 3, -0.040)$. A focus of contraction appears at $(x_U, y_V) = (75, 75)$.

Table 2 presents the optical flow components estimated from the reference frames depicted in Fig. 8(a) and (c). We also performed an interpolation test with a fragmented sequence

Motion Estimation Parameters					
u_0	u_1	u_2	v_0	v_1	v_2
−0.112766	0.022996	−0.007487	2.169647	−0.013922	0.054095

Table 2. Optical flow components obtained through Black's algorithm (Black & Anandan, 1996).

from a real video (an old Three Stooges picture). It appears in Fig. 9 , and as Example 3 in our website. In the original sequence, the camera recedes, leading to a sudden discontinuity between frames (see the website), due to the appearance of new objects in the field of view. The net visual effect is that of an elastic deformation of the scene, which in this case is much more complex than in the previous examples, and not very amenable to our affine approach.

Table 3 presents the optical flow components obtained from the reference frames in Fig. 9(a) and (c).

(a) A frame of reference - 280 × 210 pixels.

(b) Interpolating frame, obtained from the image in (a) through the Green's Matrix approach.

(c) A second frame of reference - 280 × 210 pixels.

Fig. 9. Motion parameters were $(u_0, u_1, v_0, v_2) = (-0.015, 0.000, -3, 0.029)$. A focus of expansion appears at $(x_U, y_V) = (140, 105)$.

Motion Estimation Parameters					
u_0	u_1	u_2	v_0	v_1	v_2
-0.142945	-0.002325	-0.001457	-6.492826	-0.009811	0.026797

Table 3. Optical flow components obtained through Black's algorithm (Black & Anandan, 1996).

Our next experiment is based on a sequence created by Chad Soriano, which can be found at `http://www.chadsorianophotoblog.com/2011_03_11_archive.html`. It presents a more favorable situation for the application of the Green's function approach, by showing a single object on a dark background. In Fig. 10, we display two reference frames in the sequence, and the interpolated one.

(a) A frame of reference - 240 × 131 pixels.

(b) Interpolated frame with our Green's Matrix from the image in (a).

(c) A second frame of reference - 240 × 131 pixels.

Fig. 10. Motion parameters were $(u_0, u_1, v_0, v_2) = (2, -0.008, 1, -0.008)$. A focus of contraction appears at $(x_U, y_V) = (120, 66)$.

Table 4 presents the optical flow components obtained from the reference frames of Fig. 10(a) and (c).

Next, we will illustrate the use of the Green's matrix approach for the removal of motion effects.

Motion Estimation Parameters					
u_0	u_1	u_2	v_0	v_1	v_2
−10.093081	0.016620	−0.000431	0.104955	−0.002782	0.106368

Table 4. Optical flow components obtained through Black's algorithm (Black & Anandan, 1996).

3.2.2 Application 2: Motion reversion

Here, as a means of validation of the obtained results, the mean reconstruction error and the similarity measure, defined respectively as

$$\epsilon = \frac{\sum_{j=1}^{M-1}\sum_{k=1}^{N-1}\|f(j,k) - f_R(j,k)\|}{M.N.P} \tag{25}$$

and

$$(1-\epsilon) * 100\% \tag{26}$$

have been used for assessing the quality of the restorations. In the first equation, f denotes the original image, f_R is its restored version, M and N stand for image width and height, respectively, and P for the number of gray levels.

Fig. 11 (also Example 5 in our website) shows an example of reversal of motion effects.

(a) The original input - 270 × 250 pixels.

(b) The result of applying the Green's matrix over the original input.

(c) The result of motion reversal.

Fig. 11. Motion Reversal Experiment. Here the Green's Matrix has been applied to each image line. Motion parameters for generating (b) were $(u_0, u_1) = (-7, 0.052)$. A focus of expansion appears at $x_U = 135$.

In this case, through the Green's matrix approach, the image in Fig. 11 (a) underwent an expansion along the x direction (we used $u_1 > 0$), generating the image in Fig. 11(b). Next, both images were used for estimating the optical flow, and for computing the Green's matrix parameters as in Section 3.1. We finally obtained the inverse to that matrix and used it to generate Fig. 11(c). As shown, this has the effect of removing most of the blur and motion components. The quality measures of the reconstructed image are presented in Table 5. We also performed another test, where we reversed the Green's matrix separately, in the horizontal and in the vertical directions. Fig. 12 (Example 6 in our website) shows more

Quality Measures		
Figures	Mean error	Similarity (%)
Figs. 11(a) and (c)	0.089303	91.07

Table 5. Mean error and similarity results.

results of reversal of motion effects. The image of Fig. 12(b) was obtained by consecutively

(a) The original input - 280×210 pixels.

(b) The result of applying the Green's matrix over the original input.

(c) The result of motion reversal.

Fig. 12. Motion Reversal Experiment. The motion parameters were empirically chosen as $(u_0, u_1, v_0, v_2) = (0.03, -0.0002, 6.5, -0.062)$. A focus of contraction appears at $(x_U, y_V) = (140, 105)$.

applying the Green's matrix over the lines and columns of Fig. 12(a) (the matrix parameters were empirically chosen). The restored image in Fig. 12(c) was then generated by inverting both processes. It can be seen that the motion and blur components are substantially reduced, but the edges are substantially enhanced. Also, there appear some artifacts at the image boundary, which are probably due to boundary conditions which have not been appropriately considered here (Hansen et al., 2006) . Table 6 shows the quality measures of the restoration.

Quality Measures		
Figures	Mean error	Similarity (%)
Figs. 12(a) and (c)	0.046797	95.32

Table 6. Mean error and similarity results.

4. Conclusions and future works

Here we have proposed a matrix formulation for the Green's function affine matching approach. Such formulation leads to reduced computational costs, and affords application to complex problems, such that of interpolating video sequences including camera zoom-in and camera zoom-out. We have also proposed an inverse filtering scheme for the reversal of motion effects. This has so far yielded promising results, allowing the partial removal of both motion and blur components introduced by the Green's function itself. We are now testing its application to real motion sequences.

As future developments of our work, we envision a better treatment of the boundary problem associated with the Green's function, and also its regularization. A possible application could lie in super-resolution for synthetic zooming. The robustness under noise of our approach should also be assessed.

5. References

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

Ferreira Júnior, P., E.; Torreão, J., R., A.; Carvalho, P.,C.,P.; and Velho, L. (2005). Video Interpolation Through Green's Functions of Matching Equations, *Proc. of IEEE International Conference on Image Processing*, Vol. 3, pp. III-1080-3. doi:10.1109/ICIP.2005.1530583 Key: citeulike:2804665

Ferreira Júnior, P.,E.; Torreão, J.,R.,A.; Carvalho, P.,C.,P.; and Vieira, M.,B. (2008). Motion Synthesis Through 1D Affine Matching, *Pattern Analysis and Applications* 11, pp. 45-58

Ferreira Júnior, P.,E.; Torreão, J.,R.,A.; and Carvalho, P., C., P. (2009). A Comparative Analysis of Green's Functions of 1D Matching Equations for Motion Synthesis, *Pattern recognition Letters* 30(14),pp. 1321–1334

Foley, J.; Van Dam, A.; Feiner, S.; Hughes, J. (1995). Computer Graphics: Principles and Practice in C, *Addison-Wesley Professional*, 2^{nd} Edition.

Folland, G., B. (1995). Introduction to Partial Differential Equations, *Princeton University Press*, 2^{nd} edition

Hansen, P. C.; Nagy, J,; O'Leary, D. (2006). Deblurring Images: Matrices, Spectra, and Filtering, *Society for Industrial and Applied Mathematics*, 1^{st} edition

Jahne, B.; Hausseker, H.; Geissler, P. (1999). In: Handbook of Computer Vision and Applications, Vol. 2, *Academic Press*

Jerri, A. (1999). Introduction to Integral Equations with Applications, *Wiley-Interscience*, 2^{nd} edition

Li, X. (2005). Super-Resolution for Synthetic Zooming, *Journal on Applied Signal Processing*, Volume 2006, Article ID 58195, pp. 1–12, DOI 10.1155/ASP/2006/58195

Oppenheim, A.,V.; and Willsky, A.,S. (1996). Signals and Systems, *Prentice Hall*, 2^{nd} edition

Sondhi, M. (1972). The Removal of Spatially Invariant Degradations, *Proc. of IEEE* 60(7), pp. 842–853

Stakgold, I. (1998). Green's Functions and Boundary Value Problems, *A Wiley-Interscience Pub.*, 2^{nd} edition

Torreão, J., R.,A. (2001). A Green's Function Approach to Shape from Shading, *Pattern Recognition* 34, pp. 2367-2382

Torreão, J.,R.,A. and Amaral, M.,S. (2002). Signal Differentiation Through a Green's Function Approach, *Pattern Recognition Letters* 23(14), pp. 1755–1759

Torreão, J.,R.,A. and Amaral, M.,S. (2006). Efficient, recursively implemented differential operator, with application to edge detection, *Pattern Recognition Letters* 27(9), pp. 987–995

Torreão, J.,R.,A. (2007). Disparity Estimation Through Green's functions of Matching Equations, *Biol. Cybernetics* 97, pp. 307–316

Verri, A.; Girosi, F.; Torre, V. (1989). Mathematical Properties of the 2*D* Motion Field: from Singular Points to Motion Parameters, *Journal of the Optical Society of America A* 6(5), pp. 698–712