SAND2016-7360R

# AN OVERVIEW OF THE GRADIENT-BASED LOCAL DIC FORMULATION FOR MOTION ESTIMATION IN DICE

D. Z. TURNER

Abstract. This document outlines the gradient-based digital image correlation (DIC) formulation used in DICe, the Digital Image Correlation Engine (Sandia's open source DIC code). The gradient-based algorithm implemented in DICe directly reflects the formulation presented here. Every effort is made to point out any simplifications or assumptions involved in the implementation. The focus of this document is on determination of the motion parameters. Computing strain is not discussed herein.

## 1. Formulation

The goal of the gradient-based algorithm is to determine a vector of parameters, $\mathbf{p}$, of a mapping $\psi(\mathbf{x}, \mathbf{p})$, that relates the reference position of a point, $\mathbf{x} = (x, y)$, to the deformed position, $\mathbf{w}$, engendered by the motion. A pictorial description of this mapping is shown in Figure 1. In the local DIC formulation, we enforce this parameterization over a sub-region of the image such that $\mathbf{p}$ is constant over a subset. The parameter vector is composed of the following parameters, $u$, the horizontal displacement, $v$, the vertical displacement, $\theta$, the rotation, $e_x$, the normal extension in the $x$-direction, $e_y$, the normal extension in the $y$-direction, and $\gamma_{xy}$, the shear stretch such that $\mathbf{p} = [u, v, \theta, e_x, e_y, \gamma_{xy}]$.

To simplify the notation, we introduce auxiliary variables $\mathbf{z}(\mathbf{x}, \mathbf{p})$ and $\mathbf{w}(\mathbf{x}, \mathbf{z}) = \mathbf{x} + \mathbf{z}$. The variable $\mathbf{z}(\mathbf{x}, \mathbf{p})$ defines the so-called shape functions of the parameterization

$$\mathbf{z} = \mathbf{R}(\theta) \begin{bmatrix} (1 + e_x)(x - c_x) + \gamma_{xy}(y - c_y) \\ (1 + e_y)(y - c_y) + \gamma_{xy}(x - c_x) \end{bmatrix} + \begin{bmatrix} u \\ v \end{bmatrix} , \tag{1}$$

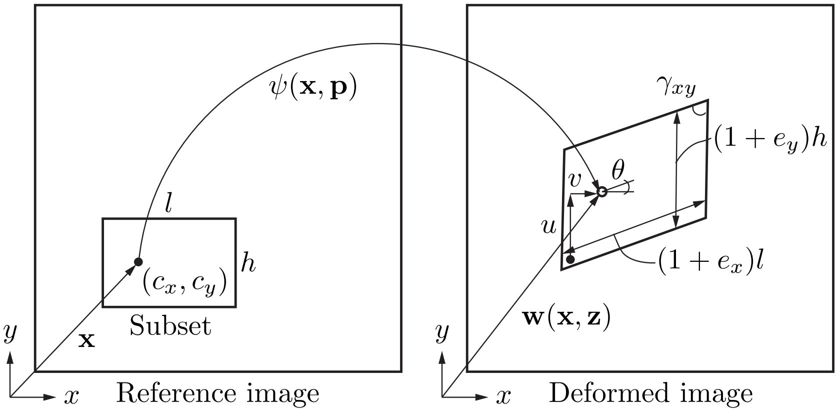

Figure 1. Definition of terms used in the local DIC formulation.

[Date]: July 14, 2016.

where $c_x$ and $c_y$ represent the coordinates of the subset origin (which can be specified arbitrarily) and $\mathbf{R}(\theta)$ is the rotation matrix given as

$$\mathbf{R} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} . \tag{2}$$

When tracking rigid body motion, it is common to neglect the $e_x$, $e_y$, and $\gamma_{xy}$ parameters. The default settings for the tracking algorithm in DICe de-activates these terms. It is also important to point out that $e_x$, $e_y$, and $\gamma_{xy}$ do not accurately represent the engineering strain in the material, but rather an averaged value over the subset. Further analysis is required to compute strains, but this topic is not discussed in this work.

If we denote the normalized, scalar, reference image intensity field as $\phi_0$ and the normalized, deformed image intensity field as $\phi$, the least squares minimization problem can be stated as follows: Find $\mathbf{p}^*$ such that

$$\mathbf{p}^* = \arg\min_{\mathbf{p}} \frac{1}{2} \int_{\Omega} \left(\phi(\mathbf{w}(\mathbf{x},\mathbf{p})) - \phi_0(\mathbf{x})\right)^2 dx \, . \tag{3}$$

where $\Omega$ represent the domain bounded by the subset. Note that $\Omega$ can be of any arbitrary shape in DICe, and need not be square or rectangular. Also note that $\phi(\mathbf{w}) = \phi_0(\mathbf{x})$ satisfies the optical flow constraint

$$\phi_t + \mathbf{b} \cdot \nabla \phi = 0 \, , \tag{4}$$

where $\phi_t$ is the time derivative of the image intensity and $\mathbf{b} \approx \mathbf{z}/t$ is the image velocity. Equation (3) represents an underdetermined least-squares system for which the solution, in a continuous sense, is not unique, nor is the resulting optimality system always invertable. Because of these considerations, in the local DIC formulation, care must be taken to ensure that $\Omega$ is large enough (on the order of three speckles or more) to get a solution to (3). Further details regarding solving this ill-posed problem are provided in [2], suffice to point out that numerical artifacts may arise for a given problem.

We refer to $\phi$ and $\phi_0$ as the normalized image intensities so that we can introduce various correlation criteria. In DICe, the zero-normalized cross-correlation criteria is used such that

$$\phi(\mathbf{x}) = \frac{I(\mathbf{x}) - \bar{I}}{\sqrt{\int_{\Omega} (I(\mathbf{y}) - \bar{I})^2 \, dy}} \qquad \text{and} \qquad \phi_0(\mathbf{x}) = \frac{I_0(\mathbf{x}) - \bar{I}_0}{\sqrt{\int_{\Omega} (I_0(\mathbf{y}) - \bar{I}_0)^2 \, dy}} \, , \tag{5}$$

where $I(\mathbf{x})$ is the interpolated value of the pixel image intensity in the deformed image, $I_0(\mathbf{x})$ is the interpolated reference pixel intensity and $\bar{I}$ and $\bar{I}_0$ are the mean values over the subset $\Omega$.

1.1. **Notation and identities.** Before proceeding, it will be helpful to define the notation and some useful identities. For example, the derivative of $\mathbf{z}$ with respect to position is

$$\frac{\partial \mathbf{z}}{\partial \mathbf{x}} = \mathbf{R}\mathbf{S} \, , \tag{6}$$

where

$$\mathbf{S} = \begin{bmatrix} (1 + e_x) & \gamma_{xy} \\ \gamma_{xy} & (1 + e_y) \end{bmatrix} . \tag{7}$$

The derivatives of $\mathbf{z}$ with respect to each of the parameters are as follows

$$\frac{\partial \mathbf{z}}{\partial u} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} , \tag{8}$$

$$\frac{\partial \mathbf{z}}{\partial v} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} , \tag{9}$$

$$\frac{\partial \mathbf{z}}{\partial \theta} = \begin{bmatrix} -\sin(\theta) & -\cos(\theta) \\ \cos(\theta) & -\sin(\theta) \end{bmatrix} \begin{bmatrix} (1+e_x)(x-c_x) + \gamma_{xy}(y-c_y) \\ (1+e_y)(y-c_y) + \gamma_{xy}(x-c_x) \end{bmatrix} , \tag{10}$$

$$\frac{\partial \mathbf{z}}{\partial e_x} = \mathbf{R} \begin{bmatrix} x-c_x \\ 0 \end{bmatrix} , \tag{11}$$

$$\frac{\partial \mathbf{z}}{\partial e_y} = \mathbf{R} \begin{bmatrix} 0 \\ y-c_y \end{bmatrix} , \tag{12}$$

$$\frac{\partial \mathbf{z}}{\partial \gamma_{xy}} = \mathbf{R} \begin{bmatrix} y-c_y \\ x-c_x \end{bmatrix} , \tag{13}$$

Another useful expression is that the derivative of the deformed position with respect to the auxiliary variable is the identity tensor

$$\frac{\partial \mathbf{w}}{\partial \mathbf{z}} = \mathbf{I} , \tag{14}$$

where $\mathbf{I}$ is the identity tensor. As part of taking the directional derivative of the objective function (3), to establish the optimality system, we will require the derivative of the image intensity with respect to the parameters, for instance,

$$\frac{\partial \phi(\mathbf{w})}{\partial u} = \frac{\partial \phi(\mathbf{w})}{\partial \mathbf{w}} \cdot \frac{\partial \mathbf{w}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial u} . \tag{15}$$

We can re-write the expression above using the following substitution

$$\frac{\partial \phi(\mathbf{w})}{\partial \mathbf{w}} = (\mathbf{RS})^{-1} \frac{\partial \phi(\mathbf{w})}{\partial \mathbf{x}} = \mathbf{G} , \tag{16}$$

which was obtained from the relation

$$\frac{\partial \phi(\mathbf{w})}{\partial \mathbf{x}} = \frac{\partial \mathbf{w}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{x}} \frac{\partial \phi(\mathbf{w})}{\partial \mathbf{w}} . \tag{17}$$

We evaluate the term $(\mathbf{RS})^{-1}$ as

$$(\mathbf{RS})^{-1} = \mathbf{S}^{-1}\mathbf{R}^{-1} = \frac{1}{(1+e_x)(1+e_y) - \gamma_{xy}^2} \begin{bmatrix} (1+e_y) & -\gamma_{xy} \\ -\gamma_{xy} & (1+e_x) \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} . \tag{18}$$

The term $\partial \phi(\mathbf{w}) / \partial \mathbf{x}$ represents the image gradients of the *deformed* image. If the rotation $\theta$ and the stretches $e_x$, $e_y$, and $\gamma_{xy}$ are small, the deformed image gradients will be nearly identical to the reference image gradients. In the case of large deformation the deformed image gradients must be used. There is an input parameter in DICe that enables the user to select the reference or deformed image gradients. Using the reference gradients whenever the rotation and stretches are small saves computation time because only the gradients need only to be computed once, rather than for each frame. In (16) $\mathbf{G}$ represents the transformed image gradients, taking into account the change of variables. Again, if the rotations and stretches are small $\mathbf{G} \approx \partial \phi(\mathbf{w}) / \partial \mathbf{x}$. This approximation is used in the implementation in DICe. After substitution we write

$$\frac{\partial \phi(\mathbf{w})}{\partial u} = \mathbf{G} \cdot \frac{\partial \mathbf{z}}{\partial u} . \tag{19}$$

Similarly,

$$\frac{\partial \phi(\mathbf{w})}{\partial v} = \mathbf{G} \cdot \frac{\partial \mathbf{z}}{\partial v} , \tag{20}$$

$$\frac{\partial \phi(\mathbf{w})}{\partial \theta} = \mathbf{G} \cdot \frac{\partial \mathbf{z}}{\partial \theta} , \tag{21}$$

$$\frac{\partial \phi(\mathbf{w})}{\partial e_x} = \mathbf{G} \cdot \frac{\partial \mathbf{z}}{\partial e_x}\,, \tag{22}$$

$$\frac{\partial \phi(\mathbf{w})}{\partial e_y} = \mathbf{G} \cdot \frac{\partial \mathbf{z}}{\partial e_y}\,, \tag{23}$$

and

$$\frac{\partial \phi(\mathbf{w})}{\partial \gamma_{xy}} = \mathbf{G} \cdot \frac{\partial \mathbf{z}}{\partial \gamma_{xy}}\,. \tag{24}$$

1.2. **Root finding.** The roots of the objective function (3) are determined by taking the derivatives of the function with respect to the parameters leading to a scalar equation for each parameter, $p \in \mathbf{p}$. This set of equations is then solved using a Newton-Raphson approach. The scalar equation for each parameter is written

$$\int_\Omega \big(\phi(\mathbf{w}) - \phi_0(\mathbf{x})\big) \frac{\partial \phi(\mathbf{w})}{\partial p}\, dx = 0\,, \tag{25}$$

where $p$ is replaced by $u$, $v$, $\theta$, $e_x$, $e_y$ or $\gamma_{xy}$. These are the six residual equations, $R_u$, $R_v$, $R_\theta$, $R_{e_x}$, $R_{e_y}$, and $R_{\gamma_{xy}}$. Given a guess for the solution (within the zone of convergence), $\mathbf{p}_n$, the iterative update equation for $\mathbf{p}_{n+1}$ is $\mathbf{p}_{n+1} = \mathbf{p}_n + \delta\mathbf{p}$ where $\delta\mathbf{p}$ is obtained from solving the linear system

$$\mathbf{K}\delta\mathbf{p} = -\mathbf{r}\,, \tag{26}$$

where $\mathbf{K}$ is the tangent matrix, and $\mathbf{r}$ is the vector of residuals $R_u$ through $R_{\gamma_{xy}}$. The elements of the tangent matrix are

$$K_{pm} = \int_\Omega \frac{\partial R_p}{\partial m}\, dx\,, \tag{27}$$

where $p$ and $m = u$, $v$, $\theta$, $e_x$, $e_y$, or $\gamma_{xy}$.

1.2.1. *Simplification of the tangent terms.* In computing the elements of $K_{pm}$ we neglect all second order and higher terms, i.e. $\partial^2\phi(\mathbf{w})/\partial p^2 \approx 0$, leading to the following tangent matrix

$$\int_\Omega \begin{bmatrix}
\frac{\partial \phi(\mathbf{w})}{\partial u}\frac{\partial \phi(\mathbf{w})}{\partial u} & \frac{\partial \phi(\mathbf{w})}{\partial u}\frac{\partial \phi(\mathbf{w})}{\partial v} & \frac{\partial \phi(\mathbf{w})}{\partial u}\frac{\partial \phi(\mathbf{w})}{\partial \theta} & \frac{\partial \phi(\mathbf{w})}{\partial u}\frac{\partial \phi(\mathbf{w})}{\partial e_x} & \frac{\partial \phi(\mathbf{w})}{\partial u}\frac{\partial \phi(\mathbf{w})}{\partial e_y} & \frac{\partial \phi(\mathbf{w})}{\partial u}\frac{\partial \phi(\mathbf{w})}{\partial \gamma_{xy}} \\
\frac{\partial \phi(\mathbf{w})}{\partial v}\frac{\partial \phi(\mathbf{w})}{\partial u} & \frac{\partial \phi(\mathbf{w})}{\partial v}\frac{\partial \phi(\mathbf{w})}{\partial v} & \frac{\partial \phi(\mathbf{w})}{\partial v}\frac{\partial \phi(\mathbf{w})}{\partial \theta} & \frac{\partial \phi(\mathbf{w})}{\partial v}\frac{\partial \phi(\mathbf{w})}{\partial e_x} & \frac{\partial \phi(\mathbf{w})}{\partial v}\frac{\partial \phi(\mathbf{w})}{\partial e_y} & \frac{\partial \phi(\mathbf{w})}{\partial v}\frac{\partial \phi(\mathbf{w})}{\partial \gamma_{xy}} \\
\frac{\partial \phi(\mathbf{w})}{\partial \theta}\frac{\partial \phi(\mathbf{w})}{\partial u} & \frac{\partial \phi(\mathbf{w})}{\partial \theta}\frac{\partial \phi(\mathbf{w})}{\partial v} & \frac{\partial \phi(\mathbf{w})}{\partial \theta}\frac{\partial \phi(\mathbf{w})}{\partial \theta} & \frac{\partial \phi(\mathbf{w})}{\partial \theta}\frac{\partial \phi(\mathbf{w})}{\partial e_x} & \frac{\partial \phi(\mathbf{w})}{\partial \theta}\frac{\partial \phi(\mathbf{w})}{\partial e_y} & \frac{\partial \phi(\mathbf{w})}{\partial \theta}\frac{\partial \phi(\mathbf{w})}{\partial \gamma_{xy}} \\
\frac{\partial \phi(\mathbf{w})}{\partial e_x}\frac{\partial \phi(\mathbf{w})}{\partial u} & \frac{\partial \phi(\mathbf{w})}{\partial e_x}\frac{\partial \phi(\mathbf{w})}{\partial v} & \frac{\partial \phi(\mathbf{w})}{\partial e_x}\frac{\partial \phi(\mathbf{w})}{\partial \theta} & \frac{\partial \phi(\mathbf{w})}{\partial e_x}\frac{\partial \phi(\mathbf{w})}{\partial e_x} & \frac{\partial \phi(\mathbf{w})}{\partial e_x}\frac{\partial \phi(\mathbf{w})}{\partial e_y} & \frac{\partial \phi(\mathbf{w})}{\partial e_x}\frac{\partial \phi(\mathbf{w})}{\partial \gamma_{xy}} \\
\frac{\partial \phi(\mathbf{w})}{\partial e_y}\frac{\partial \phi(\mathbf{w})}{\partial u} & \frac{\partial \phi(\mathbf{w})}{\partial e_y}\frac{\partial \phi(\mathbf{w})}{\partial v} & \frac{\partial \phi(\mathbf{w})}{\partial e_y}\frac{\partial \phi(\mathbf{w})}{\partial \theta} & \frac{\partial \phi(\mathbf{w})}{\partial e_y}\frac{\partial \phi(\mathbf{w})}{\partial e_x} & \frac{\partial \phi(\mathbf{w})}{\partial e_y}\frac{\partial \phi(\mathbf{w})}{\partial e_y} & \frac{\partial \phi(\mathbf{w})}{\partial e_y}\frac{\partial \phi(\mathbf{w})}{\partial \gamma_{xy}} \\
\frac{\partial \phi(\mathbf{w})}{\partial \gamma_{xy}}\frac{\partial \phi(\mathbf{w})}{\partial u} & \frac{\partial \phi(\mathbf{w})}{\partial \gamma_{xy}}\frac{\partial \phi(\mathbf{w})}{\partial v} & \frac{\partial \phi(\mathbf{w})}{\partial \gamma_{xy}}\frac{\partial \phi(\mathbf{w})}{\partial \theta} & \frac{\partial \phi(\mathbf{w})}{\partial \gamma_{xy}}\frac{\partial \phi(\mathbf{w})}{\partial e_x} & \frac{\partial \phi(\mathbf{w})}{\partial \gamma_{xy}}\frac{\partial \phi(\mathbf{w})}{\partial e_y} & \frac{\partial \phi(\mathbf{w})}{\partial \gamma_{xy}}\frac{\partial \phi(\mathbf{w})}{\partial \gamma_{xy}}
\end{bmatrix} dx\,. \tag{28}$$

1.2.2. *Initialization procedure.* For each frame, the initial guess for the solution $\mathbf{p}$ is set to the solution from the previous step. This initialization method places stringent requirements on the frame rate to capture high speed motion. In practice, the motion should be less than two or three pixels per frame for this initialization method to work. Other methods for determining the initial guess are available in DICe, but using the previous frame's solution is the default initialization method. This method is also fastest computationally.

1.2.3. *Jump tolerances.* To ensure that the root finding does not converge on another local minimum, jump tolerances are used such that if the converged solution is more than the prescribed number of pixels from the previous solution or if the angle changes by more than the set tolerance tracking is aborted. The details of how to enforce the jump tolerances are given in the DICe reference manual [3].

## 2. Other algorithmic details

2.1. **Filtering.** Filtering of the reference and deformed images is performed using a 5, 7, 9, 11, or 13 point Gauss convolution filter. If the filter coefficients for each point are denoted $\mathbf{c}_g$, the mask matrix is computed as $\mathbf{c}_g^T \mathbf{c}_g$. The mask coefficients, $\mathbf{c}_g$ are given in the tables below.

Table 1. 5 point Gauss filter coefficients

| | | | | |
|---|---|---|---|---|
| 0.0014 | 0.1574 | 0.62825 | 0.1574 | 0.0014 |

Table 2. 7 point Gauss filter coefficients

| | | | | | | |
|---|---|---|---|---|---|---|
| 0.0060 | 0.0606 | 0.2418 | 0.3831 | 0.2418 | 0.0606 | 0.0060 |

Table 3. 9 point Gauss filter coefficients

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0.0007 | 0.0108 | 0.0748 | 0.2384 | 0.3505 | 0.2384 | 0.0748 | 0.0108 | 0.0007 |

Table 4. 11 point Gauss filter coefficients

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.0001 | 0.0017 | 0.0168 | 0.0870 | 0.2328 | 0.3231 | 0.2328 | 0.0870 | 0.0168 | 0.0017 | 0.0001 |

2.2. **Quadrature.** Single point quadrature is used to evaluate the integrals in the optimality system where the integration points correspond to the pixel centroids in the reference image. Using this quadrature scheme (which is not exact given the irregularity of $\phi$) leads to the following expression for the residual equations

$$\sum_{i=1}^{n} \big(\phi(\mathbf{w}_i) - \phi_0(\mathbf{x}_i)\big) \frac{\partial \phi(\mathbf{w}_i)}{\partial p} = 0\,, \tag{29}$$

where $n$ is the number of pixels in the subset. The tangent terms can also be re-written in terms of summations over pixels rather than integrals.

2.3. **Computing image gradients.** Image gradients in $x$ are computed using a row filter with coefficients 1/12[1,-8,0,8,-1]. The image gradients in $y$ are computed using a column filter with the same coefficients. Additional smoothing of the computed gradients is available via setting an input parameter, but the default setting is non-smoothed gradients.

When image gradients are computed in the deformed image, the gradient values must be interpolated given the deformed position of the pixel. The interpolation method used for image intensity interpolation is also used for gradient interpolation. If only the reference image gradients are computed, interpolation is not necessary.

2.4. **Interpolation.** The default interpolant in DICe is the Keys fourth order interpolant. The details of ththis interpolant can be found in [1]. Bilinear and bicubic polynomial interpolants are also available as settings in the input parameters.

## Acknowledgment

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Table 5. 13 point Gauss filter coefficients

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0.0001 | 0.0012 | 0.0085 | 0.0380 | 0.1109 | 0.2108 | 0.2611 | 0.2108 |
| 0.1109 | 0.0380 | 0.0085 | 0.0012 | 0.0001 | | | |

## References

[1] R. Keys. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 29:1153–1160, 1981.

[2] R. B. Lehoucq, D. Z. Turner, and C. A. Garavito-Garzón. PDE-constrained optimization for digital image correlation. *SAND-2015-8515*, 2015.

[3] D. Z. Turner. Digital Image Correlation Engine (DICe) Reference Manual. *Sandia Report*, SAND2015-10606 O, URL: http://dicengine.github.io/dice/, 2015.

Correspondence to: Daniel Z. Turner, Center for Computing Research, Sandia National Laboratories, P.O. Box 5800; Albuquerque, New Mexico 87185. *E-mail address:* dzturne@sandia.gov, *Phone:* (505) 845-7446