Optics and Lasers in Engineering 65 (2015) 117–122

Contents lists available at ScienceDirect

Optics and Lasers in Engineering

journal homepage: www.elsevier.com/locate/optlaseng

Speeding up digital image correlation computation using the integral image technique

Lebin Jiang [a], Huimin Xie [a,*], Bing Pan [b]

[a] AML, School of Aerospace, Tsinghua University, Beijing 100084, China
[b] Institute of Solid Mechanics, Beijing University of Aeronautics & Astronautics, Beijing 100191, China

ARTICLE INFO

Article history:
Received 16 February 2014
Received in revised form
26 June 2014
Accepted 1 July 2014
Available online 21 July 2014

Keywords:
Digital image correlation (DIC)
Integral image
Interpolation method

ABSTRACT

In recent years, with the requirement of high-resolution and real-time measurement, the computation speed of digital image correlation (DIC) has become increasingly important. At present, the DIC algorithms based on the iterative spatial domain cross-correlation algorithm are widely recognized as the most robust and rapid. In this paper, the integral image technique is extended to handle the complex items in the equations of the DIC algorithm in order to accelerate the calculation process. The influence of the interpolation method on the performance of the DIC algorithm is also investigated. In addition, the analysis of computational complexity and numerical experiment results are presented to illustrate the effectiveness of this method. The results successfully verify that the proposed method can improve the computation speed of the DIC algorithm greatly, and the improvement is more notable when the fast interpolation method is utilized.

1. Introduction

Digital image correlation (DIC) is a noncontact, full-field optical method for measuring the surface displacement, motion and deformation of solid objects [1]. The principle of the DIC technique is to reconstruct the displacement field by tracking the subsets in the reference image and the deformed image. The available literature shows that the measurement accuracy and the computational speed are the two important concerns in the research of the DIC algorithm.

The measurement accuracy is the first important concern of any measuring method. Early works on the accuracy of DIC primarily focused on the study of the error assessment [2–5]. In recent years, the works on the measurement accuracy were mainly conducted to improve the robustness of the DIC. For example, Zhang et al. [6] used a ring template and quadrilateral element to improve the accuracy of the DIC for the large rotation measurement, and Pan et al. [7] proposed the incremental calculation method for a large deformation measurement with the reliability-guided DIC technique.

The computation speed is the other important concern in the research of the DIC, and this has become more important in recent years due to the increasing demand for real-time processing and high-resolution measurements. Although different types of methods could be utilized for the DIC computation, such as discrete relaxation labeling methods and modern optimization methods including simulated annealing, evolutionary algorithms, genetic algorithms, particle swarm optimization (PSO), and ant colony optimization (ACO) [8], the iterative spatial domain cross-correlation algorithms such as the forward additive Newton–Raphson method [10] are the most popular and effective for the DIC calculation due to their robustness and high speed [9]. These algorithms utilize an iterative process to retrieve the displacement fields, which could optimize the correlation function, and in every iteration, the incremental deformation parameters are obtained to update the parameters of the current estimate. The quality of the initial estimate may affect the convergence property and the convergence speed of the iterative process and thus is critical to the DIC [1,11,12]. Pan [7] proposed the reliability-guided DIC, which could improve the quality of the initial estimate by optimizing the computational order of the subsets, and the result verified that the robustness and the computation speed of the reliability-guided DIC is better than that of the general DIC algorithms. The inverse compositional algorithm, which was proposed by Baker [13], is an effective subset matching algorithm because, in this algorithm, the Hessian matrix remains constant and could be computed only once in each subset matching process. In each iteration, repeated interpolation should be performed. Therefore, the performance of the interpolation algorithms greatly influences the overall performance of the DIC

* Corresponding author.
E-mail address: xiehm@mail.tsinghua.edu.cn (H. Xie).

http://dx.doi.org/10.1016/j.optlaseng.2014.07.001
0143-8166/

algorithm. Pan [14] proposed a method to accelerate the interpolation algorithm by caching the inner parameters.

The integral image, which is also known as a summed area table, is an effective technique to accelerate the computation of the sum of in a rectangular region [15,16]. This technique is extraordinarily useful when the sum of values in a rectangular region is queried frequently. The integral image technique is applied widely in the fields of computer graphic and computer vision such as texture mapping [17], face detection [18] and the creation of scale- and rotation-invariant detectors and descriptors [19]. Additionally, Huang et al. [20] applied this technique to find the initial estimate in the DIC computation.

In this paper, an inverse compositional DIC algorithm using the integral image to speed the calculation of Hessian matrix is presented. The accuracy of this algorithm is the same as that of the existing inverse compositional algorithm because their mathematics equation is equivalent. The time complexity of the algorithm is analyzed, and the performance of the algorithms is verified by the numerical experiments.

2. Principles

2.1. General theory of optical flow and DIC

In the optical flow estimation and the DIC, the gray value and the gradient of the gray value of the corresponding point should be a constraint [21]. These values can be expressed in the form of energy:

$$E_{data}(\boldsymbol{u}) = \int_\Omega \Psi(|I_2(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})) - I_1(\boldsymbol{x})|^2),$$

$$E_{grad}(\boldsymbol{u}) = \int_\Omega \Psi(|\nabla I_2(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})) - \nabla I_1(\boldsymbol{x})|^2), \tag{1}$$

where Ω denotes the image domain, I_1, I_2 denote the first and second frame (the reference image and the target image), $\boldsymbol{x}=(x, y)^T$ denotes the coordinate of the point in the image domain, $\boldsymbol{u}=(u, v)^T$ denotes the optical flow field or the displacement field, Ψ denotes the norm function, which can be $\Psi(x^2)=|x|$, $\Psi(x^2)=x^2$, $\Psi(x^2)=(x^2+\varepsilon^2)^{1/2}$, and so on. The algorithms that only take account of these two energy items may have ambiguous solutions [21]. Therefore, the regularity constraint is often added

$$E_{smooth}(\boldsymbol{u}) = \int_\Omega \Psi(|\nabla \boldsymbol{u}(\boldsymbol{x})|^2 + |\nabla \boldsymbol{v}(\boldsymbol{x})|^2). \tag{2}$$

Therefore, the general form of the energy can be written as follows:

$$E = E_{data} + \alpha E_{grad} + \beta E_{smooth} + \gamma E_{extra} \tag{3}$$

where E_{extra} denotes the extra energy item, which is used for a special purpose.

After establishing the exact form of the energy, the problem of the optical flow estimation can be reduced to the energy minimization problem. The iterative spatial domain cross-correlation algorithm, which is generally used in the DIC, translates the equation of the energy to the corresponding Euler equation by the variational method and then solves the Euler equation with an iterative procedure.

In DIC computations, generally, the energy only takes the item of E_{data} and the motion model adopts the affine model. The original and straightforward numerical scheme of the DIC is the same as that of a classic optical flow algorithm, the Kanade–Lucas–Tomasi (KTL) algorithm, which is a combination of the forward additive match strategy and the steepest descent approximation. Compared with the classic KTL algorithm, the inverse compositional match strategy combined with Gauss–Newton or Levenberg–Marquardt approximation is much more efficient [13].

2.2. The inverse compositional algorithm

When the critical function SSD is used, the energy expression of the inverse compositional DIC in the discrete form is

$$E = \sum_\Omega \left[I_1(W(\boldsymbol{x}; \Delta\boldsymbol{p})) - I_2(W(\boldsymbol{x}; \boldsymbol{p}))\right]^2, \tag{4}$$

where W is the wrap function, p is the deformation parameters and Δp is the incremental deformation parameters. When the affine transformation model is applied, $W(\boldsymbol{x}; \boldsymbol{p})$ can be expressed as

$$W(\boldsymbol{x}; \boldsymbol{p}) = \begin{pmatrix} 1+p_1 & p_3 & p_5 \\ p_2 & 1+p_4 & p_6 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \tag{5}$$

In each iteration, the algorithm computes the value of $\Delta\boldsymbol{p}$ that could minimize the energy E. Then, the update parameters $\boldsymbol{p}$ are

$$W(\boldsymbol{x}; \boldsymbol{p}_{n+1}) = W(\boldsymbol{x}; \boldsymbol{p}_n) W(\boldsymbol{x}; \Delta\boldsymbol{p})^{-1}. \tag{6}$$

When the Gauss–Newton approximation algorithm is used, the value of $\Delta\boldsymbol{p}$ can be determined by the following equation:

$$\Delta\boldsymbol{p} = H^{-1} \sum_\Omega [\nabla I_1 \partial W/\partial\boldsymbol{p}]^T [I_2(W(\boldsymbol{x}; \boldsymbol{p})) - I_1], \tag{7}$$

where H is the Hessian matrix

$$H = \sum_\Omega \left[\nabla I_1 \frac{\partial W}{\partial \boldsymbol{p}}\right]^T \left[\nabla I_1 \frac{\partial W}{\partial \boldsymbol{p}}\right]. \tag{8}$$

For the affine transformation model, the warp function W has the following Jacobian:

$$\frac{\partial W}{\partial \boldsymbol{p}} = \begin{pmatrix} x & 0 & y & 0 & 1 & 0 \\ 0 & x & 0 & y & 0 & 1 \end{pmatrix}, \tag{9}$$

and $\nabla I_1 = [\partial I_1/\partial x, \partial I_1/\partial y]^T = [I_x, I_y]^T$. Thus, the Hessian matrix is

$$H = \sum_\Omega \begin{pmatrix} x^2 I_x^2 & x^2 I_x I_y & xy I_x^2 & xy I_x I_y & x I_x^2 & x I_x I_y \\ x^2 I_x I_y & x^2 I_y^2 & xy I_x I_y & xy I_y^2 & x I_x I_y & x I_y^2 \\ xy I_x^2 & xy I_x I_y & y^2 I_x^2 & y^2 I_x I_y & y^2 I_x^2 & y I_x I_y \\ xy I_x I_y & xy I_y^2 & y^2 I_x I_y & y^2 I_y^2 & y I_x I_y & y I_y^2 \\ x I_x^2 & x I_x I_y & y^2 I_x^2 & y I_x I_y & I_x^2 & I_x I_y \\ x I_x I_y & x I_y^2 & y I_x I_y & y I_y^2 & I_x I_y & I_y^2 \end{pmatrix} \tag{10}$$

2.3. Efficient calculation of Hessian matrix using integral images

The integral image is a common technique to speed up the computation of the sum in a rectangular region. For the 2D analysis, an auxiliary function S can be defined in a recursive form

$$S(x, y; f) = \begin{cases} f(x, y) + S(x-1, y; f) + S(x, y-1; f) - S(x-1, y-1; f), & xy \neq 0 \\ 0 & xy = 0 \end{cases} \tag{11}$$

where f is an arbitrary function and x, y is the integer number, respectively.

Thus, the sum of $f(x, y)$ in a rectangular region and the function S has the following relationship:

$$\sum_{i=x_1}^{x_2} \sum_{j=y_1}^{y_2} f(i, j) = S(x_2, y_2; f) - S(x_1-1, y_2; f) - S(x_2, y_1-1; f) + S(x_1-1, y_1-1; f), \tag{12}$$

The left expression in Eq. (12) computes the sum in the rectangle region $[x_1, x_2] \times [y_1, y_2]$ with the integral image technique, which could be abbreviated as an auxiliary function T for the sake of convenience:

$$T(x_1, y_1, x_2, y_2; f) \triangleq S(x_2, y_2; f) - S(x_1-1, y_2; f) - S(x_2, y_1-1; f) + S(x_1-1, y_1-1; f) \tag{13}$$