

# Regularizing Optical-Flow Computation using Tensor Theory and Complex Analysis

Dan Koppel\*, Chang-Ming Tsai, and Yuan-Fang Wang  
Department of Computer Science  
University of California, Santa Barbara

## Abstract

*This paper reports a technique that improves the robustness and accuracy in computing dense optical-flow fields. We propose a global formulation with a regularization term. The regularization expressions are derived based on tensor theory and complex analysis. It is shown that while many regularizers have been proposed (image-driven, flow-driven, homogeneous, inhomogeneous, isotropic, anisotropic), they are all variations of a single base expression  $\nabla u \nabla u^T + \nabla v \nabla v^T$ . These regularizers, strictly speaking, are valid for uniform 2D translational motion only, because what they do essentially is to penalize changes in a flow field. However, many flow patterns—such as rotation, zoom, and their combinations, induced by a 3D rigid-body motion—are **not** constant. The traditional regularizers then incorrectly penalize these legal flow patterns and result in biased estimates. The purpose of this work is then to derive a new suite of regularization expressions that treat all valid flow patterns resulting from a 3D rigid-body motion equally, without unfairly penalizing any of them.*

## 1. INTRODUCTION

We address the problem of computing the optical-flow fields induced by rigid-body motions in space. The formulation uses regularization, with the regularization expressions derived based on tensor theory and complex analysis. The proposed framework is applicable to the scenarios where a static scene (or a dynamic scene with independently, but rigidly, moving objects) is viewed by a moving camera or observer, such as in mobile robotic applications. .

## 2. METHODS

We propose to investigate the problem of designing flexible and robust regularizers in a combined local-global framework [1], using tensor theory and complex analysis. As shown in Table 1, while many regularizers have been proposed (image-driven, flow-driven, homogeneous,

inhomogeneous, isotropic, anisotropic), we argue that *they are all variations of a single base expression  $\nabla u \nabla u^T + \nabla v \nabla v^T$* . However,  $\nabla u \nabla u^T + \nabla v \nabla v^T$  and its variations are *valid for uniform 2D translational motion only*, because what these regularizers do essentially is to penalize *changes* in the flow field. Some [5] will do it blindly, while others will do it only at certain locations ([10]) or along certain directions ([9]). Still others will use a robust norm to minimize the influence of the outliers ([2, 8, 7]).

Should the legal flow fields be restricted to uniform translational only? One can easily envision many other flow patterns that are observed as a result of a 3D rigid-body motion. E.g., if the camera is advancing toward (or away from) some object, the observed flow field shows a zoom (in or out) pattern. If the camera is observing, say, a revolution in a 2D plane head-on, the flow field will be a 2D rotation. *In both cases (and many others), the flow fields are **not** constant. The regularizers in Table 1 then incorrectly penalize these legal patterns and may result in biased estimates.*

A more rigorous proof of this phenomenon is given below: Consider an object undergoing a rigid-body motion in space. Let  $t$  and  $t'$  denote two time instants when the object is imaged. The movement of a point on the object  $\mathbf{P} = [X, Y, Z]^T$  at  $t$  to  $\mathbf{P}' = [X', Y', Z']^T$  at  $t'$  is given by (assuming that  $\delta t = t' - t = 1$ )

$$\mathbf{P}' = \mathbf{P} + \boldsymbol{\Omega} \times (\mathbf{P} - \mathbf{P}_o) + \mathbf{T} = \mathbf{P} + \boldsymbol{\Omega} \times \mathbf{P} - \boldsymbol{\Omega} \times \mathbf{P}_o + \mathbf{T}, \quad (1)$$

where  $\boldsymbol{\Omega} = [\Omega_X, \Omega_Y, \Omega_Z]^T$  represents the instantaneous angular velocity, and  $\mathbf{T} = [T_x, T_y, T_z]^T$  the instantaneous linear velocity of the object, and  $\mathbf{P}_o$  is any point on the rotation axis. As  $\boldsymbol{\Omega} \times \mathbf{P}_o$  is a constant, we absorb this term into the linear velocity  $\mathbf{T}$ . At a slight abuse of notation, we still use  $\mathbf{T}$  to represent the combined quantity  $\mathbf{T} - \boldsymbol{\Omega} \times \mathbf{P}_o$ .

We assume an idealized pinhole camera model and the perspective projection equations  $x = X/Z$  and  $y = Y/Z$ . Based on Eq. 1, the flow velocity  $\mathbf{u} = [x' - x, y' - y]^T$  can be shown [6] to decompose into six components in Table 2, induced by the six valid degrees-of-freedom (DOFs)—three translational and three rotational—of a rigid 3D motion. Using those definitions, we have

$$\mathbf{u} = \mathbf{u}_{T_X} + \mathbf{u}_{T_Y} + \mathbf{u}_{T_Z} + \mathbf{u}_{\Omega_X} + \mathbf{u}_{\Omega_Y} + \mathbf{u}_{\Omega_Z} \quad (2)$$

\*The author's current address: STI Medical Systems, 733 Bishop Street, Suite 3100, Honolulu, HI 96813.

Table 1. Different incarnations of the base regularization expression  $\nabla u \nabla u^T + \nabla v \nabla v^T$ , where  $g$  is a decreasing, strictly positive function,  $\lambda$  is a weighting factor, and  $h$  is a convex function representing a robust error norm.

Expression	Citation	Property
$tr(\nabla u \nabla u^T + \nabla v \nabla v^T)$	[5]	homogeneous and isotropic
$g(\nabla I)(tr(\nabla u \nabla u^T + \nabla v \nabla v^T))$	[10]	inhomogeneous and isotropic, image-driven
$tr\left(\frac{1}{ \nabla I ^2 + 2\lambda^2} \begin{bmatrix} \nabla u^T \\ \nabla v^T \end{bmatrix} ([\nabla I_\perp \nabla I_\perp^T] + \lambda^2 \mathbf{I}) \begin{bmatrix} \nabla u \\ \nabla v \end{bmatrix}\right)$	[9]	inhomogeneous and anisotropic, image-driven
$h(tr(\nabla u \nabla u^T + \nabla v \nabla v^T))$	[2, 8]	inhomogeneous and isotropic, flow-driven
$tr(h(\nabla u \nabla u^T + \nabla v \nabla v^T))$	[7]	inhomogeneous and anisotropic, flow-driven

Table 2. The six flow components that are induced by the six DOFs of a 3D rigid-body motion.

DOF	Constraint	$\mathbf{u} = [u, v]^T$
$X$ trans	$T_{\{Y,Z\}} = \Omega_{\{X,Y,Z\}} = 0$	$\mathbf{u}_{T_X} = [T_X/Z, 0]$
$Y$ trans	$T_{\{X,Z\}} = \Omega_{\{X,Y,Z\}} = 0$	$\mathbf{u}_{T_Y} = [0, T_Y/Z]$
$Z$ trans	$T_{\{X,Y\}} = \Omega_{\{X,Y,Z\}} = 0$	$\mathbf{u}_{T_Z} = -T_Z[\frac{x}{Z}, \frac{y}{Z}]$
$X$ rot	$T_{\{X,Y,Z\}} = \Omega_{\{Y,Z\}} = 0$	$\mathbf{u}_{\Omega_X} = -\Omega_X[x, 1 + y^2]$
$Y$ rot	$T_{\{X,Y,Z\}} = \Omega_{\{X,Z\}} = 0$	$\mathbf{u}_{\Omega_Y} = \Omega_Y[1 + x^2, xy]$
$Z$ rot	$T_{\{X,Y,Z\}} = \Omega_{\{X,Y\}} = 0$	$\mathbf{u}_{\Omega_Z} = \Omega_Z[-y, x]$

What is significant about the decomposition in Eq. 2 is that, of the six valid image flow patterns resulting from a 3D rigid-body motion, only two of them,  $\mathbf{u}_{T_X}$  and  $\mathbf{u}_{T_Y}$ , can be considered *roughly* constant and translational—assuming a *small image patch and a constant object depth  $Z$* . Other valid flow patterns include rotation ( $\mathbf{u}_{\Omega_Z}$ ), zoom ( $\mathbf{u}_{T_Z}$ ), and more complicated patterns ( $\mathbf{u}_{\Omega_X}$  and  $\mathbf{u}_{\Omega_Y}$  that represent non-uniform translational fields), which are **not** constant. The regularizers in Table 1 thus incorrectly penalize these four valid flow patterns and their combinations.

Our goal is then to derive a new set of regularization expressions that treat all six valid flow patterns and their combinations equally, without unfairly penalizing any of them.<sup>1</sup> Furthermore, we will show that our regularization expressions have the following novel features:

- We demonstrate that the design of the new regularizers is firmly rooted in tensor theory and complex analysis. The validity can be proven mathematically.
- The new regularization expressions contain only the quadratic terms of the unknown flow variables and their derivatives. Because the variational derivatives of a quadratic expression result in equations that are linear in the unknowns [11, 3], this enables the resulting equations to be solved efficiently using linear algebra methods. Hence, *the new regularizers are not any more expensive computationally but provide more accurate results*.
- The regularization terms work readily with the combined local-global technique [1] and can be used in both the image-driven and the flow-driven formulations.

Our analysis is inspired by the stress-strain tensor analysis in 2D continuum mechanics [4]. Continuum mechanics

<sup>1</sup>We demonstrate the design and use of the component regularizers in this paper. How to use them together synergistically will be reported in another forthcoming paper.

deal with the force (stress) applied to a solid and the resulting deformation (strain). We can draw an intuitive analogy as follows: Consider the perspective projection of a 3D object in two images at time  $t$  and  $t'$ . The corresponding image points  $\mathbf{p} = [x, y]^T$  and  $\mathbf{p}' = [x', y']^T$  are related by  $\mathbf{p}' = \mathbf{p} + \mathbf{u}$  (assuming, again,  $t' - t = 1$ ). *If we treat  $\mathbf{p}$  and  $\mathbf{p}'$  as the configurations of the object's projection before and after a 2D force is applied, the flow field  $\mathbf{u}$  then describes essentially the same phenomenon as the displacement field of a solid deformed by the presence of (2D) stress.*

Many useful tensor expressions are available from the stress-strain analysis to characterize the displacement (hence the flow) fields. For example, it is known that for the displacement to represent a rigid-body motion, it is necessary and sufficient that the Lagrange strain tensor vanishes everywhere in the body [4]

$$\mathcal{E} = \frac{1}{2} \begin{bmatrix} 2u_x + u_x u_x + v_x v_x & u_y + v_x + u_x u_y + v_x v_y \\ u_y + v_x + u_x u_y + v_x v_y & 2v_y + u_y u_y + v_y v_y \end{bmatrix} \quad (3)$$

While the Lagrange strain tensor is mathematically exact, it contains product terms that make the expression nonlinear. A simpler expression, called the engineering strain tensor, drops the product terms to obtain

$$\mathbf{E} = \begin{bmatrix} u_x & \frac{1}{2}(u_y + v_x) \\ \frac{1}{2}(u_y + v_x) & v_y \end{bmatrix}. \quad (4)$$

The engineering strain tensor is an approximate expression and is valid only for small deformations.<sup>2</sup> Nonetheless, we base our regularizers on the engineering strain to keep the regularization expressions simple. Furthermore, expressions in Table 2 are approximations, closer in spirit to Eq. 6 rather than Eq. 5. Hence, using the full machinery of the Lagrange strain is not justified unless a correspondingly exact

<sup>2</sup> The difference between Eq. 3 and 4 can best be understood by an example. A pure rotation (a rigid-body motion) should induce no strain (deformation). An exact representation of a 2D rotation is

$$\mathbf{p}' = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \mathbf{p}, \quad \mathbf{u} = \mathbf{p}' - \mathbf{p} = \begin{bmatrix} \cos\theta - 1 & -\sin\theta \\ \sin\theta & \cos\theta - 1 \end{bmatrix} \mathbf{p} \quad (5)$$

which, when the rotation angle is small, can be approximated as

$$\mathbf{p}' \approx \begin{bmatrix} 1 & -\theta \\ \theta & 1 \end{bmatrix} \mathbf{p}, \quad \mathbf{u} = \mathbf{p}' - \mathbf{p} \approx \begin{bmatrix} 0 & -\theta \\ \theta & 0 \end{bmatrix} \mathbf{p}. \quad (6)$$

It is then a simple exercise to show that  $\mathbf{u}$  in Eq. 5 satisfies the Lagrange strain in Eq. 3 exactly but not the engineering strain in Eq. 4, while  $\mathbf{u}$  in Eq. 6 satisfies the engineering strain in Eq. 4 exactly but not the Lagrange strain in Eq. 3.

formulation is made in all other expressions. Finally, one recalls that a regularization expression should be *generally* valid, but not necessarily *always* and *exactly* true. Hence, a complicated expression is probably not justified, especially if it incurs significant computational costs.

If the engineering strain tensor expression in Eq. 4 is used as the regularizer

$$R \equiv \|\mathbf{E}\|_F^2 \quad (7)$$

where  $\|\mathbf{E}\|_F = \sqrt{\sum_i \sum_j E_{ij}^2}$  is the Frobenius norm of a matrix, one can immediately verify that it helps in properly handling not only the pure translational cases ( $\mathbf{u}_{TX}$  and  $\mathbf{u}_{TY}$ ) but also the pure rotational case ( $\mathbf{u}_{\Omega Z}$ , which the regularizers in Table 1 penalize incorrectly). This is because the traditional regularizer  $tr(\nabla u \nabla u^T + \nabla v \nabla v^T)$  is the same as the square of the Frobenius norm of the matrix  $\nabla \mathbf{u} = [\nabla u \ \nabla v]$ , which is called the deformation gradient tensor [4].  $\nabla \mathbf{u}$  can be decomposed as

$$\nabla \mathbf{u} = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T) + \frac{1}{2}(\nabla \mathbf{u} - \nabla \mathbf{u}^T) \quad (8)$$

into a symmetric part and an anti-symmetric part. It can be easily verified that the symmetric part is identical to the engineering strain tensor  $\mathbf{E}$  in Eq. 4. The anti-symmetric part actually captures the amount of *rotation* executed at this point by the image. If pure rotations (see  $\mathbf{u}_{\Omega Z}$  in Table 2) are not to be penalized by the regularizer, then clearly this term must be suppressed. This leaves the engineering strain, which is the expression we are advocating (due to its zero value for this scenario).

We now turn to the problem of allowing another commonplace flow pattern: uniform rescaling (or “zoom” which can result from an actual zoom operation or a forward camera motion). To accomplish this and still retain the proper handling of image rotations, the regularizer requires some modification. Using the engineering strain tensor as our “building block,” the new expression below allows (i.e., does not incorrectly penalize) flow fields that are composed of any *arbitrary* combination of 2D translation, rotation, and zoom movements, *but it allows no others*. This includes all 2D rigid-body motions (with or without zoom) as well as commonplace real-world motions involving some physical rescaling (such as sequenced satellite images of a hurricane, where all three of our listed modes—translation, rotation, and rescaling—are simultaneously present).

The regularization expression is made up of two terms, which require the following equations to be satisfied:

$$2tr(\mathbf{E}^2) - (tr\mathbf{E})^2 = 0 \quad div\mathbf{E} = 0 \quad (9)$$

where  $div$  is the divergence operator  $[\partial/\partial x, \partial/\partial y]$ . While it may not be immediately apparent why such a regularization expression is used, we prove in the following theorem that the conditions stipulated in Eq. 9 are exactly the ones that are satisfied by the flows made of a combination of some 2D translation, rotation, and zoom motions, *but nothing else*.

In more detail, if the pixel movements between two images are a combination of some 2D translation, rotation, and zoom, we can relate the corresponding pixels as

$$\mathbf{p}' = \rho \mathbf{R} \mathbf{p} + \mathbf{t} \quad (10)$$

where  $\mathbf{R}$  is a 2D rotation matrix,  $\mathbf{t}$  a 2D translation vector, and  $\rho (\neq 0)$  a scalar. Then the flow field  $\mathbf{u}$  is

$$\mathbf{u} = \mathbf{p}' - \mathbf{p} = (\rho \mathbf{R} - \mathbf{I})\mathbf{p} + \mathbf{t} = \rho' \mathbf{R}' \mathbf{p} + \mathbf{t} \quad (11)$$

where  $\mathbf{R}'$  is still a rotation and  $\rho'$  a scalar.<sup>3</sup> We now proceed with the following theorem.

**Theorem:** A flow field  $\mathbf{u} = [u, v]^T$  represents an arbitrary combination of some 2D translation, rotation, and uniform rescaling if and only if  $2tr(\mathbf{E}^2) - (tr\mathbf{E})^2 = 0$  and  $div\mathbf{E} = 0$ , when  $\mathbf{u}$  is used as the displacement field in the engineering tensor expression  $\mathbf{E}$ .

**Proof:** We first prove the “if” part. If  $\mathbf{u}$  is defined by Eq. 11 using a rotation angle of  $\theta'$ , then it is easily verified that the resulting engineering strain tensor is given by

$$\mathbf{E} = \begin{bmatrix} \rho' \cos \theta' & 0 \\ 0 & \rho' \cos \theta' \end{bmatrix} \quad (12)$$

in this case  $2tr(\mathbf{E}^2) - (tr\mathbf{E})^2 = 4\rho'^2 \cos^2 \theta' - 4\rho'^2 \cos^2 \theta' = 0$ , and  $div\mathbf{E} = 0$  since  $\mathbf{E}$  is a constant tensor.

To prove the “only if” part, we note that

$$\begin{aligned} 2tr(\mathbf{E}^2) - (tr\mathbf{E})^2 &= 2(u_x^2 + \frac{1}{2}(u_y + v_x)^2 + v_y^2) - (u_x + v_y)^2 \\ &= u_x^2 - 2u_x v_y + v_y^2 + (u_y + v_x)^2 \\ &= (u_x - v_y)^2 + (u_y + v_x)^2 = 0 \end{aligned} \quad (13)$$

$$\begin{aligned} div\mathbf{E} &= \begin{bmatrix} \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \end{bmatrix} \begin{bmatrix} u_x & \frac{1}{2}(u_y + v_x) \\ \frac{1}{2}(u_y + v_x) & v_y \end{bmatrix} \\ &= \begin{bmatrix} u_{xx} + \frac{1}{2}(u_{yy} + v_{xy}) & \frac{1}{2}(u_{xy} + v_{xx}) + v_{yy} \end{bmatrix} = 0 \end{aligned} \quad (14)$$

If Eq. 13 is true, we have

$$u_x = v_y \quad u_y = -v_x. \quad (15)$$

which leads to

$$\begin{aligned} u_y = -v_x &\implies \partial u_y / \partial x = -\partial v_x / \partial x \implies u_{xy} + v_{xx} = 0 \\ u_y = -v_x &\implies \partial u_y / \partial y = -\partial v_x / \partial y \implies u_{yy} + v_{xy} = 0 \end{aligned} \quad (16)$$

Eq. 14 is then simplified by means of Eq. 16, yielding

$$div\mathbf{E} = [u_{xx} \ v_{yy}] = 0. \quad (17)$$

In Eq. 17,  $u_{xx} = 0$  implies that  $u = f_1(y)x + f_2(y)$ . Similarly  $v_{yy} = 0$  implies that  $v = g_1(x)y + g_2(x)$ . However,  $u_x = v_y$  from Eq. 15 requires  $f_1(y) = g_1(x) = a$ , and  $u_y = -v_x$  from Eq. 15 requires that  $f_2'(y) = -g_2'(x) = b$ , or  $f_2(y) = by + c$  and  $g_2(x) = -bx + d$ . Putting these terms into  $u$  and  $v$ , we obtain  $u = ax + by + c$  and  $v = ay - bx + d$ . Defining  $\rho = \sqrt{a^2 + b^2}$ , we obtain

$$\mathbf{u} = \begin{bmatrix} a & b \\ -b & a \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c \\ d \end{bmatrix} = \rho \begin{bmatrix} \frac{a}{\rho} & \frac{b}{\rho} \\ -\frac{b}{\rho} & \frac{a}{\rho} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c \\ d \end{bmatrix} \quad (18)$$

<sup>3</sup>The proof that a unique orthogonal  $\mathbf{R}'$  satisfying Eq. 11 can always be found is not derived here. It is fairly straightforward and depends mainly on an application of the cosine rule [11].

This resulting matrix is indeed an orthogonal matrix, yielding the desired combination of a translation (by  $[c, d]^T$ ), a rotation (by  $\theta = -\tan^{-1}\frac{b}{a}$ ), and a scaling (by  $\rho$ ).  $\square$

Several observations can be made about the new regularization expressions in Eq. 9. First, Eq. 9 is equivalent to Eqs. 13 and 17, whose enforcement requires an expression no higher than quadratic in  $u$  and  $v$  (and their derivatives). Hence, the governing equations resulting from the variational derivatives will be linear in these variables. Second, the expressions state that the flow fields satisfying  $u_x = v_y$  and  $u_y = -v_x$  should not be penalized. If we think of a flow field as a complex function  $f(z) = u(x, y) + iv(x, y)$ ,  $z = x + iy$ , then the above requirement corresponds to the well known Cauchy-Riemann Equations from the field of complex analysis [14]. Complex functions that satisfy the Cauchy-Riemann Equations are called analytic or holomorphic, and holomorphic functions in the complex domain have convergent Taylor series [14]

$$\begin{aligned} f(z) &= f(z_o) + f'(z_o)(z - z_o) + \frac{f''(z_o)}{2}(z - z_o)^2 \dots, \\ f(z) &= f(z_o) + re^{i\theta}(z - z_o) + \frac{f''(z_o)}{2}(z - z_o)^2 + \dots \end{aligned} \quad (19)$$

In our context, the 0th-order term corresponds to a constant translation ( $u(x, y) = u(x_o, y_o)$  and  $v(x, y) = v(x_o, y_o)$ ), and the 1st-order term corresponds to either a rotation (due to nonzero  $\theta$ ), a zoom (due to non-unity  $r$ ), or some combination of both. Hence, the Cauchy-Riemann equations express the fact that a holomorphic complex function maps a 2D space to a new space, such that a *local neighborhood* of points is acted upon by a translation, a rotation, and an isotropic (or uniform) rescaling. However, this statement is true *locally* only, and as such still admits a large class of 2D flows. To enforce the conditions in a *more widespread manner* (i.e., beyond an infinitesimal neighborhood) requires some additional constraint, which we now address.

We show in the following lemma that  $\text{div} \mathbf{E} = 0$  corresponds to  $f''(z) = 0$ . Using the fact that

$$f^{(n)} = \frac{d^{(n-2)}}{dz^{(n-2)}} f'' \quad n > 2, \quad (20)$$

an identically vanishing second derivative implies that all higher derivatives vanish as well. Hence,  $f$  is constrained to be of the form  $f(z_o) + re^{i\theta}(z - z_o)$ . The condition can thus be seen to be equivalent to a narrowing of the available pool of functions to only the linear ones. This corresponds to flow fields that are some combination of translation, rotation, and zoom components *over an extended region* (i.e., a non-infinitesimal neighborhood). The stated equivalency is now proven in the lemma below.

**Lemma:**  $\text{div} \mathbf{E} = 0$  if and only if  $f''(z) = 0$ , where  $f(z) = u(x, y) + iv(x, y)$ .

**Proof:** By definition, the derivatives of an *analytic* complex function can be evaluated along *any* direction at a point and always yield the same value [14]. In particular, taking the derivatives along the  $x$  and  $iy$  directions gives

$$f'' = u_{xx} + iv_{xx} \quad f'' = -u_{yy} - iv_{yy} \quad (21)$$

The matching real and imaginary components can then be equated and this fact then used to yield:

$$f'' = u_{xx} - iv_{yy} \quad (22)$$

(This can also be verified by the fact that the components of an analytic complex function must be harmonic [14], i.e.,  $v_{xx} + v_{yy} = 0$ .) Assuming here that the Cauchy-Riemann condition is true and also recalling that this condition implies Eq. 16, we can rewrite the above equation as:

$$f'' = u_{xx} + \frac{1}{2}(u_{yy} + v_{xy}) - i(v_{yy} + \frac{1}{2}(u_{xy} + v_{xx})) \quad (23)$$

Comparing Eqs. 14 and 23 shows that they have identical  $u$  components and their  $v$  components differ only by a sign. Thus if one expression is zero, so must the other (and vice-versa). This establishes the equivalence of the strain formulation (Eq. 9) and the complex function formulation.  $\square$

Table 3. Legal flow patterns induced by a 3D rigid-body motion and their regularization expressions.

DOF	Regularization expressions
$X$ translation	$R_1 = u_x^2 + u_y^2 + v_x^2 + v_y^2$
$Y$ translation	$R_2 = u_x^2 + u_y^2 + v_x^2 + v_y^2$
$Z$ translation	$R_3 = (u_x - v_y)^2 + u_y^2 + v_x^2$
$X$ rotation	$R_4 = (2u_x - v_y)^2 + v_x^2 + u_{yy}^2$
$Y$ rotation	$R_5 = (u_x - 2v_y)^2 + u_y^2 + v_{xx}^2$
$Z$ rotation	$R_6 = (u_y + v_x)^2 + u_x^2 + v_y^2$
2D motion (rot, trans, and scale)	$R_{2D} = (u_x - v_y)^2 + (u_y + v_x)^2 + u_{xx}^2 + v_{yy}^2$

In Table 3 we introduce a new suite of regularization expressions that specifically target the perspective-projection-based flow patterns enumerated in Table 2.<sup>4</sup> The readers can apply the new regularizers ( $R_3$  to  $R_6$ ) to the corresponding flow fields  $\mathbf{u}_{T_Z}$ ,  $\mathbf{u}_{\Omega_X}$ ,  $\mathbf{u}_{\Omega_Y}$ , and  $\mathbf{u}_{\Omega_Z}$  in Table 2 to verify that they vanish (do not penalize) for the particular type of motion indicated but are nonzero (penalize) for the other types. Note that all the new expressions admit a pure translational motion ( $\nabla u \approx 0$ ,  $\nabla v \approx 0$ ) as a special case.

One observes that all new regularizers contain no higher than the quadratic (2nd-order) terms of the derivatives of the flow variables. This is significant as we can show that the variational derivatives of the cost expression then contain only linear terms of the flow variables and their derivatives, and hence, the resulting system of equations is linear and can be solved using linear algebra methods.

### 3. EXPERIMENTAL RESULTS

We compare the proposed algorithm against the CLG algorithm [1], which is considered one of the best available for computing the flow field [12]. We use the same

<sup>4</sup>Assume that  $Z$  is roughly constant for a small patch, so both 3D  $X$  and  $Y$  translations result in constant flow fields and  $R_1$  and  $R_2$  are the same. The proof of  $R_4$  is given in the Appendix. The proofs of other regularizers are similar and will not be repeated here.



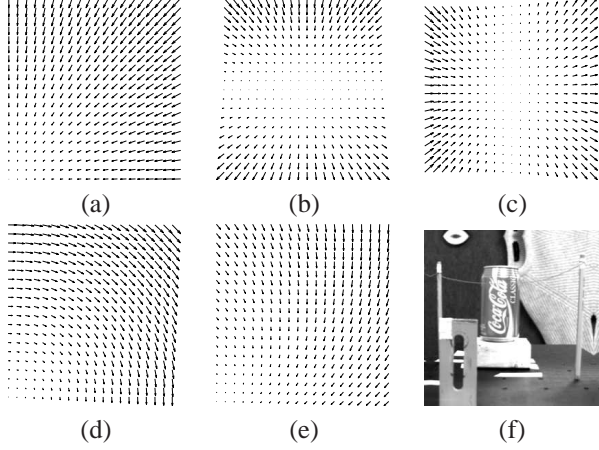


Figure 1. (a) A zoom flow ( $\mathbf{u}_{T_Z}$  case 2), (b) A divergent translational flow ( $\mathbf{u}_{\Omega_X}$  case 4), (c) A divergent translational flow ( $\mathbf{u}_{\Omega_Y}$  case 6), (d) A rotational flow ( $\mathbf{u}_{\Omega_Z}$  case 9), (e) a 2D flow fields (case 11) by combining rotation ( $\mathbf{u}_{\Omega_Z}$ ), translation ( $\mathbf{u}_{T_X}$  and  $\mathbf{u}_{T_Y}$ ), and zoom ( $\mathbf{u}_{T_Z}$ ), and (f) the texture pattern used.

advanced features in the CLG technique, but replace the traditional regularizer with the new ones. More specifically, the iterative successive over-relaxation (SOR) method is used [13] for numerical solution. Furthermore, we adopt the multi-resolution technique to incrementally compute the optical-flow field using a coarse-to-fine strategy [1]. The flow fields are computed using an image pyramid. The flows at the coarser scales are used to position (warp) the image at the next finer scale so that only the motion *increments* need to be computed. The final flow field is the summation of all the flow increments.

**Synthetic Data.** We use twelve spatially-varying flow patterns:  $\mathbf{u}_{T_Z}$  corresponds to a zoom-out pattern with the focus-of-expansion (FOE) at the center (1) or a corner (2) of the image. We also use a zoom-in pattern with the FOE at the corner (3).  $\mathbf{u}_{\Omega_X}$  represents a non-uniform translational flow field with the rotation axis aligned with the center (4) or an edge (5) of the image. Similarly we have two cases, 6 and 7, for  $\mathbf{u}_{\Omega_Y}$ .  $\mathbf{u}_{\Omega_Z}$  represents a rotational field. Again, two cases (8 and 9) are used with the center of rotation at the center and a corner of the image. We also generate a number of 2D flow fields (10, 11, and 12) by combining rotation, translation, and zoom. Sample flow fields (ground truth) are shown in Fig. 1(a)-(e), and the texture pattern used in simulating the flow patterns is shown in Fig. 1(f).

We added different amounts of pixelwise-uncorrelated, additive Gaussian noise to the images (up to 15% of the maximum 255 gray levels per pixel) and charted the performance of the twelve flows under three noise levels in two different ways: (1) Average angular error (AAE) in Fig. 2, where the angular error is defined as  $\cos^{-1}((\hat{\mathbf{u}}_g \cdot \hat{\mathbf{u}}_c)/(|\hat{\mathbf{u}}_g||\hat{\mathbf{u}}_c|))$ ,  $\hat{\mathbf{u}} = [u, v, 1]^T$ , and subscripts  $g$  and  $c$  denote the ground truth and the computed flow fields, and (2) Average residual error (ARE) in Fig. 3,

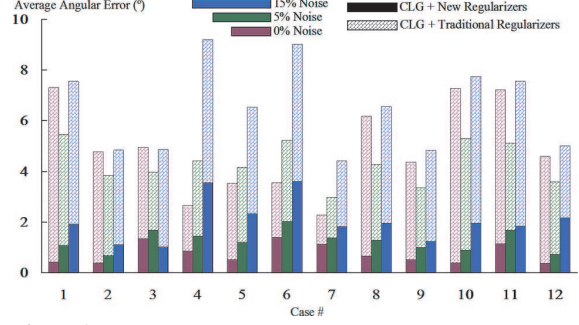


Figure 2. Average angular error (in degrees) for different noise levels.

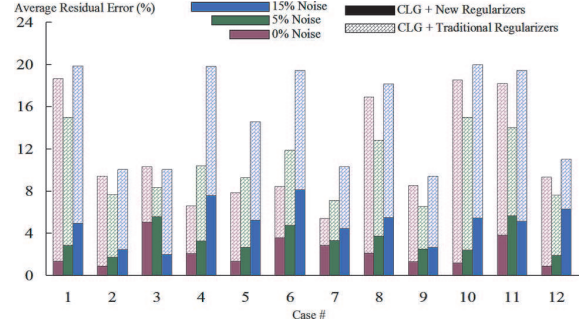


Figure 3. Average residual error (in percentage) for different noise levels.

where the residual error is defined as  $|\mathbf{u}_g - \mathbf{u}_c|/|\mathbf{u}_g|$ .

As can be seen in Fig. 2 and 3, the proposed regularizers outperform the traditional ones for all test patterns and noise levels. Table 4 summarizes the average AAE and ARE for all twelve cases and shows that the proposed method achieves a three- to six-fold reduction in error.

Table 4. Average AAE and ARE for all test cases.

noise	Figs. 2 & 3					
	0%		5%		15%	
	Trad	New	Trad	New	Trad	New
AAE	4.89	0.76	4.31	1.25	6.51	2.04
ARE	11.51	2.19	10.46	3.36	15.17	4.98
Pics	Fig. 4					
	Yosemite		Office		Street	
	Trad	New	Trad	New	Trad	New
AAE	2.07	1.88	5.49	3.78	2.19	1.94
ARE	4.91	4.23	12.08	8.64	5.15	4.44

**Standard Test Sequences.** The results of applying our 2D regularizer to some standard test sequences are shown in Fig. 4. From top to bottom: the Yosemite sequence comprises both “zoom” (mountain) and translation (sky) motions. In the Office sequence, the camera performed a  $Z$  translation (a zoom). In the Street sequence, the camera was panning right while the car was also moving right, but at a faster speed. In the taxi sequence, the two black cars were moving to the right while the white taxi was turning. Fig. 4 shows that we correctly recover all these motion patterns. Table 4 show that the new regularizer achieved a significant reduction in both AAE and ARE over the traditional one.



Figure 4. Left: a sample frame in the video sequence. Right: the computed optical-flow field. From top to bottom: Yosemite, Office, Street, and Taxi sequences.

(The Taxi sequence does not have ground truth, and hence, the result was presented in a graphical form only.) Again, as shown in Table 4, We achieved a reduction in AAE of at least 10% (Yosemite) to as high as 40% (Office).

## APPENDIX

**Lemma:**  $\mathbf{u} = -\Omega_x[xy, 1 + y^2]^T + \mathbf{u}_o = \mathbf{u}_{\Omega_x} + \mathbf{u}_o$ , where  $\mathbf{u}_o$  is an arbitrary constant flow field if and only if  $(2u_x - v_y)^2 + v_x^2 + u_{yy}^2 = 0$ .

**Proof:** The *if* part is trivially verified. The *only if* part is proven as follows. If  $v_x = 0$ , we must have  $v = g(y)$ . If  $u_{yy} = 0$ , we must have  $u = f_1(x)y + f_o(x)$ . Now,  $2u_x - v_y = 0$  implies that  $2f_1'(x)y + 2f_o'(x) = g'(y)$ , which means that  $f_1(x) = cx + d$ ,  $f_o(x) = ax + b$ ,  $u = (cx + d)y + (ax + b)$ ,  $g'(y) = 2cy + 2a$ , and  $g(y) = cy^2 + 2ay + e$ .

Rewrite  $u$  and  $v$  as

$$\begin{aligned} u &= cxy + ax + dy + b \\ &= c(x + d/c)(y + a/c) + b - (ad)/c = cx'y' + u_o \\ v &= c(y^2 + 2(a/c)y) + e \\ &= c(y + a/c)^2 + c + e - a^2/c - c = c(y'^2 + 1) + v_o \end{aligned}$$

We see that  $\mathbf{u}$  can indeed be expressed as  $-\Omega_x[xy, 1 + y^2]^T + \mathbf{u}_o$  in a “canonical” system where  $x' = x + d/c$  and  $y' = y + a/c$ , if we choose  $c = -\Omega_x$ . While the flow expression changes in different coordinate frames (Eq. 24),  $(2u_x - v_y)^2 + v_x^2 + u_{yy}^2$  is always zero regardless of the values of  $a$ ,  $c$ , and  $d$ . Hence, the regularizer can be applied in any coordinate system with or without calibration.  $\square$

## References

- [1] A. Bruhn and J. Weickert. Lucas/Kanade Meets Horn/Schunck: Combining Local and Global Optical Flow Methods. *Int. J. Comput. Vision*, 61:211–231, 2005. 1, 2, 4, 5
- [2] C. Schnörr and R. Sprengel. A Nonlinear Regularization Approach to Early Vision. *Biol. Cybern.*, 72:141–149, 1994. 1, 2
- [3] R. Courant and D. Hilbert. *Methods of Mathematical Physics, Vol. I*. Interscience, London, 1953. 2
- [4] Y. C. Fung and P. Tong. *Classical and Computational Solid Mechanics*. World Science Publishing, 2001. 2, 3
- [5] B. Horn and B. Schunck. Determining Optical Flow. *Artificial Intell. Journal*, 17:185–203, 1981. 1, 2
- [6] M. Irani. Multi-Frame Optical Flow Estimation Using Subspace Constraints. In *Proc. Int. Conf. Comput. Vision*, 1999. 1
- [7] J. Weickert and C. Schnörr. A Theoretical Framework for Convex Regularizers in PDE-Based Computation of Image Motion. *Int. J. Comput. Vision*, 45:245–264, 2001. 1, 2
- [8] M. A. Snyder. On the Mathematical Foundations of Smoothness Constraints for the Determination of Optical Flow and for Surface Reconstruction. *IEEE Trans. Pattern Anal. Machine Intell.*, 13:1105–1114, 1991. 1, 2
- [9] H. H. Nagel and W. Enkelmann. An Investigation of Smoothness Constraints for the Estimation of Displacement Vector Fields from Image Sequences. *IEEE Trans. Pattern Anal. Machine Intell.*, 8:565–593, 1986. 1, 2
- [10] P. Perona and J. Malik. Scale-Space and Edge Detection Using Anisotropic Diffusion. *IEEE Trans. Pattern Anal. Machine Intell.*, 12:629–638, 1990. 1, 2
- [11] L. Rade and B. Westergren. *Mathematics Handbook for Science and Engineering, 5th Ed*. Springer, 2004. 2, 3
- [12] S. Roth and M. J. Black. On the Spatial Statistics of Optical Flow. In *Proc. Int. Conf. Comput. Vision*, volume 1, pages 42–49, 2005. 4
- [13] Y. Saad. *Iterative Methods for Linear Systems, 2nd Ed*. SIAM, 2003. 5
- [14] D. A. Wunsch. *Complex Variables with Applications, 2nd Ed*. Addison Wesley, 1993. 4