Euro. Jnl of Applied Mathematics (2020), *vol.* 31, *pp.* 277–296 © Cambridge University Press 2019.
doi:10.1017/S0956792519000020

Equi-affine differential invariants for invariant feature point detection†

STANLEY L. TUZNIK[1], PETER J. OLVER[2]
and ALLEN TANNENBAUM[1,3]

[1]*Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, NY 11794, USA*
e-mail: stanley.tuznik@stonybrook.edu
[2]*Department of Mathematics, University of Minnesota, Minneapolis, MN 55455, USA*
e-mail: olver@umn.edu
[3]*Department of Computer Science, Stony Brook University, Stony Brook, NY 11794, USA*
e-mail: arobertan@gmail.com

(*Received 20 June 2018*; *revised 05 December 2018*; *accepted 30 January 2019*; *first published online 6 March 2019*)

Image feature points are detected as pixels which locally maximise a detector function, two commonly used examples of which are the (Euclidean) image gradient and the Harris–Stephens corner detector. A major limitation of these feature detectors is that they are only Euclidean-invariant. In this work, we demonstrate the application of a 2D equi-affine-invariant image feature point detector based on differential invariants as derived through the equivariant method of moving frames. The fundamental equi-affine differential invariants for 3D image volumes are also computed.

Key words: Differential invariant, feature detection, equi-affine group, moving fram

2010 Mathematics Subject Classification: 53A15; 53A55; 68U10

1 Introduction

Image feature points have been widely used for decades as fundamental components of computer vision algorithms, and they continue to find use in such applications as medical image registration, estimation of homographies in stereo vision, and stitching of images for panoramas or satellite imagery. Features are generally defined as pixels (or sets of pixels) in an image which can be reliably identified and distinctly described across images. The goal of feature detection and description is to abstract images as point sets which can be matched to each other. These correspondences can then be used to compute (estimate) the transformation between the point sets and, thus, between the images.

Feature detectors are scalar functions which measure some sort of response on an image, and features are typically identified as local extrema in these functions. For example, the popular Harris–Stephens corner detector identifies feature points as pixels, which locally maximise the function

$$R(x, y) = \det(H) - \alpha \operatorname{tr}(H)^2 \tag{1}$$

† This work was supported by AFOSR [grant number FA9550-17-1-0435] and the National Institutes of Health [grant number R01-AG048769].

https://doi.org/10.1017/S0956792519000020 Published online by Cambridge University Press

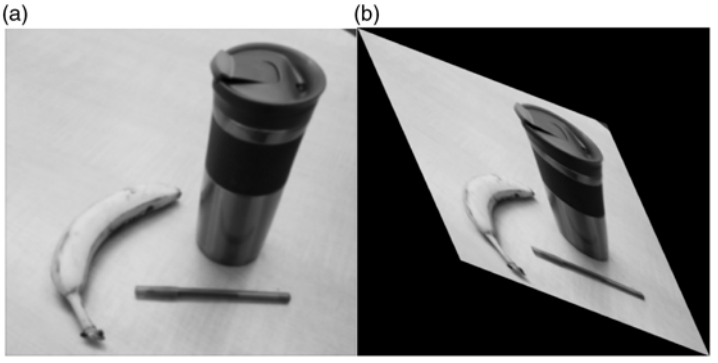


FIGURE 1. Two images of a scene related by an equi-affine transformation.

where α is a parameter and H is the second moment matrix of the image [14]. This function responds strongly on pixels where the image varies greatly in two directions, and thus is a corner detector. The popular SIFT feature algorithm uses a multi-scale image representation and computes a difference-of-Gaussians function (an approximation of the Laplacian-of-Gaussian) as a detector [16]. SIFT features are local maxima of this function over both space and scale. This multi-scale representation allows identification of features at different spatial scales and also serves to smooth noise out of the image.

A major drawback to these detectors is that they are only Euclidean-invariant. In other words, these detectors will identify roughly the same image points as feature points in two images only if the two images are related by a Euclidean transformation (rotation, translation, reflection). The Euclidean group can be generalised to the affine group by including stretching and skewing transformations (see Figure 1). The equi-affine group – i.e. the group of area- or volume-preserving affine transformations – in particular, is useful, i.e. a good approximation to projective transformations that are near the identity. Additionally, these transformations do not suffer from known difficulties associated with projective equivalence [3]. Several affine-invariant image detectors have been proposed in the literature [17,18]. An affine-invariant extension of the SIFT algorithm is proposed in Ref. [33]. This is a brute-force method that detects affine-invariant feature points in two images by simulating many affine transformations of each image and performing the SIFT algorithm between all image pairs. In contrast to affine-invariant detectors, projective invariance and moving frame-based signatures have been successfully applied in vision [12,13].

In this work, we propose the use of an invariant-theory-based approach to equi-affine-invariant feature detection. The equivariant method of moving frames [21,22] was inspired by the classical moving frame theory, most closely associated with Élie Cartan – see [11] – but whose history can be traced back to the early nineteenth century [1]. The equivariant method can be readily employed to compute various invariants of a given transformation (Lie) group, including differential invariants, joint invariants, joint or semi-differential invariants, etc. [20]. These invariants are then combined into an equi-affine-invariant feature detector function [26]. Standard methods for feature description can then be applied to characterise each point. In contrast to the linear smoothing performed during the aforementioned multi-scale feature algorithms, we apply a nonlinear, affine-invariant scale-space to smooth our image.

The paper will have a tutorial flavour so as to remain fairly self-contained. In Section 2, we introduce the fundamental concepts of the method of moving frames and discuss how the method

https://doi.org/10.1017/S0956792519000020 Published online by Cambridge University Press

leads to the computation of differential invariants for a given Lie group acting on a manifold. In Section 3, the method is explicitly applied to the equi-affine group acting on coordinates of functions on $\mathbb{R}^2$ (i.e. images), and the fundamental second-order differential invariants of this action are computed. The differential invariants for the equi-affine group acting on functions in $\mathbb{R}^3$ are also computed. In Section 4, these invariants are used to compute a feature detector which is demonstrated to perform well at equi-affine registration of 2D image pairs.

2 Method of moving frames

The method of moving frames is a powerful tool used to study the geometric properties of submanifolds under the action of transformation (Lie) groups [21, 22]. In particular, the method allows direct computation of the differential invariants of a transformation group. The equivarant method of moving frames was introduced in Ref. [9] and permits a systematic application to general transformation groups acting on manifolds. The salient points of this theory are summarised here.

Let G be an r-dimensional Lie group, acting on an m-dimensional manifold, M. A *moving frame* ρ is defined as a G-equivariant map $\rho : M \to G$. We will exclusively use right equivariant moving frames, meaning that for $g \in G$ and z a local coordinate on M, we have

$$\rho(g \cdot z) = \rho(z) \cdot g^{-1}.$$

As we will demonstrate, the computation of a moving frame for a Lie group action will allow us to compute the fundamental differential invariants of the group action on the submanifolds of a given dimension, e.g. surfaces in $\mathbb{R}^3$.

The key theorem regarding the existence of such moving frames is as follows [21]:

Theorem 1 *A moving frame exists near a point $z \in M$ if and only if G acts freely and regularly near z.*

The freeness condition is best expressed in terms of isotropy subgroups of G. The isotropy subgroup of G with respect to $z \in M$ is the set

$$G_z = \{g \mid g \cdot z = z\},$$

that is, the set of all group elements which fix z. The group action is free if

$$G_z = \{e\} \text{ for all } z \in M, \tag{2}$$

where e is the group identity. In other words, the action is free if for any $z \in M$, the only group element fixing z is the identity. This requires that the orbits of G have the same dimension as G itself, and hence a necessary (but not sufficient) condition for freeness is that $\dim G \leq \dim M$. The group action is regular if the orbits form a regular foliation of M.

Remark: *In many cases arising in applications, the action is only locally free, meaning that the isotropy subgroup is discrete, or, equivalently the group orbits have dimension equal to the dimension of the group. Locally free actions produce locally equivariant moving frames and local invariants. A good example is the Euclidean group acting on plane curves. The action is only locally free on the first jet space since a* 180° *rotation does not affect the tangent to the*

https://doi.org/10.1017/S0956792519000020 Published online by Cambridge University Press

curve. As a result, the basic signed curvature differential invariant κ is only locally invariant since it changes sign under a 180° *rotation of the curve.*

If the group action is both free and regular, Cartan's method of normalisation is used to compute a moving frame. We begin by choosing a coordinate cross-section to the group orbits. A cross-section is a restriction of the form

$$K = \{z_1 = c_1, \ldots, z_r = c_r\} \tag{3}$$

i.e. we restrict r of the coordinates to be fixed. This cross-section must be transverse to the group orbits. By freeness, for any $z \in M$, there is a group element g mapping z to this cross-section, i.e. to the unique point in the intersection of the cross-section with the group orbit through z. This unique g is then the right moving frame for the action. This is formalised in the following theorem, which provides a method for practical construction of the right moving frame for a free and regular group action based on a given cross-section.

Theorem 2 *Let G act freely and regularly on M, and let K be a cross-section. For $z \in M$, let $g = \rho(z)$ be the unique group element mapping z to this cross-section:*

$$g \cdot z = \rho(z) \cdot z \in K. \tag{4}$$

Then $\rho : M \to G$ is a right moving frame for the group action.

The goal is to obtain the group transformation parameters $g = (g_1, g_2, \ldots, g_r)$ in equation (4) by applying the cross-section (3). Explicitly, if we write

$$w(g, z) = g \cdot z \tag{5}$$

then the cross-section choice gives us a system of r equations

$$w_j(g, z) = c_j, \qquad j = 1, 2, \ldots, r. \tag{6}$$

The system (6) is solved for the group parameters $g = g(g_1, \ldots, g_r)$ and the right moving frame is ultimately given as

$$g = \rho(z). \tag{7}$$

This group element clearly maps each z to the cross-section K since the group parameters were chosen to do so. Note that r of the coordinates of $g \cdot z$ are specified by the cross-section, and so there remain $m - r$ 'un-normalised' coordinates, namely $w_{r+1}(g, z), \ldots, w_m(g, z)$. If we substitute the previously computed moving frame parameters $g = \rho(z)$ into these coordinates, we obtain the fundamental invariants, as stated in the following theorem.

Theorem 3 *Let $g = \rho(z)$ be the moving frame solution to the normalisation equations (6). Then*

$$I_1(z) = w_{r+1}\big(\rho(z), z\big), \quad \ldots \ , \ I_{m-r}(z) = w_m\big(\rho(z), z\big)$$

are a complete system of functionally independent invariants, called the fundamental invariants.

https://doi.org/10.1017/S0956792519000020 Published online by Cambridge University Press

In this way, we have computed all of the coordinates of the map $z \mapsto \rho(z) \cdot z$ as

$$\rho(z) \cdot z = \big(c_1, \dots, c_r, I_1(z), \dots, I_{m-r}(z)\big) \tag{8}$$

As the name implies, the fundamental invariants are very useful, as evidenced by the following theorem.

Theorem 4 *Any invariant $I(z)$ can be uniquely expressed as a function*

$$I(z) = H\big(I_1(z), \dots, I_{m-r}(z)\big)$$

of the fundamental invariants.

Furthermore, given any (scalar) function on our manifold, we can 'invariantise' the function by composing it with the moving frame map.

Definition 5 Given a function $F : M \to \mathbb{R}$, the invariantisation of F with respect to right moving frame $g = \rho(z)$ is the invariant function $I = \iota(F)$ defined as

$$I(z) = F\big(\rho(z) \cdot z\big), \tag{9}$$

and is the unique invariant function that agrees with F on the cross-section.

As a result, invariantisation does not affect invariants: $I = \iota(I)$. In fact, the expression of invariants in terms of the fundamental invariants as in Theorem 4 can be accomplished from the following Replacement Theorem.

Theorem 6 *If $F(z_1, \dots, z_m)$ is any function, then we have*

$$\iota\big[F(z_1, \dots, z_m)\big] = F\big(c_1, \dots, c_r, I_1(z), \dots, I_{m-r}(z)\big).$$

In particular, if $I(z_1, \dots, z_m)$ is any invariant, then we have

$$\iota\big[I(z_1, \dots, z_m)\big] = I\big(c_1, \dots, c_r, I_1(z), \dots, I_{m-r}(z)\big).$$

Thus, any invariant is easily rewritten in terms of the fundamental invariants, which serves to prove Theorem 4.

A major difficulty arises when we attempt to apply this construction to some common group actions: many group actions are not free, typically because the dimension of the manifold is less than the dimension of the group. In these cases, the normalisation equations (6) cannot be fully solved for the group parameters. Examples of non-free group actions include the Euclidean and equi-affine groups on $\mathbb{R}^2$. Theorem 1 does not allow direct construction of moving frames for such non-free group actions. However, we may increase the dimension of the manifold by prolonging the group action to the jet spaces; that is, we consider instead the prolonged group action on the jet space of sufficiently high order n, which is coordinatised by the independent and dependent variables and their derivatives up to order n:

$$G^{(n)} : J^n(M, p) \to J^n(M, p). \tag{10}$$

(See below for explicit examples and [19] for the general theory.) Generally, if the action is prolonged to a high enough order jet space, the action will become (locally) free, [19], and we

https://doi.org/10.1017/S0956792519000020 Published online by Cambridge University Press

can proceed with our construction; we will have enough equations to solve system (6). This prolongation can be accomplished by implicit differentiation, as demonstrated in the following section.

3 Equi-affine invariants for 2D images

In this section, we follow the second author's note [23] to apply the above procedure to construct the basic 2D equi-affine differential invariants in the context of 2D image transformations. In particular, we are considering a greyscale image as a function $u : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$, and we seek the differential invariants of this function under equi-affine transformations of the domain coordinates, $(x, y) \in \Omega$. This expression in terms of image derivatives is useful since these can be computed (approximated) directly from image data.

The equi-affine group $\mathrm{SA}(2) = \mathrm{SL}(2) \ltimes \mathbb{R}^2$ acts on plane coordinates (x, y) as

$$\begin{pmatrix} z \\ w \end{pmatrix} = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} a \\ b \end{pmatrix}, \tag{11}$$

subject to the unimodular constraint

$$\alpha\delta - \beta\gamma = 1$$

required so that the transformation is area-preserving. The general transformation in SA(2) contains five independent parameters, and hence SA(2) is a five-dimensional Lie group. Note that these transformations are always invertible, with inverse

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \delta & -\beta \\ -\gamma & \alpha \end{pmatrix} \begin{pmatrix} z - a \\ w - b \end{pmatrix}. \tag{12}$$

We let $M = \mathbb{R}^3$ with coordinates x, y, u, where the image function u depends on x, y and is not affected by the equi-affine group transformations. Since $\dim G = 5 > \dim M = 3$, this group action is not free. Hence, Theorem 1 does not apply to the action on M, and we must prolong the group action to a higher-order jet space such that the action becomes free. For this example, we need only to prolong to the second jet space, which has dimension 8, with coordinates $x, y, u, u_x, u_y, u_{xx}, u_{xy}, u_{yy}$ corresponding to the derivatives of $u = f(x, y)$ with respect to x, y up to order 2. (Although the first-order jet space has dimension 5, the fact that u is invariant means that the action is not free at order 1.) We expect $8 - 5 = 3$ independent differential invariants, one of which is u itself.

The prolonged action of the transformation indicates how the coordinate mapping (11) will affect the image function u and its derivatives. Clearly, the function u is unchanged by a change of coordinates. Applying the chain rule of multivariate calculus, we can use the transformation (11) to construct the prolonged derivatives, that is, how the derivatives of the 'transformed' function relate to the original derivatives and the transformation parameters. For the x-coordinate,

$$u_z = u_x \frac{\partial x}{\partial z} + u_y \frac{\partial y}{\partial z} = \delta u_x - \gamma u_y. \tag{13}$$

Similarly,

$$u_w = -\beta u_x + \alpha u_y. \tag{14}$$

https://doi.org/10.1017/S0956792519000020 Published online by Cambridge University Press

More generally, this differentiation can be represented by the implicit differentiation operators

$$D_z = \delta D_x - \gamma D_y, \qquad D_w = -\beta D_x + \alpha D_y$$

where the D represents total differentiation operators with respect to their subscripts [19]. Continuing in this fashion, we can construct the higher-order prolonged derivatives by repeated application of these operators. The second-order derivatives are

$$u_{zz} = \delta^2 u_{xx} - 2\gamma\delta u_{xy} + \gamma^2 u_{yy}, \tag{15}$$

$$u_{ww} = \beta^2 u_{xx} - 2\alpha\beta u_{xy} + \alpha^2 u_{yy}, \tag{16}$$

$$u_{zw} = -\beta\delta u_{xx} + (\alpha\delta + \beta\gamma)u_{xy} - \alpha\gamma u_{yy}. \tag{17}$$

The next step in the procedure is to normalise by choosing a cross-section to the group orbits. This amounts to equating several of expressions (11), (13–17) to constants. These constants must be chosen such that the resulting system is solvable for the group parameters. Since SA(2) is five-dimensional, we need to choose five constants. The cross-section we will use is

$$K = \{z = w = u_z = u_{zw} = 0, \quad u_w = 1\}. \tag{18}$$

as in Ref. [23].

Remark: *Cross-sections are chosen so as to make the computations are easy as possible. Usually this means setting coordinates to zero unless this is not possible. For example, here if we were to set $u_w = 0$ we would not have a valid cross-section, which we would immediately detect by being unable to solve the resulting equations for the group parameters.*

The (nonlinear) system (18) is readily solved for the group parameters. Notice that since we normalised $u_z = 0$ and $u_w = 1$, we obtain

$$\delta = \gamma \frac{u_y}{u_x} \qquad \text{and} \qquad \alpha = \frac{1 + \beta u_x}{u_y}, \tag{19}$$

respectively. Combining these with the unimodularity constraint, $\alpha\delta - \beta\gamma = 1$ and then using expression (19), we find

$$\gamma = u_x \qquad \text{and} \qquad \delta = u_y. \tag{20}$$

Substituting these relationships into the second-order equation $u_{zw} = 1$, we can solve for β as

$$\beta = \frac{u_y u_{xy} - u_x u_{yy}}{u_x^2 u_{yy} - 2u_x u_y u_{xy} + u_y^2 u_{xx}}. \tag{21}$$

With this in hand, we substitute these expressions back into (19) to compute

$$\alpha = \frac{u_y u_{xx} - u_x u_{xy}}{u_x^2 u_{yy} - 2u_x u_y u_{xy} + u_y^2 u_{xx}}. \tag{22}$$

The translation parameters a and b can be computed easily from the choices $z = w = 0$ as

$$a = -\alpha x - \beta y \qquad \text{and} \qquad b = -\gamma x - \delta y, \tag{23}$$

https://doi.org/10.1017/S0956792519000020 Published online by Cambridge University Press

where the values found for the other parameters may be substituted in. Taken together, these formulas for the group parameters define the right moving frame $\rho : J^2(M, 2) \to \mathrm{SA}(2)$ corresponding to our cross-section K.

With the moving frame formulas in hand, we are now in a position to compute invariant objects. In particular, given any function F of u and its derivatives, we can invariantise F by transforming it under (11) and then replacing all instances of the group parameters by their moving frame expressions. Some natural functions to transform are those used in the cross-section normalisation. For example, let $F = u_x$ and note

$$u_x \mapsto u_z = \delta u_x - \gamma u_y.$$

Plugging in the above expressions,

$$\iota(u_x) = u_y u_x - u_x u_y = 0. \tag{24}$$

This is obvious, of course, since the expressions for the group parameters were chosen according to the cross-section in which we insisted $u_z = 0$. The invariantised forms of the functions used in the cross-section normalisation are constant, and are known as the phantom invariants [9, 23].

More interestingly, Theorem 3 guarantees the existence of a set of fundamental second-order invariants. Consider the second derivatives u_{zz} and u_{ww} which were not used in the normalisation. Invariantising these, we obtain the well-known second-order equi-affine differential invariants:

$$\iota(u_{zz}) = u_y^2 u_{xx} - 2u_x u_y u_{xy} + u_x^2 u_{yy} = J, \tag{25}$$

and

$$\iota(u_{ww}) = \frac{u_{xx}u_{yy} - u_{xy}^2}{J} = \frac{H}{J}. \tag{26}$$

According to Theorem 4, H and J form a complete system of second-order differential invariants. It can be checked directly via chain rule that these are truly equi-affine invariant. These fundamental equi-affine differential invariants were found in [26] where they were used to define an equi-affine invariant gradient for edge detection in digital images.

Figure 2 demonstrates H and J computed on the image pair in Figure 1. These functions are thresholded for clarity, but it can be observed that they are generally unchanged by the transformation. Differences can be attributed to the fact that our derivative computations via finite differences are computed on a Euclidean grid and are thus not equi-affine-invariant themselves.

Remark: *Moving frames can be used to design group-invariant numerical approximations to differential invariants and invariant differential equations, as in Refs. [7, 31]. A future project is to develop equi-affine finite difference approximations to the differential invariants, which would better serve to localise invariant feature points.*

Remark: *Higher order differential invariants can be computed by either further prolongation and invariantisation or by the method of invariant differentiation [21, 22].*

3.1 The 3D case

The extension to 3D image volumes has recently been investigated. Fully three-dimensional image data have become more popular recently, for example, in the medical field. Here we describe the computation of differential invariants of the equi-affine group acting on three-dimensional images.

https://doi.org/10.1017/S0956792519000020 Published online by Cambridge University Press

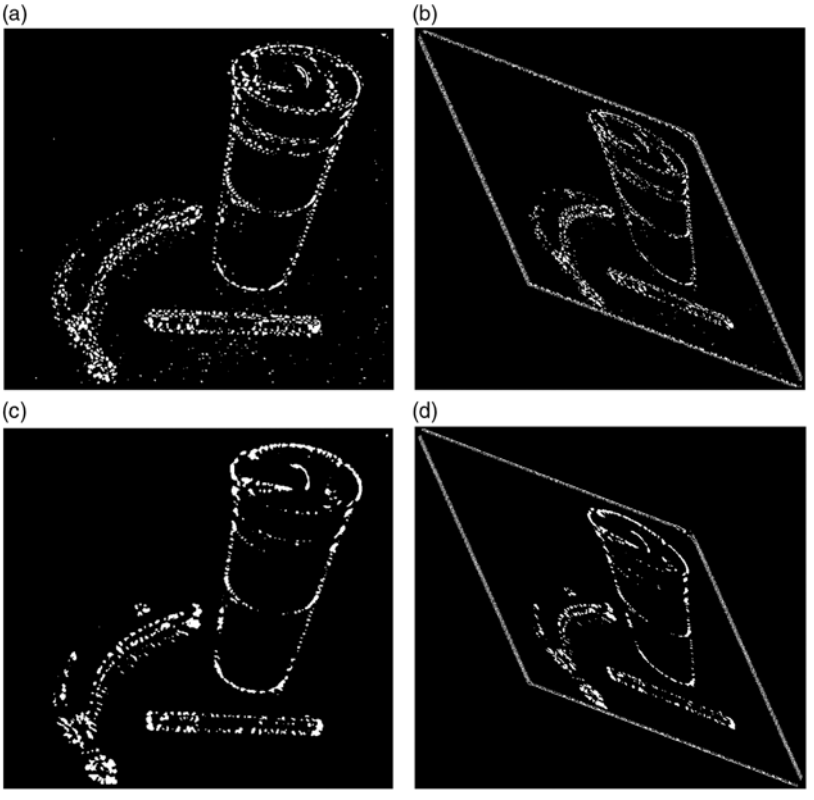


FIGURE 2. Thresholded differential invariants H (top) and J (bottom) for the images in Figure 1.

The equi-affine group $\mathrm{SA}(3) = \mathrm{SL}(3) \ltimes \mathbb{R}^3$ acts on space coordinates (x, y, z) as

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} \alpha & \beta & \gamma \\ \delta & \epsilon & \zeta \\ \eta & \theta & \lambda \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} a \\ b \\ c \end{pmatrix} = A \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \vec{a}, \tag{27}$$

subject to the constraint $\det(A) = 1$. Clearly, SA(3) is an 11-dimensional Lie group since there are 11 independent parameters. As in the two-dimensional case, this group action is certainly not free, and we must prolong the group action to a higher-order jet space such that the action becomes free. Interestingly, when we prolong to the second jet space, J^2, the group action is still not free, even though the dimensions satisfy

$$\dim J^2 = 13 > 11 = \dim G.$$

Through our constructions, we will produce 3 functionally independent second-order differential invariants, and this proves that the (generic) orbits are 10-dimensional, which precludes freeness of the second-order prolonged action.

We thus need to prolong the group action to the third-order jet space in order to find a usable cross-section. The cross-section we will employ is

$$K = \left\{ x = y = z = 0, \;\; u_x = u_y = u_{xy} = u_{yz} = u_{zz} = u_{zzz} = 0, \;\; u_z = u_{xz} = 1 \right\} \tag{28}$$

https://doi.org/10.1017/S0956792519000020 Published online by Cambridge University Press

Note that the variables u_{xx} and u_{yy} are not involved in the normalisation, and so the invariantisation of these quantities will give our desired fundamental second-order invariants.

The next step in the procedure would be to solve the system (28). In this case, this is a nonlinear system of 11 equations, and computing the closed-form solution is not as straightforward as it was in the 2D case. Instead, we will extend the 2D differential invariants H and J to the 3D case and then apply the Replacement Theorem 6. Recall that this theorem allows us to express any invariant in terms of the moving frame invariants. From the cross-section (28), we see

$$\iota(\nabla u) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad \text{and} \quad \iota(\nabla^2 u) = \begin{pmatrix} \iota(u_{xx}) & 0 & 1 \\ 0 & \iota(u_{yy}) & 0 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} L & 0 & 1 \\ 0 & K & 0 \\ 1 & 0 & 0 \end{pmatrix}, \tag{29}$$

where $L = \iota(u_{xx})$ and $K = \iota(u_{yy})$, not used in the normalisation, are the fundamental invariants we wish to determine.

Note that the invariants H and J found in the 2D case can be written in more general forms:

$$H = u_{xx}u_{yy} - u_{xy}^2 = \det\left(\nabla^2 u\right) \tag{30}$$

and

$$\frac{J}{H} = \frac{1}{\det\left(\nabla^2 u\right)}\left(u_x^2 u_{yy} - 2u_x u_y u_{xy} + u_y^2 u_{xx}\right) = \nabla u^T \left(\nabla^2 u\right)^{-1} \nabla u. \tag{31}$$

If we consider now extending these to the 3D case, where $u = u(x, y, z)$, the invariantisation results (29) give

$$H = \iota\left(\det\left(\nabla^2 u\right)\right) = -K \qquad \text{and} \qquad J/H = \iota\left(\nabla u^T \left(\nabla^2 u\right)^{-1} \nabla u\right) = -L. \tag{32}$$

That is, we can compute the fundamental invariants L and K for the 3D case directly as a generalisation of the 2D fundamental invariants J/H and H, respectively. Hence, the fundamental invariants are now expressed as

$$\iota(u_{xx}) = \nabla u^T \left(\nabla^2 u\right)^{-1} \nabla u = J/H, \qquad \iota(u_{yy}) = \det\left(\nabla^2 u\right) = H. \tag{33}$$

which can be computed directly from the image.

In Figure 3, we show these invariants for a typical 3D image. Several slices of an image volume are shown, along with the differential invariants computed on these slices. They exhibit similar characteristics as in the 2D case, being clustered about edges and corners, and are equi-affine invariant.

Note that we do not need to compute any third-order differential invariants. However, these can be generated from the second-order differential invariants by invariant differentiation, but are omitted for brevity since the computations prove somewhat lengthy.

4 Application: 2D feature detection with scale-space

In this section, the differential invariants H and J are combined into an equi-affine-invariant function which will serve as a feature detector. To make detection more robust to noise, we construct a scale-space of our images. Unlike traditional methods such as SIFT [16], our scale-space is nonlinear, affine-invariant, and based on a geometric curve evolution equation.

https://doi.org/10.1017/S0956792519000020 Published online by Cambridge University Press

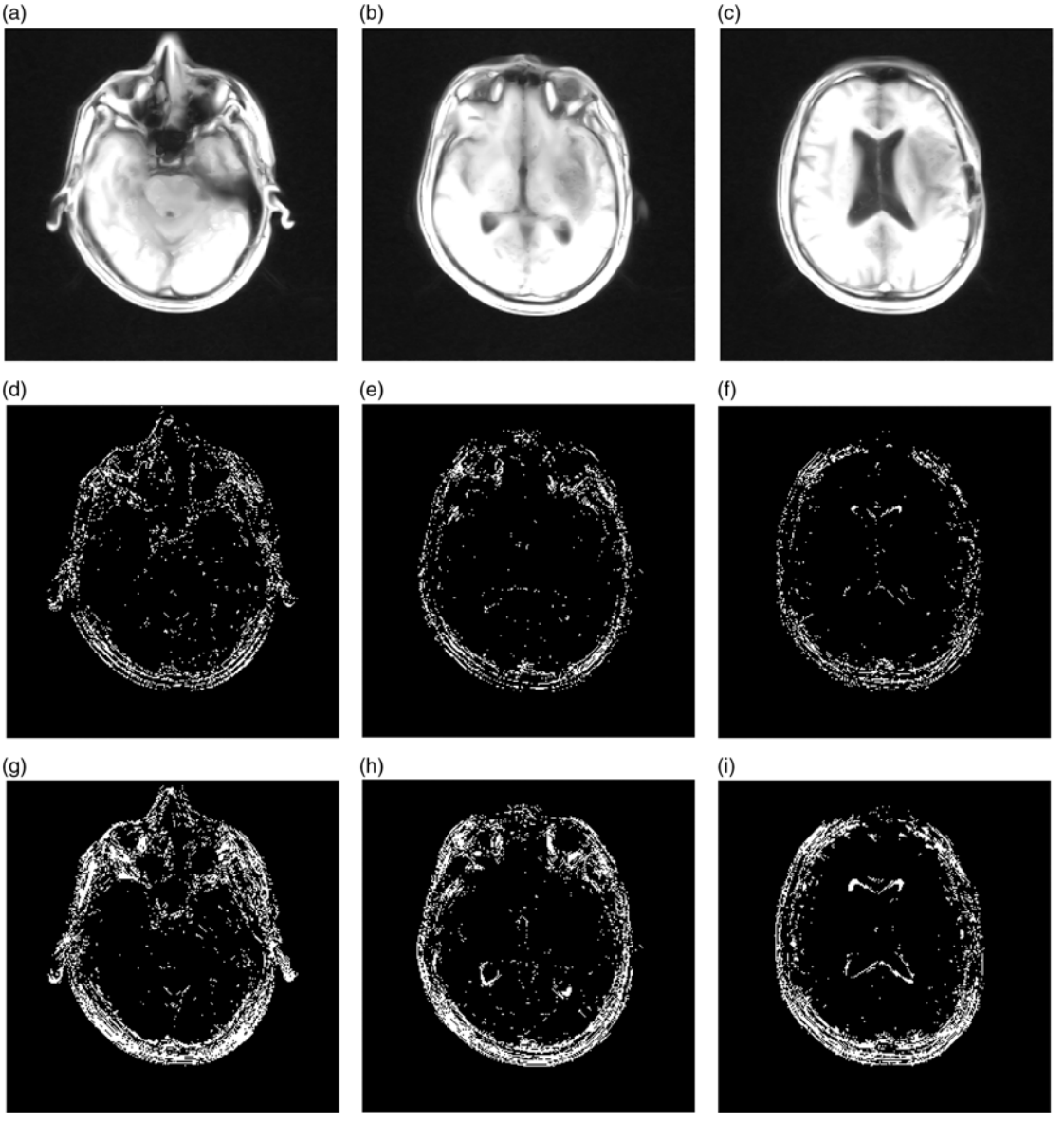


FIGURE 3. Thresholded differential invariants H (top) and J (bottom) for a 3D image of a human brain: image slices (top row), H (middle row), and J (bottom row).

4.1 Affine-invariant gradient magnitude

The equi-affine differential invariants computed above were used in Ref. [26] to construct an equi-affine-invariant analogue to the traditional Euclidean gradient magnitude, namely

$$\nabla_{\text{aff}} u = \left| \frac{H}{J} \right|. \tag{34}$$

https://doi.org/10.1017/S0956792519000020 Published online by Cambridge University Press

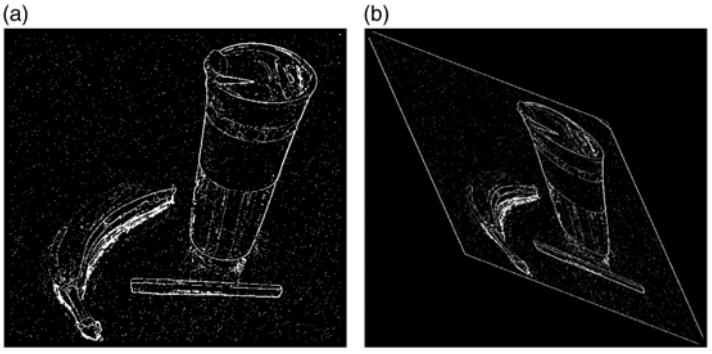


FIGURE 4. Equi-affine-invariant gradient magnitudes of the images in Figure 1, thresholded for clarity.

This function is slightly modified to avoid zero divisors:

$$\widehat{\nabla}_{\text{aff}}\, u = \sqrt{\frac{H^2}{J^2 + 1}}. \tag{35}$$

An example of this function applied to the image pair in Figure 1 is shown in Figure 4. This function was previously applied as an edge detector for an equi-affine-invariant active contour model since it gives high response along edges and corners, as is typical of the traditional Euclidean gradient magnitude [26]. Indeed, the function can be seen to give large response along object boundaries.

Some example feature points identified as local maxima of the function (35) are shown in Figures 6–8. As expected, these points largely cluster around object boundaries. This poses a difficulty, since edge points are not distinct and tend to foil matching algorithms. A further modification may be made such that, for example, candidate feature points are only retained as true feature points if image variation is significant in multiple directions.

4.2 Equi-affine-invariant scale-space

An important pre-processing step for feature detection is the construction of the Gaussian (or linear) scale-space of the image intensity signal by convolution of the image with Gaussian kernels of increasing standard deviation or, equivalently, evolution of the image via the linear diffusion equation

$$u_t = \Delta u, \tag{36}$$

with initial condition $u(x, 0) = u_0(x)$, where $u_0 : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ is the original image, Δ is the usual Laplacian operator, and t is an artificial time parameter. The solution to this equation, $u(x, t)$, for $x \in \mathbb{R}^2$ and $t \in [0, \infty)$ is the Gaussian scale-space of the image [32]. This is a multi-scale representation of the image, a one-parameter family of images $u(\cdot, t)$ with the characteristic that as t is increased, noise and high-frequency features are removed and we obtain successively smoothed versions of the original image data [30]. Feature points can then be extracted from this scale-space image representation by finding local detector maxima in both space and scale [16]. By searching across all scale-space, feature points can be identified at a variety of scales. A notable drawback of this linear approach is that important image features such as edges and corners are blurred by the smoothing, and so feature localisation is lost.

https://doi.org/10.1017/S0956792519000020 Published online by Cambridge University Press

Nonlinear scale-spaces have also been investigated for feature detection. The popular KAZE algorithm [2] relies on the Perona–Malik anisotropic diffusion framework in which an image scale-space is constructed as the solution to the nonlinear diffusion equation

$$\Phi_t = \operatorname{div}(g \cdot \nabla \Phi), \tag{37}$$

where g is a conductance function which serves to slow diffusion near edges in an image [27]. This image representation preserves edges better than its linear counterpart (36), which is obtained from (37) when $g \equiv 1$.

For a fully equi-affine-invariant multi-scale feature detection pipeline, we require both an equi-affine-invariant scale-space image representation and an equi-affine-invariant feature point detector. A fully equi-affine-invariant scale-space for plane curves was developed in Ref. [28]. In this work, a closed plane curve $C_0 : [0, 1] \to \mathbb{R}^2$ evolves according to the partial differential equation (PDE)

$$\frac{\partial C}{\partial t} = \kappa^{1/3} N, \tag{38}$$

where t is an artificial time parameter, κ is the Euclidean curvature of C, and N is the inward unit normal vector to C. The solution to this equation is a one-parameter family of plane curves, $C(\cdot, t)$, which evolve in an equi-affine-invariant manner to smooth out the curve. By evolving the PDE for larger amount of time, we obtain successively smoothed versions of the curve.

This nonlinear curve smoothing equation gives rise to an equi-affine-invariant scale-space for images through an application of the level set method for interface propagation [29]. Given image u, the level contours of u, namely the sets

$$K_c = \left\{(x, y) \in \mathbb{R}^2 \mid u(x, y) = c\right\} \tag{39}$$

for each $c \in \mathbb{R}$, are made to evolve simultaneously according to (38). Note that differentiating with respect to t along the k-level contour we obtain

$$\frac{d}{dt} u(x, y) = u_t + \nabla u \cdot C_t = 0, \tag{40}$$

where subscripts indicate partial derivatives. If we let this curve evolve according to (38),

$$C_t = \kappa^{1/3} N = -\kappa^{1/3} \frac{\nabla u}{\|\nabla u\|}, \tag{41}$$

where we used the fact that the gradient of a function is normal to the level curves of that function. Combining equations (40) and (41), we obtain

$$u_t = \kappa^{1/3} \, \|\nabla u\|. \tag{42}$$

Lastly, we wish to write (42) entirely in terms of u so that it may be applied directly to images without explicitly considering the level sets of the image. Hence, we require an expression for κ, the Euclidean curvature of the level contour, in terms of the image function u. This is well known and can be shown to be

$$\kappa = \operatorname{div}\left(\frac{\nabla u}{\|\nabla u\|}\right) = \frac{u_x^2 u_{yy} - 2u_x u_y u_{xy} + u_y^2 u_{xx}}{\left(u_x^2 + u_y^2\right)^{3/2}}. \tag{43}$$

https://doi.org/10.1017/S0956792519000020 Published online by Cambridge University Press

Substitution of this into (42) gives

$$u_t = \left(u_x^2 u_{yy} - 2u_x u_y u_{xy} + u_y^2 u_{xx}\right)^{1/3}, \tag{44}$$

which is the 2D equi-affine-invariant geometric heat equation, introduced in Ref. [26]. The equi-affine-invariant scale space of 2D images is constructed by solving the PDE (44) with initial data u_0, the original image. As with the curve evolution equation (38), we obtain smoothed versions of the image as we evolve in time. An example of this image smoothing is shown and compared with the linear case in Figure 5. This evolution is truly equi-affine-invariant, in the sense that for two initial images $u_1(x)$ and $u_2(x)$ related by an equi-affine transformation, the scale-space images at any time $t > 0$, namely $u_1(x, t)$ and $u_2(x)$, are related by the same equi-affine transformation.

These invariant flows are examples of a more general framework, developed in [24, 25], for constructing invariant curve and surface flows under general Lie groups. Of future interest, the curve flow PDE (42) was extended in Ref. [25] to an equi-affine-invariant surface evolution equation

$$S_t = \kappa_+^{1/4} \|\nabla S\| \tag{45}$$

where S is the surface in 3D and $\kappa_+ = \max\{0, \kappa\}$ is the (non-negative) mean curvature of S. Similar to above, this equation generates an equi-affine-invariant scale-space for 3D images and provides invariant smoothing with edge preservation.

In our feature detection pipeline, the equi-affine-invariant scale-space of an image is first computed by solving (44) via finite differences. As in traditional linear methods, we sample the scale-space at six or eight discrete times, and we search for candidate feature points over each sample image. In this way, we capture details at several different scales and levels of smoothness.

4.3 Examples

Once feature points are detected in both images, SURF descriptors are computed for each feature point [5]. These descriptors represent smoothed versions of the local Hessian matrix about the pixel of interest, which are known to represent local shape. These vectors are not equi-affine-invariant, and so we apply an affine region normalisation procedure prior to descriptor computation [4]. Points are matched between images by comparing their descriptor vectors. Different metrics can be used to assess the similarity between features. For example, the traditional Euclidean difference between the feature vectors is a simple choice, which we use here.

Once matches (correspondences) are found between the feature points, we may now align the point sets. Traditionally, an affine transformation can be fit to the correspondences by the least-squares method. However, this method considers every correspondence as equal. In reality, feature algorithms often return erroneous matches, and so we need a fitting algorithm that is robust to mismatches. One such robust estimation method is RANSAC for model fitting [8,10]. In this algorithm, candidate equi-affine transformation models are generated by selecting a minimal set of correspondences and computing the transformation based on only these points. Then, the number of correspondences that agree (within some threshold) with this model are counted and are called inliers. The entire process is repeated many times with different randomly-selected sets of correspondences. Ultimately, the generated transformation with the largest number of inliers

https://doi.org/10.1017/S0956792519000020 Published online by Cambridge University Press



FIGURE 5. First column: samples of linear (Gaussian) scale-space of an image. Second column: samples of the equi-affine-invariant scale-space of the same image. Scale parameter t increases as we travel down the columns.

https://doi.org/10.1017/S0956792519000020 Published online by Cambridge University Press

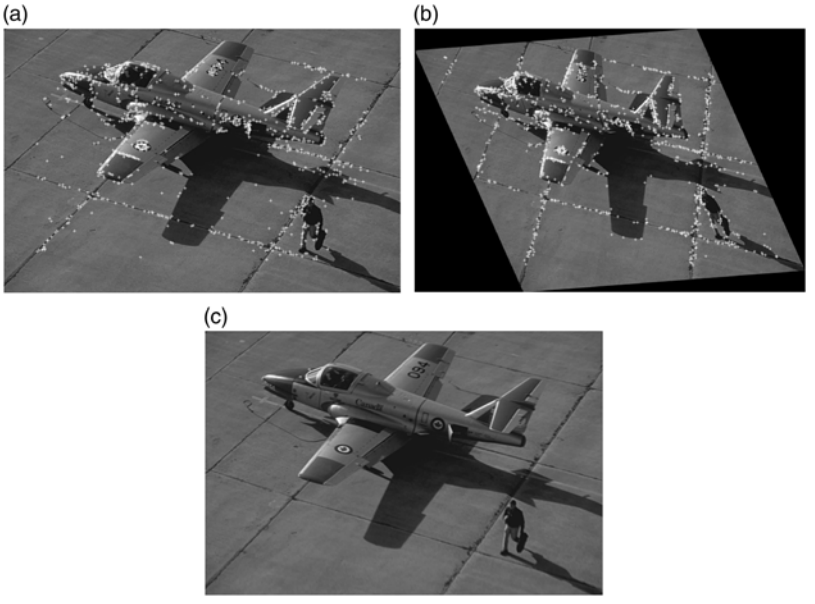


FIGURE 6. Feature points and alignment of an equi-affine related image pair. A very good registration is obtained as a result of the large amount of variety of neighbourhoods about the feature points (e.g. across the body of the plane).

is selected as the ideal model. In this manner, outliers are filtered out, since models generated using outliers will not have a large number of inliers.

Example alignments of equi-affine transform image pairs are shown in Figures 6–8. In Figure 6, the computed transformation does a very good job of aligning the images; any differences between the overlaid images are difficult to notice. This image has a lot of variety, and so we expect to do well here.

The computed equi-affine transform in Figure 7 exhibits errors. This is likely due to the equi-affine-invariant gradient detector exhibiting strong response along object edges. Edge points are difficult to match since they often only vary in one direction, and so could be believably matched to any other pixel along the same edge. The transform shown in Figure 8 suffers from the same limitations.

Another limitation of the method is that the differential invariant computations are only approximations via finite differences using traditional rectangular grids. As such, they are imperfect, and even a close inspection of Figure 2 will reveal that the responses for the two images are not exactly equal up to equi-affine transformation. This results in inaccurate localisation of feature points, and compounds the aforementioned issue with the edge points. A further investigation might pursue more accurate (invariant) numerical derivative approximations, but these would slow the algorithm.

Despite these limitations, the algorithm performs well for using no prior information. The method might be used as a rough initial registration, after which a more sophisticated deformable registration might be applied. Registration could be improved by incorporating other detectors (e.g. Harris corners [14]) which, though perhaps not equi-affine-invariant, may return points which could be reliably matched.

https://doi.org/10.1017/S0956792519000020 Published online by Cambridge University Press

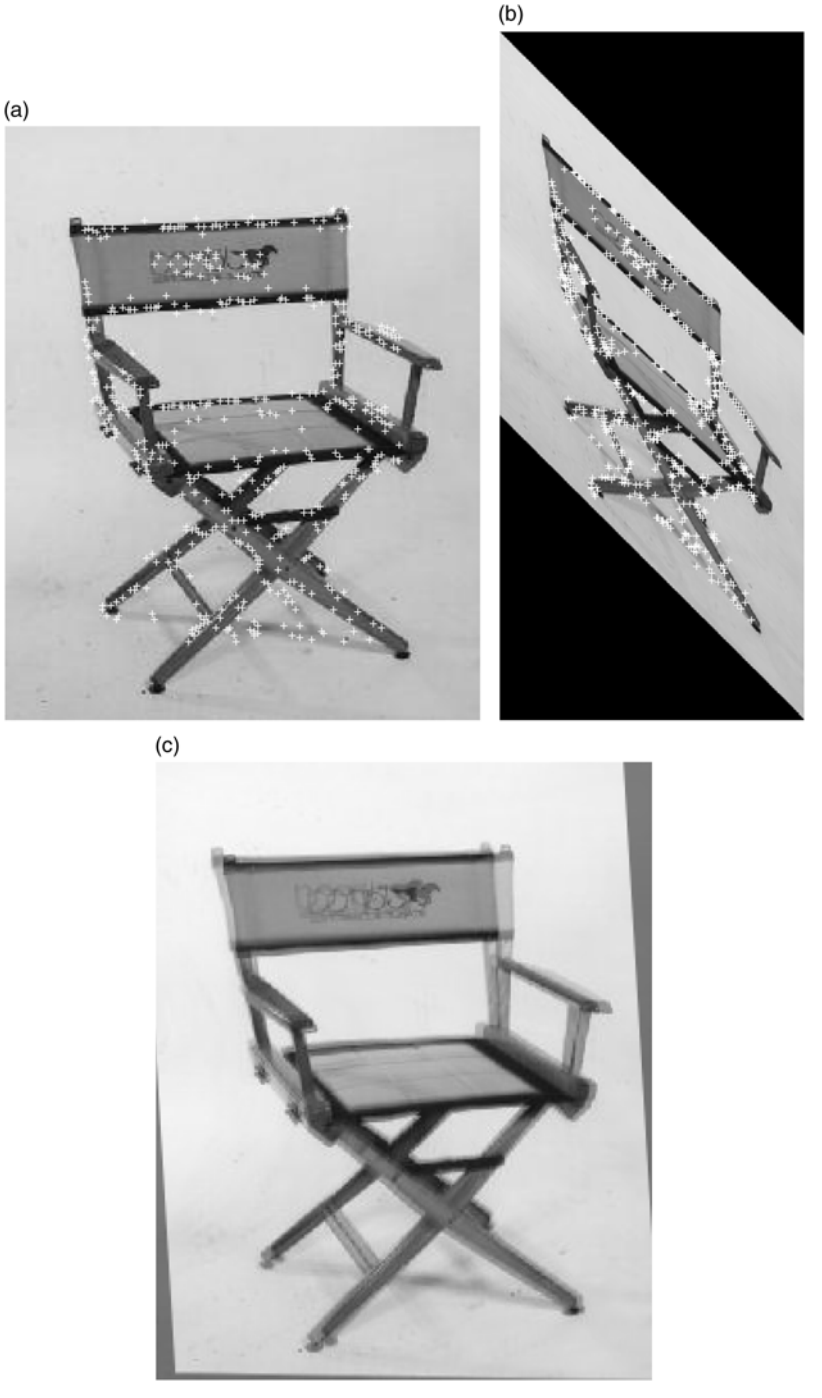


FIGURE 7. Feature points and alignment of an equi-affine related image pair. The homogeneity of the background and the presence of unremarkable edges leads to an imperfect alignment.

https://doi.org/10.1017/S0956792519000020 Published online by Cambridge University Press

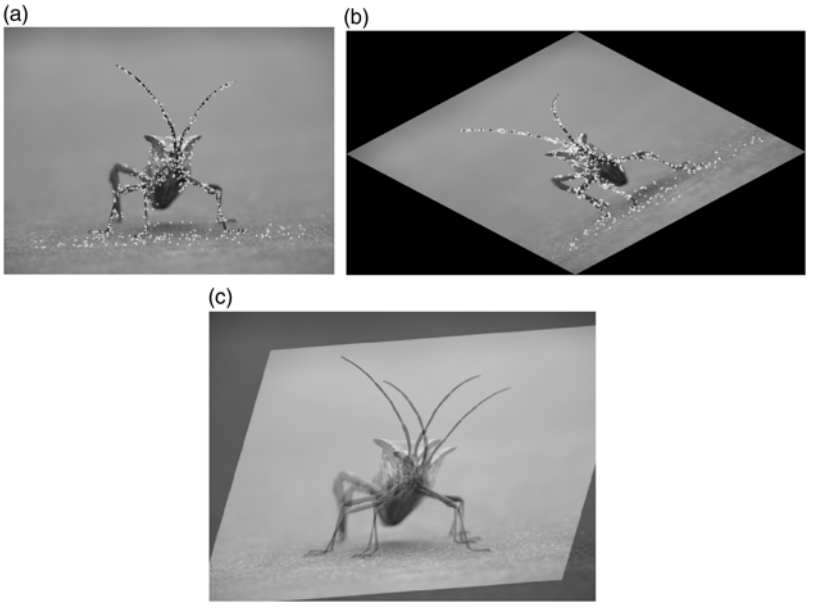


FIGURE 8. Feature points and alignment of an equi-affine related image pair. The homogeneity of the background and the presence of unremarkable edges leads to an imperfect alignment.

5 Future directions

We have computed the fundamental equi-affine differential invariants for 3D image volumes. A future investigation will focus on the application of these invariants to an analogous invariant feature point detection and registration pipeline.

Moving frames have been used to design group-invariant numerical approximations, as in Refs. [7, 31]. These methods may be applied to develop equi-affine finite difference approximations to the differential invariants, which would better serve to localise invariant feature points.

The most interesting invariants for computer vision applications are for the 3D-to-2D projective group, since these transformations model how the real world is projected onto the image plane of a camera. Unfortunately, the projective group invariants are numerically difficult in that they involve derivatives of orders higher than two, and this is prohibitive for applications in which speed is a priority. The computation of projective invariants is being investigated. Related works connect the differential invariants of 3D curves and their 2D projections through the method of moving frames [6, 15].

6 Conclusion

In this paper, we have shown how the equivariant method of moving frames is used to compute the fundamental second-order differential invariants of the equi-affine group acting on scalar functions on $\mathbb{R}^2$. These invariants were used to construct an equi-affine-invariant feature point detector function, which was demonstrated to perform well at alignment of equi-affine-related image pairs. The differential invariants for the equi-affine group acting on 3D image volumes are also computed, and the extension of the 2D pipeline to 3D image volumes (e.g. Magnetic Resonance Imaging (MRI)) is an interesting future direction.

https://doi.org/10.1017/S0956792519000020 Published online by Cambridge University Press

Conflict of interest

None.

References

[1] AKIVIS, M. & ROSENFELD, B. (1993) *Élie Cartan (1869–1951)*. Translations Math. Monographs, Vol. 123, American Math. Soc., Providence, R.I.

[2] ALCANTARILLA, P. F., BARTOLI, A. & DAVISON, A. J. (2012) Kaze features. In: *European Conference on Computer Vision*, Springer, pp. 214–227.

[3] ASTROM, K. (1995) Fundamental limitations on projective invariants of planar curves. *IEEE Trans. Pattern Anal. Mach. Intell.* **17** (1), 77–81.

[4] BAUMBERG, A. (2000) Reliable feature matching across widely separated views. In: *Proceedings IEEE Conference on Computer Vision and Pattern Recognition, 2000*, Vol. 1. IEEE, pp. 774–781.

[5] BAY, H., TUYTELAARS, T. & GOOL, L. V. (2006) *Surf: Speeded up Robust Features*, *ECCV 2006*, Springer, pp. 404–417.

[6] BURDIS, J. M., KOGAN, I. A. & HONG, H. (2013) Object-image correspondence for algebraic curves under projections. *SIGMA: Symmetry Integr. Geom. Methods Appl.* **9** (023), 1–31.

[7] CALABI, E., OLVER, P. J. & TANNENBAUM, A. (1996) Affine geometry, curve flows, and invariant numerical approximations. *Adv. Math.* **124** (1), 154–196.

[8] COLLEU, T., SHEN, J.-K., MATUSZEWSKI, B., SHARK, L.-K. & CARIOU, C. (2006) Feature-based deformable image registration with ransac based search correspondence. In: *AECRIS'06-Atlantic Europe Conference on Remote Imaging and Spectroscopy*, pp. 57–64.

[9] FELS, M. & OLVER, P. J. (1999) Moving coframes: II. regularization and theoretical foundations. *Acta Appl. Math.* **55**(2), 127–208.

[10] FISCHLER, M. A. & BOLLES, R. C. (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM.* **24** (6), 381–395.

[11] GUGGENHEIMER, H. (1963) *Differential Geometry*, McGraw–Hill, New York.

[12] HANN, C. E. (2001) *Recognising Two Planar Objects Under a Projective Transformation.* PhD dissertation, University of Canterbury, Mathematics and Statistics.

[13] HANN, C. E. & HICKMAN, M. (2002) Projective curvature and integral invariants. *Acta Appl. Math.* **74**(2), 177–193.

[14] HARRIS, C. & STEPHENS, M. (1988) A combined corner and edge detector. In: *Alvey Vision Conference*, Vol. 15, no. 50, Manchester, UK, pp. 10–5244.

[15] KOGAN, I. A. & OLVER, P. J. (2015) Invariants of objects and their images under surjective maps. *Lobachevskii J. Math.* **36**(3), 260–285.

[16] LOWE, D. G. (2004) Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*. **60**(2), 91–110.

[17] MIKOLAJCZYK, K. & SCHMID, C. (2005) A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(10), 1615–1630.

[18] MIKOLAJCZYK, K., TUYTELAARS, T., SCHMID, C., MATAS, A. Z. J., SCHAFFALITZKY, F., KADIR, T., & GOOL, L. V. (2005) A comparison of affine region detectors. *Int. J. Comput. Vision.* **65** (1–2), 43–72.

[19] OLVER, P. J. (1995) *Equivalence, Invariants, and Symmetry*, Cambridge University Press, Cambridge, UK.

[20] OLVER, P. J. (2001) Joint invariant signatures. *Found. Comput. Math.* **1**(1), 3–67.

[21] OLVER, P. J. (2003) Moving frames. *J. Symb. Comput.* **36**(3), 501–512.

[22] OLVER, P. J. (2015) Modern developments in the theory and applications of moving frames. In: *Impact 150 Stories*, Vol. 1, London Mathematical Society, London, UK, pp. 14–50.

[23] OLVER, P. J. (2015) *Moving Frame Derivation of the Fundamental Equi-Affine Differential Invariants for Level Set Functions*. preprint, University of Minnesota, Minneapolis, MN, USA.

[24] OLVER, P. J., SAPIRO, G. & TANNENBAUM, A. (1994) Differential invariant signatures and flows in computer vision: a symmetry group approach. In: B. M. Ter Haar Romeny (editor),

https://doi.org/10.1017/S0956792519000020 Published online by Cambridge University Press

Geometry-Driven Diffusion in Computer Vision, Kluwer Acad. Publ., Dordrecht, The Netherlands, pp. 255–306.

[25] OLVER, P. J., SAPIRO, G. & TANNENBAUM, A. (1997) Invariant geometric evolutions of surfaces and volumetric smoothing. *Siam. J. Appl. Math.* **57** (1), 176–194.

[26] OLVER, P. J., SAPIRO, G. & TANNENBAUM, A. (1999) Affine invariant detection: edge maps, anisotropic diffusion, and active contours. *Acta Appl. Math.* **59** (1), 45–77.

[27] PERONA, P. & MALIK, J. (1990) Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.* **12** (7), 629–639.

[28] SAPIRO, G. & TANNENBAUM, A. (1993) Affine invariant scale-space. *Int. J. Comput. Vision*. **11** (1), 25–44.

[29] SETHIAN, J. A. (1999) *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, Cambridge, UK.

[30] TER HAAR ROMENY, B. M. (1994) *Geometry-Driven Diffusion in Computer Vision*. Kluwer Acad. Publ., Dordrecht, The Netherlands.

[31] WELK, M., KIM, P. & OLVER, P. J. (2007) Numerical invariantization for morphological PDE schemes. In: *International Conference on Scale Space and Variational Methods in Computer Vision*. Springer, pp. 508–519.

[32] WITKIN, A. (1984) Scale-space filtering: a new approach to multi-scale description. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'84*, Vol. 9. IEEE, pp. 150–153.

[33] YU, G. & MOREL, J.-M. (2011) ASIFT: an algorithm for fully affine invariant comparison. *Image Process. On Line*. **1**, 11–38.

https://doi.org/10.1017/S0956792519000020 Published online by Cambridge University Press