

# 基于 CLG 光流算法的云的运动分析与研究

史金龙,白素琴,施金宛,袁辰辰

(江苏科技大学 计算机科学与工程学院,江苏 镇江 212003)

**摘要:**将光流算法应用到云的运动分析中,同时探讨光流变化与云的运动之间的关系。文中采用了局部与全局(CLG)相结合的光流算法分析云的运动。CLG算法同时具备局部光流算法和全局光流算法的优点,利用CLG光流算法能得到鲁棒而且稠密的云的运动流场。文中首先详细分析了CLG三种光流算法:空间CLG、时空CLG、非线性CLG光流算法。然后将这三种算法应用到云的运动视频中,并对三种CLG光流算法在云的运动上进行了分析和比较。实验证明,光流算法对于测量云的运动有良好的效果,云的运动与光流之间具有正比关系。

**关键词:**云;运动;光流

**中图分类号:**TP391

**文献标识码:**A

**文章编号:**1673-629X(2011)12-0135-04

## Analysis and Research of Cloud Motion Based on CLG Optical Flow

SHI Jin-long, BAI Su-qin, SHI Jin-wan, YUAN Chen-chen

(School of Computer Science and Engineering, Jiangsu University of Science and Technology,  
Zhenjiang 212003, China)

**Abstract:** Introduce the optical flow algorithm to cloud motion analysis, and the relationship between optical flow change and cloud motion is discussed. CLG algorithm is adopted to analyze the cloud motion. CLG algorithm has the advantages of both local optical flow and global optical flow, and robust and dense flow can be achieved by CLG. It first elaborates the three CLG algorithms: spatial CLG, spatio-temporal CLG and nonlinear CLG. Then the three CLG algorithms are applied into cloud motion, and analysis and comparison is performed. The experiments illustrate that it has good effect to use optical flow algorithms in cloud motion analysis.

**Key words:** cloud; motion; optical flow

## 0 引言

中国有句俗语“看云识天气”,可见云与天气的变化存在一定的联系。很多的气象学者都在从事研究云与天气变化的关系,特别是在一些灾难性天气多发的国家,例如日本就投入了大量的人力和物力资源用于研究灾难性的天气。近些年来,计算机视觉领域的技术的进步为分析云的运动提供了很好的方法,一些国外学者试图将计算机视觉的知识应用于分析云的运动,从而进一步分析天气的变化<sup>[1,2]</sup>。文中详细介绍了光流算法,并尝试将其应用于云的运动分析中。

## 1 光流算法介绍

光流是指图像中像素的表面运动速度,理想的光

流应该是物体的空间速度在图像上的投影。光流的应用非常的广泛,利用光流可以进行图像分割、图像对齐、目标跟踪、三维重建等。光流计算是通过估计图像的时间和空间导数实现的,这些技术可以被分成两类,一类是局部方法,另一类是全局方法。局部方法是通过优化一些局部的类似能量的表达式来实现的,典型代表方法包括:1981年Lucas-Kanade提出的LK方法<sup>[3]</sup>、1984年Lucas<sup>[4,5]</sup>、1988年Bigun和Granlund提出的结构张量的方法<sup>[6]</sup>,以及1991年Bigun提出的方法<sup>[7]</sup>;全局方法是通过最小化全局的能量函数来实现的,典型代表方法包括1981年Horn和Schunck提出的HS方法<sup>[8]</sup>等。

局部方法对噪声具有很好的抗干扰能力,但是这种方法不能得到稠密的光流;相反全局的方法可以得到100%密度的光流,但是对噪声非常的敏感。为了消除噪声,使求导过程稳定,往往在求导之前对图像进行滤波,在局部方法中,例如Lucas-Kanade方法中假设图像像素移动很小,所以在这个很小的范围内可以

收稿日期:2011-05-30;修回日期:2011-09-04

基金项目:国家自然科学基金(51008143);江苏省汽车工程重点实验室开放基金项目(QC201005)

作者简介:史金龙(1976-),男,黑龙江宾县人,讲师,博士研究生,主要研究方向为计算机视觉。

认为灰度保持不变,因此其导数为零;在全局方法中往往使用一个平滑约束来达到同样的目的。CLG(Combining Local and Global)光流算法将局部方法 LK 算法嵌入到全局方法 HS 算法中。其中 LK 和 HS 算法的函数分别如式(1)和(2)所示<sup>[3,4]</sup>:

$$E_{LK}(u, v) = K_p * ((f_x u + f_y v + f_t)^2) \quad (1)$$

$$E_{HS}(u, v) = \int_{\Omega} ((f_x u + f_y v + f_t)^2 + \alpha(|\nabla u|^2 + |\nabla v|^2)) dx dy \quad (2)$$

CLG 算法具有局部方法的对噪声具有高鲁棒性以及全局方法的可以获取 100% 稠密光流的特点<sup>[9-12]</sup>。

这种算法包括空间、时空、非线性等方法,下面将分别介绍这些算法思想及其实现。

### 1.1 空间方法(Spatial Approach)

首先介绍一些符号:

$$w: = (u, v, 1)^T \quad (3)$$

$$|\nabla w|^2: = |\nabla u|^2 + |\nabla v|^2 \quad (4)$$

$$\nabla f: = (f_x, f_y, f_t)^T \quad (5)$$

$$J_p(\nabla f) = K_p * (\nabla f \nabla f^T) \quad (6)$$

然后,将(1)式写成(7)式,将(2)式改成(8)式。

$$\begin{aligned} E_{LK}(u, v) &= K_p * ((f_x u + f_y v + f_t)^2) \\ &= K_p * ((f_x u + f_y v + f_t)(f_x u + f_y v + f_t)) \\ &= K_p * \{[(u, v, 1)(f_x, f_y, f_t)^T][(f_x, f_y, f_t)(u, v, 1)^T]\} \\ &= K_p * \{(u, v, 1)[(f_x, f_y, f_t)^T(f_x, f_y, f_t)](u, v, 1)^T\} \\ &= K_p * w^T(\nabla f \nabla f^T)w = w^T * K_p(\nabla f \nabla f^T)w \\ &= w^T J_p(\nabla f)w = E_{LK}(w) \end{aligned} \quad (7)$$

$$\begin{aligned} E_{HS}(w) &= \int_{\Omega} ((f_x u + f_y v + f_t)^2 + \alpha(|\nabla u|^2 + |\nabla v|^2)) dx dy \\ &= \int_{\Omega} (w^T J_p(\nabla f)w + \alpha|\nabla w|^2) dx dy \end{aligned} \quad (8)$$

最后,将矩阵(8)式中的  $J_p(\nabla f)$  替换成(7)式中的  $J_p(\nabla f)$ ,  $\rho > 0$ , 便得到(9)式的 CLG 函数:

$$E_{CLG}(w) = \int_{\Omega} (w^T J_p(\nabla f)w + \alpha|\nabla w|^2) dx dy \quad (9)$$

(9)式的光流  $(u, v)$  应该满足 Euler-Lagrange 方程:

$$0 = \Delta u - \frac{1}{\alpha}(K_p * (f_x^2)u + K_p * (f_x f_y)v + K_p * (f_y f_x)) \quad (10)$$

$$0 = \Delta v - \frac{1}{\alpha}(K_p * (f_x f_y)u + K_p * (f_y^2)v + K_p * (f_x f_t)) \quad (11)$$

其中  $\alpha > 0$  是平滑权重,较大的  $\alpha$  值会减小光流梯度,从而使光流更加平滑。

### 1.2 时空的方法(Spatiotemporal Approach)

空间方法只考虑了空间上的滤波操作,为了达到更好的效果,文献[5]进一步考虑时间轴,将 2D 的高斯卷积改成 3D 的高斯卷积,便得到(12)式的方程,

$$E_{CLG3}(w) = \int_{\Omega \times [0, T]} (w^T J_p(\nabla_3 f)w + \alpha|\nabla_3 w|^2) dx dy dt \quad (12)$$

其中  $|\nabla_3 w|^2 = |\nabla_3 u|^2 + |\nabla_3 v|^2$ ,  $\nabla_3 u = (\partial u / \partial x, \partial u / \partial y, \partial u / \partial t)^T$ ,  $\nabla_3 v = (\partial v / \partial x, \partial v / \partial y, \partial v / \partial t)^T$ 。在时空高斯卷积中,在时间方向和空间方向上的高斯标准差可以不同。文中在计算时,先对  $x$  方向求卷积,然后在  $y$  方向求卷积,最后在时间方向求卷积。

设  $J_{nm}$  表示  $J_p(\nabla_3 f)$  的第  $n$  行  $m$  列个元素, 式的 Euler-Lagrange 方程表示如下:

$$\Delta_3 u - \frac{1}{\alpha}(J_{11}u + J_{12}v + J_{13}) = 0 \quad (13)$$

$$\Delta_3 v - \frac{1}{\alpha}(J_{12}u + J_{22}v + J_{23}) = 0 \quad (14)$$

式(13)、(14)中使用了时空高斯卷积,以及时空的 Laplacean 算子,时空的 Laplacean 如(15)式

$$\Delta_3 = \partial_{xx} + \partial_{yy} + \partial_{tt} \quad (15)$$

### 1.3 非线性的方法

空间方法和时空方法都是基于对二次函数优化的线性方法,在光流不连续的地方,这两种方法的效果较差。为了得到更好的效果,文献[5]等提出将(9)式, (12)式改为非二次函数形式,使其变为非线性方法。(16)式为非线性函数:

$$E_{CLG3-N}(w) = \int_{\Omega \times [0, T]} (\psi_1(w^T J_p(\nabla_3 f)w) + \alpha\psi_2(|\nabla_3 w|^2)) dx dy dt \quad (16)$$

其中,  $\psi_1(s^2)$  和  $\psi_2(s^2)$  是惩罚函数,文中使用 Charbonnier 在 1994 年提出的函数:

$$\psi_i(s^2) = 2\beta_i^2 \sqrt{1 + \frac{s^2}{\beta_i^2}}, i \in 1, 2 \quad (17)$$

其中  $\beta_1$  和  $\beta_2$  是尺度参数。(16)式的 Euler - Lagrange 方程如下:

$$0 = \text{div}(\psi_2'(|\nabla_3 w|^2) \nabla_3 u) - \frac{1}{\alpha} \psi_1'(w^T J_p(\nabla_3 f)w) (J_{11}u + J_{12}v + J_{13}) \quad (18)$$

$$0 = \text{div}(\psi_2'(|\nabla_3 w|^2) \nabla_3 v) - \frac{1}{\alpha} \psi_1'(w^T J_p(\nabla_3 f)w) (J_{12}u + J_{22}v + J_{23}) \quad (19)$$

## 2 算法实现及实验结果分析

### 2.1 空间与时空的算法实现

本节介绍计算光流分量  $u(x, y, t)$  和  $v(x, y, t)$  的实现方法:假设在图像上大小为  $h$  的方框内,其中

$u_{i(i=1,\dots,N)}$  是其中的某个像素点  $i$  上的  $u(x,y,t)$  的估计值,  $v_i$  是像素点  $i$  上的  $v(x,y,t)$  估计值。用  $J_{nmi}$  表示第  $i$  个像素点的  $J_\rho(\nabla f)$  的第  $n$  行  $m$  列元素;用  $N(i)$  表示像素点  $i$  的邻居的集合,其中邻居个数:在空间方法中是 4 个,在时空的方法中是 6 个。(10)和(11)式的 Euler - Lagrange 方程可以使用下式进行估计:

$$0 = \sum_{j \in N(i)} \frac{u_j - u_i}{h^2} - \frac{1}{\alpha} (J_{11i}u_i + J_{12i}v_i + J_{13i}) \quad (20)$$

$$0 = \sum_{j \in N(i)} \frac{v_j - v_i}{h^2} - \frac{1}{\alpha} (J_{21i}u_i + J_{22i}v_i + J_{23i}) \quad (21)$$

其中,  $i = 1, \dots, N$ 。这个方程组可以使用迭代的方式进行求解,文中将此方程组用 SOR (successive overrelaxation) 方法<sup>[11]</sup>改写为(22)、(23)式的形式,在实际的计算中,用(22)、(23)进行迭代即可。

$$u_i^{k+1} = (1 - \omega)u_i^k + \omega \frac{\sum_{j \in N^-(i)} u_j^{k+1} + \sum_{j \in N^+(i)} u_j^k - \frac{h^2}{\alpha} (J_{12i}v_i^k + J_{13i})}{|N(i)| + \frac{h^2}{\alpha} J_{11i}} \quad (22)$$

$$v_i^{k+1} = (1 - \omega)v_i^k + \omega \frac{\sum_{j \in N^-(i)} v_j^{k+1} + \sum_{j \in N^+(i)} v_j^k - \frac{h^2}{\alpha} (J_{21i}u_i^{k+1} + J_{23i})}{|N(i)| + \frac{h^2}{\alpha} J_{22i}} \quad (23)$$

其中:

$$N^-(i) = \{j \in N(i) \mid j < i\} \quad (24)$$

$$N^+(i) = \{j \in N(i) \mid j > i\} \quad (25)$$

文中具体实现的步骤如下:

1) 选取初始参数,  $\omega$ 、 $\alpha$ 、 $h$ , 文中采用  $\omega = 1.5$ ,  $\alpha = 150$ ,  $h = 3$ 。

2) 给定一个图像  $g(x,y,t)$ , 使用高斯核函数  $K_\sigma$  进行滤波, 得到滤波之后的图像  $f(x,y,t)$ 。

3) 对  $f(x,y,t)$  使用 sobel 算子求  $x$  方向、 $y$  方向的导数;利用两帧中对应像素相减的方式求  $t$  方向的导数, 得到三张导数图像  $f_x, f_y, f_t$ , 文中所用的 sobel 模版如图 1 所示。

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

图 1 sobel 模版(分别是计算  $x$ 、 $y$  方向导数的模版)

4) 再计算  $(f_x)^2 f_y, (f_y)^2 f_x, f_x f_y$ , 形成 5 个新的图像, 然后分别对这 5 幅图像用高斯核  $K_\rho$  进行滤波。

5) 根据公式:

$$J_\rho(\nabla_3 f)$$

$$= K_\rho * (\nabla_3 f \nabla_3 f^T)$$

$$= K_\rho * \begin{bmatrix} (f_x)^2 & f_x f_y & f_x f_t \\ f_x f_y & (f_y)^2 & f_y f_t \\ f_x f_t & f_y f_t & (f_t)^2 \end{bmatrix}$$

计算每个向素点  $i$  的  $J_\rho(\nabla_3 f)$ 。

6) 创建两张大小与  $g(x,y,t)$  相同的图片  $u_i$  和  $v_i$ , 将  $u_i$  和  $v_i$  的初始值设为 0。

7) 根据(22)、(23)式迭代计算  $u_i$  和  $v_i$ 。

## 2.2 非线性方法

假设在像素点  $i$  处的惩罚函数  $\psi_1$  和  $\psi_2$  的导数分别用  $\psi'_{1i} = \psi'_1(w'_i J_{\rho i} w_i)$  和  $\psi'_{2i} = \psi'_2(|\nabla_3 w_i|^2)$  表示。文献[5]得到(26)和(27)式的离散化表示方法:

$$0 = \sum_{j \in N(i)} \frac{\psi'_{2i} + \psi'_{2j}}{2} \frac{u_j - u_i}{h^2} - \frac{\psi'_{1i}}{\alpha} (J_{11i}u_i + J_{12i}v_i + J_{13i}) \quad (26)$$

$$0 = \sum_{j \in N(i)} \frac{\psi'_{2i} + \psi'_{2j}}{2} \frac{v_j - v_i}{h^2} - \frac{\psi'_{1i}}{\alpha} (J_{21i}u_i + J_{22i}v_i + J_{23i}) \quad (27)$$

求解此方程组的迭代模型如下:

$$u_i^{k+1} = (1 - \omega)u_i^k + \omega \left[ \sum_{j \in N^-(i)} \frac{\psi'_{2i} + \psi'_{2j}}{2} u_j^{k+1} + \sum_{j \in N^+(i)} \frac{\psi'_{2i} + \psi'_{2j}}{2} u_j^k - \psi'_{1i} \frac{h^2}{\alpha} (J_{12i}v_i^k + J_{13i}) \right] / \left[ \sum_{j \in N(i)} \frac{\psi'_{2i} + \psi'_{2j}}{2} + \psi'_{1i} \frac{h^2}{\alpha} J_{11i} \right] \quad (28)$$

$$v_i^{k+1} = (1 - \omega)v_i^k + \omega \left[ \sum_{j \in N^-(i)} \frac{\psi'_{2i} + \psi'_{2j}}{2} v_j^{k+1} + \sum_{j \in N^+(i)} \frac{\psi'_{2i} + \psi'_{2j}}{2} v_j^k - \psi'_{1i} \frac{h^2}{\alpha} (J_{21i}u_i^{k+1} + J_{23i}) \right] / \left[ \sum_{j \in N(i)} \frac{\psi'_{2i} + \psi'_{2j}}{2} + \psi'_{1i} \frac{h^2}{\alpha} J_{22i} \right] \quad (29)$$

文中在具体实现此算法时,除了要计算惩罚函数  $\psi_1$  和  $\psi_2$  外,其他步骤与空间与时空的算法实现过程类似。

## 2.3 实验结果及参数分析

文中利用 VC++2005 和 OpenCV1.1 开发了一个分析云的运动软件系统。本系统实现了空间 CLG、时空 CLG、非线性 CLG 光流算法,并将算法应用于四个云的运动视频,视频中每帧图像的大小是  $640 * 480$ 。

图 2 和图 3 分别为 CLG 空间法和非线性法光流的运行截屏,为了看得清楚,在截屏上只显示了大于某个域值的部分光流信息。

实验结果显示:在三种 CLG 算法中,非线性 CLG 得到的结果最佳。值得注意的是  $\omega$ 、 $\alpha$  初始值的选取对于计算光流有很大的影响,  $\omega$  的值会影响软件的收

敛速度,图4是本系统所测得的 $\omega$ 值与收敛速度曲线,可见 $\omega$ 值越大,收敛越慢。 $\alpha$ 的值会影响光流的平滑程度, $\alpha$ 的值越大,光流越平滑。

动速度,并将其作为天气变化分析的一个重要因素。

#### 参考文献:

- [1] Lin Zhou, Kambhamettu C, Goldgof D B, et al. Tracking nonrigid motion and structure from 2D satellite cloud images without correspondences [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001, 23(11):1330-1336.
- [2] Cohen I, Herlin I. Optical flow and phase portrait methods for environmental satellite image sequences [C]//ECCV '96. [s. l.]: [s. n.], 1996:141-150.
- [3] Lucas B, Kanade T. An iterative image registration technique with an application to stereo vision [C]//Proc. Seventh International Joint Conference on Artificial Intelligence. Vancouver, Canada: [s. n.], 1981: 674 - 679.
- [4] Lucas B D. Generalized image matching by the method of differences [D]. Pittsburgh, PA: School of Computer Science, Carnegie-Mellon University, 1984.
- [5] Bruhn A, Weickert J, Schnorr C. Lucas/Kanade Meets Horn/Schunck; Combining Local and Global Optic Flow Methods [J]. International Journal of Computer Vision, 2005, 61(3):211-231.
- [6] Bigun J, Granlund G H. Optical flow based on the inertia matrix in the frequency domain [C]//Proc. SSAB Symposium on Picture Processing. Lund, Sweden: [s. n.], 1988.
- [7] Bigun J, Granlund G H, Wiklund J. Multidimensional orientation estimation with applications to texture analysis and optical flow [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1991, 13(8):775-790.
- [8] Horn B K P, Schunck B G. Determining Optical Flow [J]. Artificial intelligence, 1981, 25(3):185-203.
- [9] Alvarez L, Esclar'in J, Lef'ebure M, et al. A PDE model for computing the optical flow [C]//Proc. XVI Congreso de Ecuaciones Diferenciales y Aplicaciones. Las Palmas de Gran Canaria, Spain: [s. n.], 1999:1349-1356.
- [10] Alvarez L, Weickert J, S'anchez. Reliable estimation of dense optical flow fields with large displacements [J]. International Journal of Computer Vision, 2000, 39(1):41-56.
- [11] Young D M. Iterative Solution of Large Linear Systems [M]. New York: Academic Press, 1971.
- [12] Anandan P. A computational framework and an algorithm for the measurement of visual motion [J]. International Journal of Computer Vision, 1989(2):283-310.

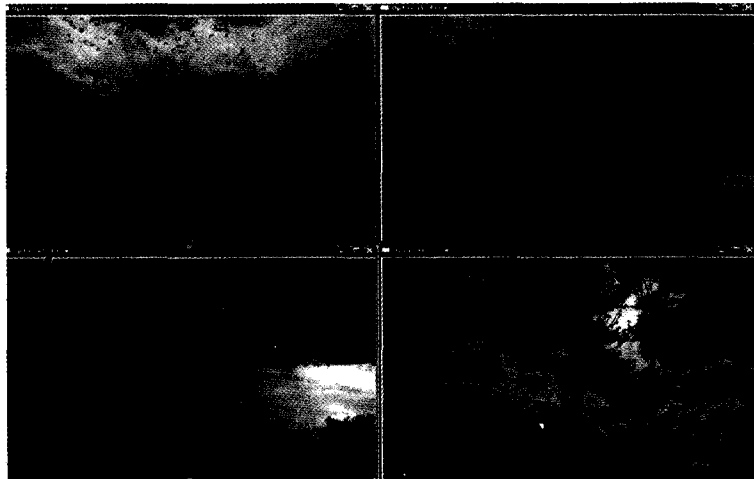


图2 空间光流法在云运动上的应用

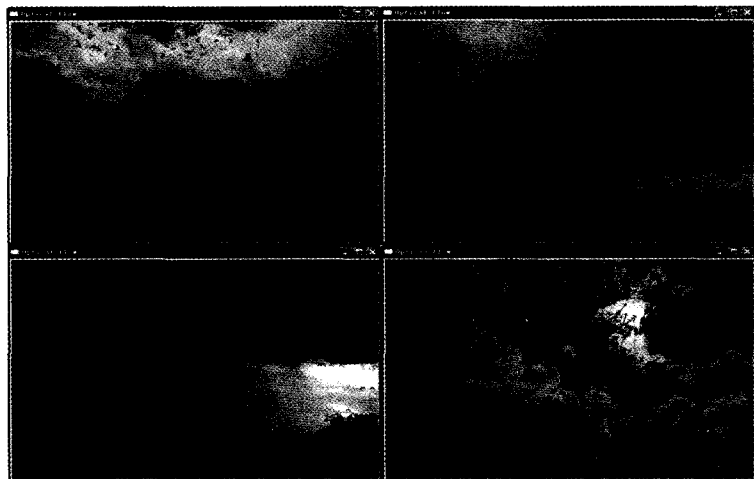


图3 非线性光流法在云运动上的应用

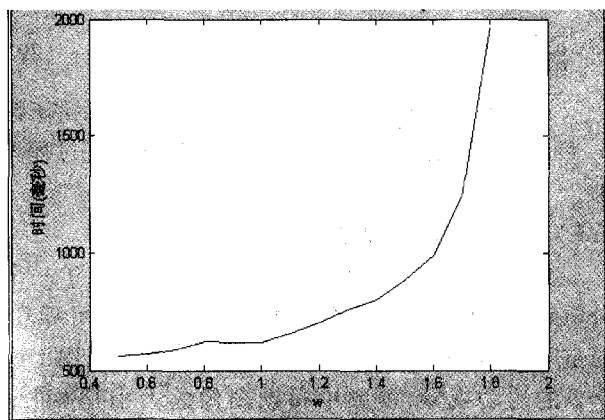


图4  $\omega$  的值收敛速度速度的影响

### 3 结束语

从云的光流中,可以发现云的运动速度和光流变化有一定的关系:云运动速度越快,光流的变化也就越快。因此也就可以通过光流变化,大体上衡量云的运