# ICP Registration Using Invariant Features

Gregory C. Sharp, *Student Member*, *IEEE*, Sang W. Lee, *Member*, *IEEE*, and David K. Wehe, *Member*, *IEEE*

**Abstract**—This paper investigates the use of Euclidean invariant features in a generalization of iterative closest point registration of range images. Pointwise correspondences are chosen as the closest point with respect to a weighted linear combination of positional and feature distances. It is shown that under ideal noise-free conditions, correspondences formed using this distance function are correct more often than correspondences formed using the positional distance alone. In addition, monotonic convergence to at least a local minimum is shown to hold for this method. When noise is present, a method that automatically sets the optimal relative contribution of features and positions is described. This method trades off error in feature values due to noise against error in positions due to misalignment. Experimental results suggest that using invariant features decreases the probability of being trapped in a local minimum and may be an effective solution for difficult range image registration problems where the scene is very small compared to the model.

**Index Terms**—Registration, range images, feature detection, invariance.

## 1 Introduction

Building 3D models of real world objects for reverse engineering, facility mapping, and computer graphics applications typically requires three stages: a data capture stage which samples the 3D world using a range camera, a data registration stage which aligns the various 3D views, and a data merge stage which simplifies the aligned views into parametric models. The goal of the registration stage is to find the relative position and orientation of each view with respect to each other view. This paper addresses the use of features for improving the probability of convergence of a popular solution to the problem of matching and aligning range images, the iterative closest point registration algorithm. We present a theoretical basis for the use of invariant features, an automatic method for selecting the trade-off between features and positions, and an experimental evaluation demonstrating improved convergence.

Range image registration is typically accomplished using a variant of the iterative closest point algorithm (ICP) [4]. ICP is an iterative descent procedure which seeks to minimize the sum of the squared distances between all points in one of the views (the scene) and their closest points in the other view (the model). When a scene and a model can be represented as two point sets with known correspondences, the rigid motion that best aligns the scene in a least square sense can be solved in closed form according to the method of Faugeras and Hebert [12] or the method of Horn [15]. Traditional registration methods construct the correspondence sets by extracting salient显著的 features from the scene and model and perform a search procedure to match the features. In ICP registration, however, Besl and McKay solve the correspondence problem by assuming that the scene is approximately aligned with the model and, therefore, that each scene point corresponds with its closest model point [4]. Zhang extended ICP to include robust statistics and adaptive thresholding to handle outliers and occlusions [29]. Masuda and Yokoya use ICP with random sampling and a least median square error measurement that is robust to a partially overlapping scene [19]. Chen and Medioni independently developed an approach similar to ICP, which minimizes the sum of squared distance between scene points and a local planar approximation of the model [7]. Correspondences are formed by projecting the scene points onto the model in the direction of their normal vectors rather than selecting the closest point. Dorai et al. extend the method of Chen and Medioni to an optimal weighted least-squares framework [9]. These methods have been further extended to make simultaneous registration over multiple views possible [2], [25], [11].

Since ICP is an iterative descent algorithm, it requires a good initial estimate in order to converge to the global minimum. A fully automated registration algorithm can choose to use multiple initial conditions sampled randomly or uniformly throughout the search space in order to ensure that the goal is found [6]. The search space is large, however, requiring many initial conditions. Therefore, several researchers have used features, either alone or together with positions, in order to improve the registration. Chua and Jarvis use principal curvatures to constrain a heuristic search for correspondences [8]. Higuchi, et al. build a spherical map of curvature values called an SAI for each view of an object [14]. The SAI are registered by rotating the spheres until the curvature values are aligned.

- *G.C. Sharp is with the Department of Electrical Engineering and Computer Science, University of Michigan, 1101 Beal Ave., Ann Arbor, MI 48109. E-mail: gsharp@umich.edu.*
- *S.W. Lee is with the School of Media and Communications, Sogang University, 1 Shinsu-Dong, Mapo-Ku, Seoul, Korea 121-742. E-mail: slee@ccs.sogang.ac.kr.*
- *D.K. Wehe is with the Department of Nuclear Engineering and Radiological Sciences, University of Michigan, 2355 Bonisteel Blvd., Ann Arbor, MI 48109. E-mail: dkw@umcih.edu.*

*Manuscript received 27 Sept. 2000; revised 27 Feb. 2001; accepted 08 Mar. 2001.*
*Recommended for acceptance by S. Sclaroff.*
*For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 112915.*

Feldmar and Ayache perform affine registration by minimizing the combined distance between positions, surface normals and curvatures [13]. Thirion uses crest lines to extract extremal points and their associated Darboux frames, which are matched in an ICP-like fashion [26]. Soucy and Ferrie locally register surface patches by minimizing the distance between Darboux frames over an entire neighborhood [24]. Yang and Allen minimize a scaled product of positional and curvature distances [28]. VandenWyngaerd et al. match bitangent curve pairs, which are pairs of curves that share the same tangent plane, between two views for rigid and affine registration [27]. Johnson uses invariants derived from the spin-image, a histogram of distances and angles to nearby surface points, to perform recognition and registration of 3D range maps [18], [17].

Our work investigates a particular instance of the feature-based ICP approach, which we call Iterative Closest Points using Invariant Features (ICPIF) [23]. This method chooses nearest-neighbor correspondences according to a distance metric which is a scaled sum of the positional and feature distances. We show that under ideal, noise free conditions, correct correspondences are chosen at least as often using ICPIF as they would be using traditional ICP. In addition, we show that ICPIF converges monotonically to a local minimum in the same manner as traditional ICP. An automatic method for computing the trade-off between positions and features is demonstrated. Experimental evidence demonstrates that ICPIF converges to the goal state in fewer iterations than traditional ICP and, that, it converges to the goal state for more initial transformations.

Section 2 of this paper introduces the ICP algorithm and three different invariant values that may be computed directly from range data. Section 3 introduces ICPIF, an extension of the ICP algorithm that uses features to improve the correspondence search. In Section 4, we analyze the noise-free properties of ICPIF and in Section 5, we describe the trade-offs between position and feature values under a Gaussian noise model. Section 6 presents experimental results on simulated and real range data and Section 7 presents concluding remarks.

## 2 Background

### 2.1 Iterative Closest Point Registration

*Iterative closest point registration* (ICP) is an accurate and reliable method for the registration of free form surfaces [4]. The goal of ICP is to find the rigid transformation $\mathbf{T}$ that best aligns a cloud of scene points $\mathcal{S}$ with a geometric model $\mathcal{M}$. The alignment process works to minimize the mean squared distance between scene points and their closest model point. ICP is efficient, with average case complexity of $O(n \log n)$ for $n$ point images and it converges monotonically to a local minimum. At each iteration, the algorithm computes correspondences by finding closest points and, then, minimizes the mean square error in position between the correspondences [12], [15]. A good initial estimate of the transformation is required and all scene points are assumed to have correspondences in the model.

**Algorithm 1 (Iterative Closest Point Registration).** Let $\mathcal{S}$ be a set of $N_s$ points, $\{\mathbf{s}_1, \cdots, \mathbf{s}_{N_s}\}$ and, let $\mathcal{M}$ be the model. Let $\|\mathbf{s} - \mathbf{m}\|$ be the distance between point $\mathbf{s} \in \mathcal{S}$ and $\mathbf{m} \in \mathcal{M}$ and, let $\mathrm{CP}(\mathbf{s}_i, \mathcal{M})$ be the closest point in $\mathcal{M}$ to the scene point $\mathbf{s}_i$.

1. Let $\mathbf{T}_0$ be an initial estimate of the transformation.
2. Repeat for $k = 1..k_{\max}$ or until termination criteria is met
   a. Build up the set of correspondences $\mathcal{C} = \bigcup_{i=1}^{N_s} \{(\mathbf{T}_{k-1}(\mathbf{s}_i), \mathrm{CP}(\mathbf{T}_{k-1}(\mathbf{s}_i), \mathcal{M}))\}$.
   b. Compute the new transformation $\mathbf{T}_k$ that minimizes mean square error between point pairs in $\mathcal{C}$ [12], [15].

### 2.2 Invariant Features

The ICP algorithm uses the closest model point as a good estimate of a correct correspondence, an assumption that fails when the scene is not approximately aligned with the model. In these cases, shape descriptors may provide additional information to improve the correspondence search. Shape descriptors are computed directly from two or more sensed views and, therefore, must be invariant to rigid camera motion. We are most interested in Euclidean invariants, quantities that are invariant to 3D rigid motion. Neither scale invariance nor affine invariance are needed for rigid registration.

#### 2.2.1 Curvature曲率

Curvature is perhaps the most familiar of all invariants. In contrast to the global invariants computed from moments and spherical harmonics described below, it is a local attribute associated with a single surface point. For this study we use the magnitudes of the principle curvatures, computed by estimating the surface normals at each point and then differentiating. This method for computing discrete curvature is well-known for its sensitivity to both sensor noise and sampling rate. However, it is still useful for its simplicity and efficiency. A complete reference for computing curvatures in range images is found in the work of Besl and Jain [3].

#### 2.2.2 Moment Invariants

Sadjadi and Hall derive the second order moment invariants in three dimensions as

$$
\begin{aligned}
J1 &= \mu_{200} + \mu_{020} + \mu_{002} \\
J2 &= \mu_{200}\mu_{020} + \mu_{200}\mu_{002} + \mu_{020}\mu_{002} - \mu_{110}^2 - \mu_{101}^2 - \mu_{011}^2 \\
J3 &= \mu_{200}\mu_{020}\mu_{002} + 2\mu_{110}\mu_{101}\mu_{011} - \mu_{002}\mu_{110}^2 \\
&\quad - \mu_{020}\mu_{101}^2 - \mu_{200}\mu_{011}^2,
\end{aligned}
$$

where the $\mu_{pqr}$ are the centralized moments. The centralized moments are defined as

$$
\mu_{pqr} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-\bar{x})^p (y-\bar{y})^q (z-\bar{z})^r \rho(x,y,z)\,dx\,dy\,dz,
$$

where $\rho(x, y, z)$ is a piecewise continuous density function of finite support and $(\bar{x}, \bar{y}, \bar{z})$ is the centroid of $\rho(x, y, z)$ [22]. By using centralized moments, the coordinate system center is fixed at the center of mass and invariance to translation is

什么叫positional distance

矩和球谐函数?

能否进行端到端地进行几何特征的提取?

计算曲率有一个很大的问题:对Noise敏感

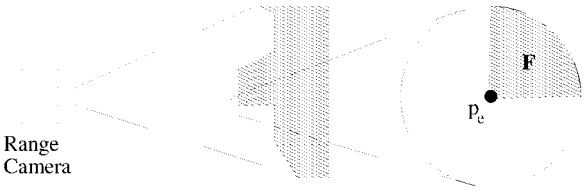


Fig. 1. The density function $\rho$ for point $\mathbf{p}_e$ is 1 within region $\mathbf{F}$ and 0 elsewhere.

achieved. The moment forms provide invariance to orthogonal transformations including 3D rotations.

#### 2.2.3 Spherical Harmonics Invariants

球谐系数

Burel and Henocq describe a method for deriving rotationally invariant features from the spherical harmonics coefficients of a global signal [5]. We consider only the simplest of these methods, the $N$ series of invariants. First, rank 1 tensors are constructed from the basis function coefficients $c_l^m$, where

$$c_l^m = \int_0^{2\pi} \int_0^{\pi} \sin(\theta) Y_{lm}^*(\theta, \phi) \rho(\theta, \phi) d\theta d\phi,$$

$Y_{lm}(\theta, \phi)$ are the harmonic bases and $\rho(\theta, \phi)$ is the density function. The coefficients form a contravariant tensor $c_l^m = (c_l^{-l}, ...c_l^l)$ and a covariant tensor $c_{lm} = (c_l^m)^*$, for which

逆变的

$$N(l) = \sum_{m=-l}^{l} c_l^m (c_l^m)^*$$

is invariant to rotation. Translation invariance may be achieved in the same manner as was done for the moment invariants, by fixing the coordinate system center.

#### 2.2.4 Defining a Global Density Function

Moment invariants and spherical harmonics invariants are global attributes of a three-dimensional signal defined over the entire space. In order to use these features to describe a surface point $\mathbf{p}_e$, we must ascribe to that point a global density function determined by the local geometry of the surface. The given point may serve as the center of its own local coordinate system and a local region $\mathbf{F}$ may be defined to be the space that is both 1) behind the scanned surface and 2) within a sphere of known radius centered at $\mathbf{p}_e$ (see Fig. 1). The density $\rho$ is then defined to be one within $\mathbf{F}$ and zero outside of $\mathbf{F}$. Although we are free to choose the center of mass as the center of the coordinate system, as is done for object recognition, it is more convenient and still translationally invariant to choose $\mathbf{p}_e$ as the coordinate center. At the image boundary, special processing is required because it is not possible to know the entire shape within this sphere. Therefore, invariant value contributions for points lying outside the image boundary are computed using a locally planar estimate based on the neighborhood points within the image boundary.

## 3 ICP Using Invariant Features

### 3.1 Notation

We shall use the term *ICP using invariant features* (ICPIF) to describe the use of invariant features in a modified distance function for correspondence selection. The specific method that we use is most similar to the method of Feldmar and Ayache [13], where each data point is represented as the concatenation of its three positional coordinates with $k$ feature coordinates. Points are matched using the $L_2$ norm in the $k+3$ dimensional space. The positional components shall be denoted $\mathbf{p}_e$ and its feature components $\mathbf{p}_f$. That is,

每一个点的描述分为：欧式坐标；特征描述；是pointwise的描述

$$\begin{aligned} \mathbf{p}_e &= (p_x, p_y, p_z) \in \mathbb{R}^3 \\ \mathbf{p}_f &= (p_{f_1}, p_{f_2}, \cdots, p_{f_k}) \in \mathbb{R}^k \\ \mathbf{p} &= (\mathbf{p}_e, \mathbf{p}_f) \in \mathbb{R}^{3+k}, \end{aligned}$$

where $p_{f_1}$ through $p_{f_k}$ are the $k$ invariant features describing point $\mathbf{p}$. When necessary, the notation $(\mathbf{p}_i)_e$ and $(\mathbf{p}_i)_f$ will be used to refer to the (Euclidean) positional and feature components of point $\mathbf{p}_i$. The combined positional and feature distance between $\mathbf{p}$ and $\mathbf{q}$ shall be denoted as

$$d(\mathbf{p}, \mathbf{q}) = d_e(\mathbf{p}, \mathbf{q}) + d_f(\mathbf{p}, \mathbf{q}),$$

where

每一个点的计算方式

$$\begin{aligned} d(\mathbf{p}, \mathbf{q}) &= \|\mathbf{p} - \mathbf{q}\|^2 \\ d_e(\mathbf{p}, \mathbf{q}) &= \|\mathbf{p}_e - \mathbf{q}_e\|^2 \\ d_f(\mathbf{p}, \mathbf{q}) &= \|\mathbf{p}_f - \mathbf{q}_f\|^2. \end{aligned}$$

The weighted feature distance is defined as

$$d_\alpha(\mathbf{p}, \mathbf{q}) = d_e(\mathbf{p}, \mathbf{q}) + \alpha^2 d_f(\mathbf{p}, \mathbf{q}), \quad (1)$$

where $\alpha$ controls the relative contribution of the positions and features. The closest point in $\mathcal{M}$ to a scene point $\mathbf{s}$ according to the distance measure $d_e$ shall be denoted $\mathrm{CP}(\mathbf{s}, \mathcal{M})$ and the closest point according to the distance measure $d_\alpha$ shall be denoted $\mathrm{CP}_\alpha(\mathbf{s}, \mathcal{M})$.

两个点不一样?

### 3.2 ICPIF Algorithm

The ICPIF algorithm performs ICP using closest point correspondences using $\mathrm{CP}_\alpha(\mathbf{s}, \mathcal{M})$. At this point, we shall assume that the user has heuristically selected an appropriate value for $\alpha$.

**Algorithm 2 (Iterative Closest Point Registration Using Invariant Features).** Let $\mathcal{S}$ be a set of $N_s$ points, $\{\mathbf{s}_1, \cdots, \mathbf{s}_{N_s}\}$ and, let $\mathcal{M}$ be the model.

1. Let $\mathbf{T}_0$ be an initial estimate of the transformation.
2. Repeat for $k = 1..k_{\max}$ or until termination criteria is met
   a. Build up the set of correspondences
   $$\mathcal{C} = \bigcup_{\mathbf{s}_i \in \mathcal{S}} \{(\mathbf{T}_{k-1}(\mathbf{s}_i), \mathrm{CP}_\alpha(\mathbf{T}_{k-1}(\mathbf{s}_i), \mathcal{M}))\}.$$
   b. Compute the new transformation $\mathbf{T}_k$ that minimizes mean square error between point pairs in $\mathcal{C}$.

pointwise 的方法，ICP 的变种似乎都可以进行借鉴

For scenes that contain occlusions or large changes in the field of view, ICPIF may be used in conjunction with thresholding methods [29] or statistical sampling methods [19] just as is done with ICP.

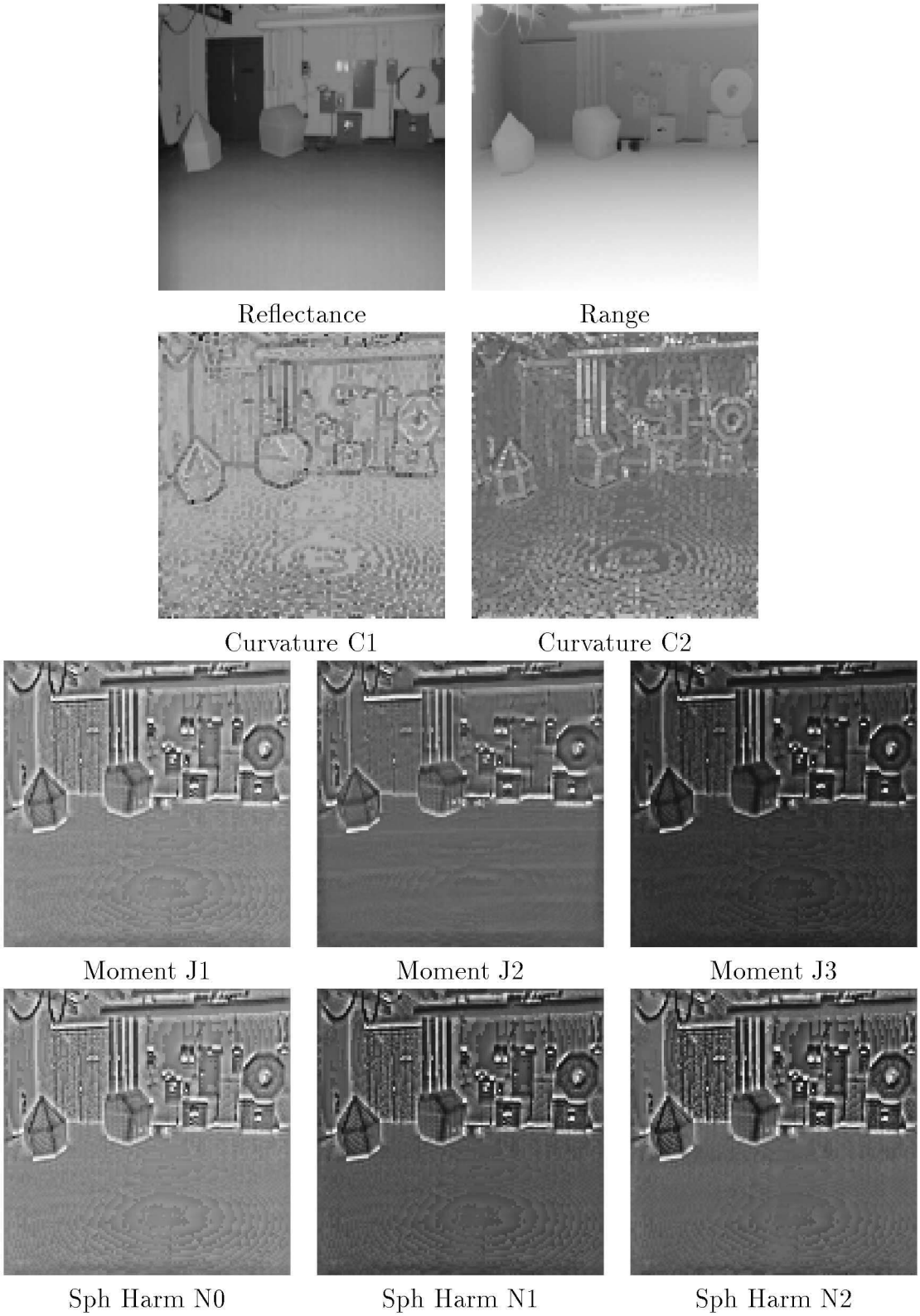


Fig. 2. A range image and eight different invariants.

## 4 Analysis of the ICPIF Metric

可否将IF作为一个soft weight?

While it is difficult to make definitive statements about the ICP registration over all possible real world scenes, we may gain some insight about how features influence the registration. In this section, ICPIF is compared with traditional ICP under ideal, noise-free conditions to demonstrate: 1) scene points will be matched with their correct correspondences for a larger set of trasformations, 2) incorrect scene alignments at the global minima will exist for a smaller set of transformations, and 3) monotonic convergence to a global minima is preserved.

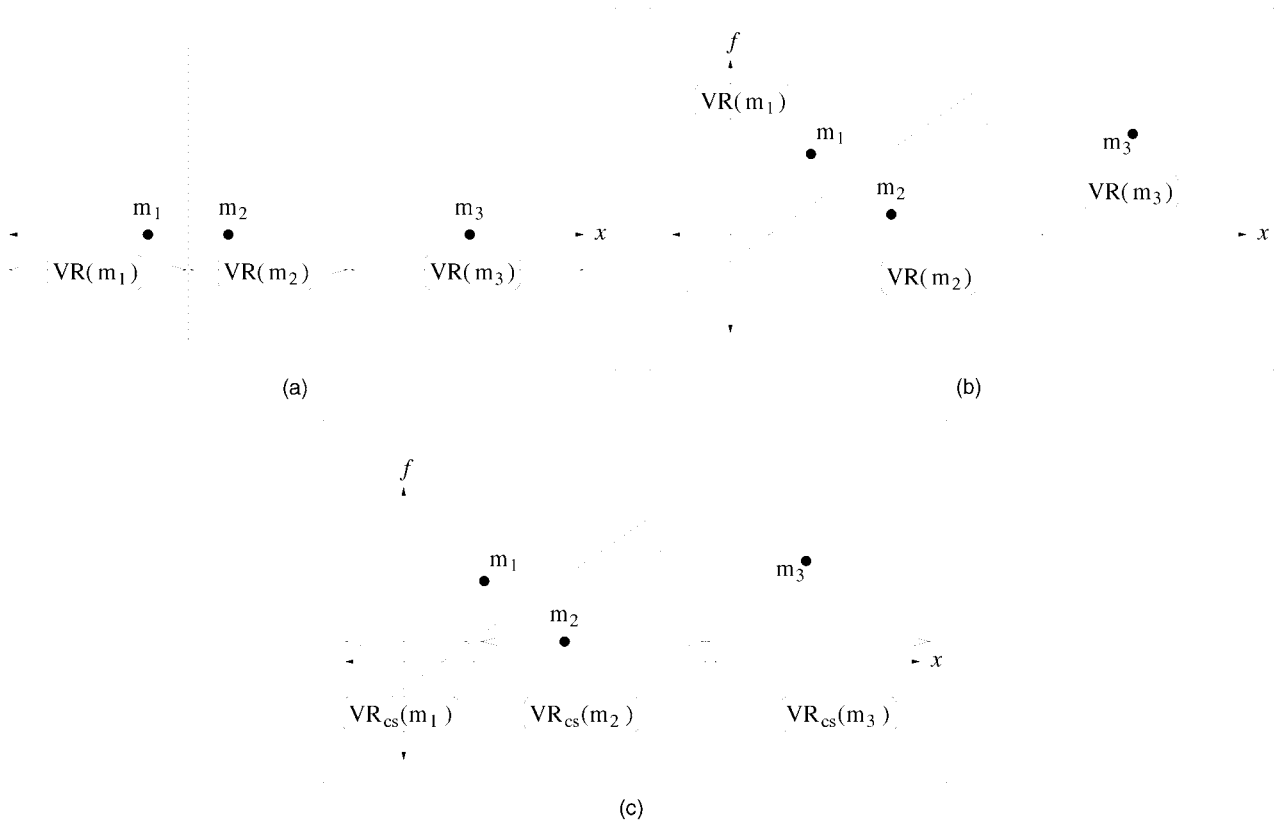


Fig. 3. (a) The Voronoi diagram of a model in positional space only. (b) The Voronoi diagram of a model in positional $\times$ invariant space. (c) The Voronoi diagram cross-section in positional space seen by point $\mathbf{s}$, where $\mathbf{s}_f = (\mathbf{m}_2)_f$.

### 4.1 Voronoi Analysis

Proximity problems such as the nearest-neighbor search for correspondences of ICP are traditionally analyzed with the aid of a Voronoi diagram [21]. Given a pointset $\mathcal{P}$ containing $N$ points in $\mathbb{R}^d$, we may subdivide $\mathbb{R}^d$ into $N$ Voronoi regions, one for each point $\mathbf{p}$, where the Voronoi region $VR(\mathbf{p})$ is defined to be the locus points in $\mathbb{R}^d$ that are closer to $\mathbf{p}$ than they are to any other point in $\mathcal{P}$. Let us assume, for the moment, that the model is a discrete set of points. In traditional ICP, $\mathbb{R}^3$ is subdivided into Voronoi regions of the model $\mathcal{M}$ and a query of the Voronoi diagram is performed each time we try to match one of the scene points with one of the model points. Because the scene is transformed after each iteration, the scene points are rather free to move around within the diagram, subject to the rigidity of the scene. For ICPIF, $\mathbb{R}^{3+k}$ is subdivided by the model $\mathcal{M}$ and again a query of the Voronoi diagram is performed to match each scene point with a model point. Again, the scene points are rather free to move about in the $\mathbb{R}^3$ subspace of positions, but their feature coordinates are fixed.

To make this idea more concrete, suppose that the scene and the model have one positional dimension ($x$) and one feature dimension ($f$). Fig. 3a shows the one dimensional Voronoi diagram of the model with respect to the positional dimension $x$, where the Voronoi region associated with a point $\mathbf{m}$ is denoted $VR(\mathbf{m})$. In traditional ICP, the scene point $\mathbf{s}$ is free to move about within the positional space $\mathbf{x}$ and a correspondence is made between $\mathbf{s}$ and $\mathbf{m}$ when $\mathbf{s}$ is positioned within $VR(\mathbf{m})$. Fig. 3b shows the two dimensional Voronoi diagram with respect to both $x$ and $f$. In ICPIF, a scene point $\mathbf{s}$ is free to move about in the $x$ direction, but is constrained to lie on the line $f = \mathbf{s}_f$. We may construct a new diagram representing the Voronoi diagram as encountered by point $\mathbf{s}$. An example of this concept is illustrated in Fig. 3c, where we construct the Voronoi diagram as seen by a scene point $\mathbf{s}$ for which $\mathbf{s}$ and $\mathbf{m}_2$ have the same feature value. Visual inspection suggests that the Voronoi region $VR_{cs}(\mathbf{s}, \mathbf{m}_2)$ for which $\mathbf{s}$ matches with $\mathbf{m}_2$ in Fig. 3c is larger than the ordinary Voronoi region $VR(\mathbf{m}_2)$ of Fig. 3a.

This simplified analysis is intended to provide insight into why correct matches become more likely when the invariant values of the scene and model points are similar, a concept made concrete in the next section.

### 4.2 Closest Point Selection

Let us assume noise free data and, that, invariant values can be computed exactly for each point in both the scene and in the model. We define a *ground truth correspondence* as a correspondence between a scene point $\mathbf{s}_i$ and a model point $\mathbf{m}_i$ where $\mathbf{s}_i$ and $\mathbf{m}_i$ represent the same point in the real world. In Proposition 1, we show that if a ground truth correspondence exists between $\mathbf{s}_i$ and $\mathbf{m}_i$, then for any scene transformation where $\mathbf{s}_i$ is paired up with $\mathbf{m}_i$ under $\mathrm{CP}(\mathbf{s}_i, \mathcal{M})$, $\mathbf{s}_i$ will be paired up with $\mathbf{m}_i$ under $\mathrm{CP}_\alpha(\mathbf{s}_i, \mathcal{M})$ also.

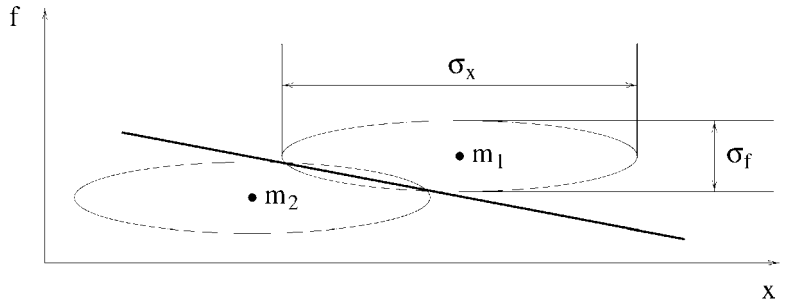


Fig. 4. Multivariate Gaussian distributions with different means but identical covariance induce a minimum-Mahalanobis-distance decision boundry. When the feature dimension $f$ is uncorrelated with respect to the position dimension $x$, scaling $f$ by $\alpha = \frac{\sigma_x}{\sigma_f}$ converts the minumum-Mahalanobis-distance decision into a nearest-neighbor decision.

**Proposition 1.** *For a noise-free scene $\mathcal{S}$ and model $\mathcal{M}$, if a scene point $\mathbf{s}_i$ has a ground truth correspondence with a model point $\mathbf{m}_i$, then $\mathbf{m}_i = \mathrm{CP}(\mathbf{s}_i, \mathcal{M})$ implies $\mathbf{m}_i = \mathrm{CP}_\alpha(\mathbf{s}_i, \mathcal{M})$.*

**Proof.** Because the two points are ground truth correspondences, $\alpha^2 d_f(\mathbf{s}_i, \mathbf{m}_i) = 0$. Furthermore, for any $\mathbf{m}$ in $\mathcal{M}$,

$$d_e(\mathbf{s}_i, \mathbf{m}_i) \leq d_e(\mathbf{s}_i, \mathbf{m}),$$

and, therefore,

$$d_\alpha(\mathbf{s}_i, \mathbf{m}_i) = d_e(\mathbf{s}_i, \mathbf{m}_i) \leq d_e(\mathbf{s}_i, \mathbf{m}) \leq d_\alpha(\mathbf{s}_i, \mathbf{m}).$$

□

**Corollary 1.** *All transformations that form ground truth correspondences for an entire scene using* CP *will do so using* $\mathrm{CP}_\alpha$.

When the model is a discrete set of points, it is no longer required that the invariant values be exact, only that the errors in invariant values be sufficiently small. The exact amount of error that is allowed depends upon the relative positions and invariant values of the data.

### 4.3 Global Minima

There is no guarantee that the global minima of the cost function is unique. However, the use of the $d_\alpha$ metric can only decrease the set of false minima in the noise-free case with perfect correspondences. Let us define an *absolute minimization* to be a scene transformation such that $d(\mathbf{s}_i, \mathbf{m}_i) = 0$ for all scene points and a *false minimization* to be an absolute minimization such that at least one correspondence is not at ground truth.

**Proposition 2.** *The set of transformations forming false minimizations under $d_\alpha$ is a subset of the transformations forming false minimizations under $d_e$.*

**Proof.** Since $d_\alpha(\mathbf{s}_i, \mathbf{m}_i) = 0$ implies $d_e(\mathbf{s}_i, \mathbf{m}_i) = 0$, all absolute minimizations under $d_\alpha$ are absolute minimizations under $d_e$. However, there is only one transformation that forms a ground truth correspondence, which does so under both metrics. □

## 5 Choosing Feature Weights

The correspondence problem may be viewed as a pattern classification problem, where each model point defines a distinct class and each scene point defines a query vector. The nearest-neighbor selection rule used by ICPIF is an optimal minimum error-rate classifier when the model points are described by multivariate Gaussian distributions with different means, but identical covariances of the form $\mathbf{\Sigma} = \sigma\mathbf{I}$. When the covariance matrix is of a more general form, the optimal classifier is the minimum-Mahalanobis-distance classifier [10], graphically depicted in Fig. 4. In ICPIF, we wish to scale the feature values to provide some trade-off between our trust in the feature and positional information. By accepting a Gaussian noise model, we can use the covariance matrix to trade-off between feature error caused by sensor noise and positional errors caused by the misalignment. The following sections describe a method for estimating the covariance matrix from the positional and feature data in the scene.

### 5.1 Errors in Feature Values

Errors in the invariant feature values appear to be well approximated by a normal distribution. To confirm this, we examine the invariant shape feature values within a large planar patch. Fig. 5 shows the 1D histograms and 2D scatter plots of moment feature values within a planar surface patch of the range image in Fig. 2. Since these distributions are unimodal and nearly symmetric, we conclude that the multivariate normal distribution is a reasonable model. The covariance matrix $\Sigma_f$ is estimated directly from the feature values of the planar region and the vector of raw features can be transformed by $\Sigma_f^{-1/2}$ into a vector of uncorrelated invariant features with unit variance.

### 5.2 Errors in Positional Values

The distribution of positional errors is largely due to an unknown amount of misalignment of the scene with respect to the model, which makes estimation of positional error difficult. It is not well modeled as a normal distribution. However, we accept the Gaussian noise model because the covariance can be used to rescale the feature values in a principled manner through the minimum-Mahalanobis-distance classifier.

Let $d_{GT}$ be the ground truth distance between the scene point and its matching model point and, let $d_{GTx}$, $d_{GTy}$, and $d_{GTz}$ be the distances in the $x$, $y$, and $z$ directions. We shall

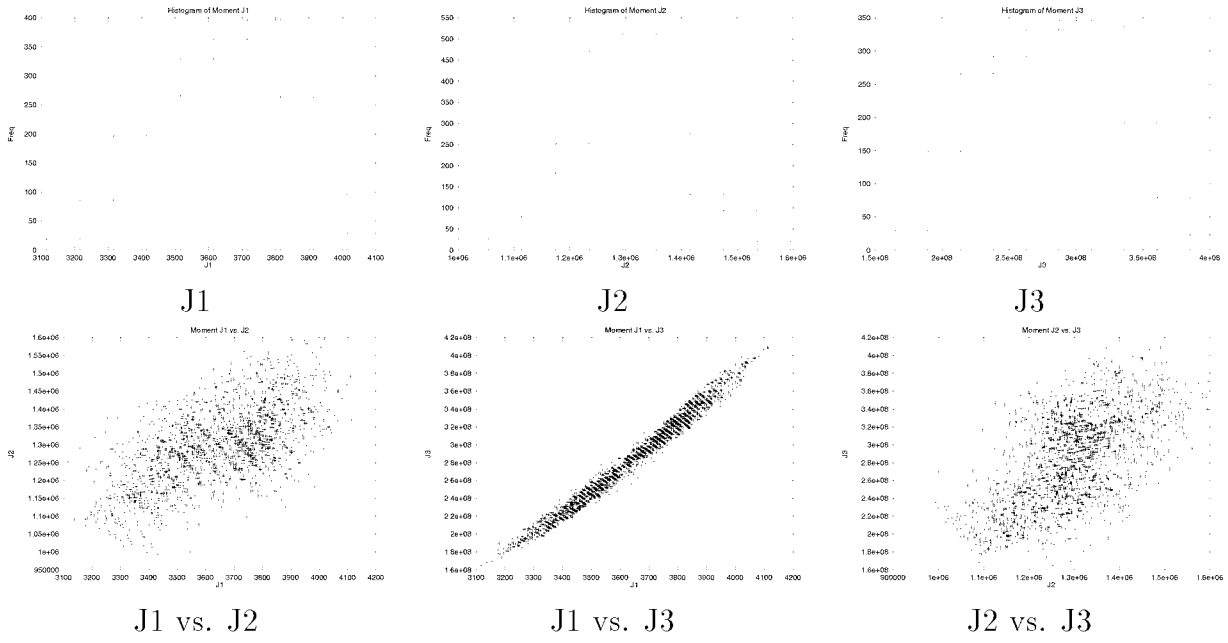


Fig. 5. Histograms and scatter plots of the moment features values within a flat surface of the range image shown in Fig. 2.

assume that misalignment error is independent of any feature noise and we shall further assume that the variances in each of the three positional dimensions are identical and independent. By linearity and independence,

$$E[d_{GT}^2] = E[d_{GTx}^2] + E[d_{GTy}^2] + E[d_{GTz}^2],$$

and since each dimension is identical,

$$E[d_{GTx}^2] = E[d_{GTy}^2] = E[d_{GTz}^2] = \frac{1}{3}E[d_{GT}^2].$$

Therefore, estimating the covariance matrix of the positional error reduces to estimating $\sigma_x^2 = E[d_{GTx}^2]$.

We will use the distance to the closest point on the model, $d_{CP}$, to estimate the distance to the ground truth correspondence $d_{GT}$ (see Fig. 6). If we suppose the model is locally planar near the ground truth correspondence point and if we further suppose that the scene point is located with uniform distribution on the surface of a sphere $S$ with radius $d_{GT}$, we find that

$$\begin{aligned} E[d_{CP}^2|d_{GT}] &= \frac{1}{4\pi d_{GT}^2}\int_S (d_{GT}\cos\phi)^2 d_{GT}^2 \sin\phi dS \\ &= \frac{1}{3}d_{GT}^2. \end{aligned}$$

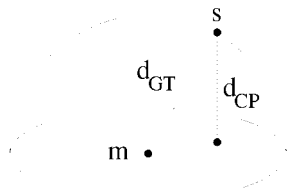


Fig. 6. The (unknown) positional error $d_{GT}$ is estimated using the (known) distance to the closest point $d_{CP}$.

And since

$$E[d_{GT}^2] = 3E[E[d_{CP}^2|d_{GT}]] = 3E[d_{CP}^2] = 3d_{CP}^2,$$

the estimate of error variance due to misalignment in the $x$ dimension is $\sigma_x^2 \approx d_{CP}^2$.

### 5.3 Determination of $\alpha$ and Implementation Details

Based on our estimates for $\sigma_x^2$ and $\Sigma_f$, we are now able make the proper choice for the parameter $\alpha$. Multiplying the feature values by $\Sigma_f^{-1/2}$ will normalize the variance of the feature values to 1. A further multiplication of the feature values by $\sigma_x^2$ will set the variance of the feature error equal to the variance of the positional error. Hence, the desired scale factor $\alpha$ after normalization by $\Sigma_f^{-1/2}$ is

$$\alpha^2 = \sigma_x^2 \approx d_{CP}^2 \approx MSE, \tag{2}$$

where $MSE$ is the mean squared distance from a scene point to its closest model point, a global estimate of $d_{CP}^2$. Note that this estimate of $\alpha$ changes after each iteration.

In Section 5.5, we will show that ICPIF converges monotonically to local minimum of the cost function when the feature weight $\alpha$ is constant or decreasing. Convergence is not guaranteed when $\alpha$ is allowed to increase. If the monotonic convergence property is used for detecting when the algorithm has converged, as is done by tracking the difference in MSE error between iterations, $\alpha$ should be restricted to be monotonically nonincreasing.

Further, it is not likely that $\alpha$ will converge to zero unless the scene is a perfect match for a subset of the model. In this case, the final answer given by ICPIF is not the optimal mean squared error in pointwise distance. Since we believe that the mean squared error in pointwise distance is the proper error metric for rigid registration, we recommend that $\alpha$ be forced to go to zero after ICPIF has converged. Failure to do so may yield a registration that has a higher $MSE$.

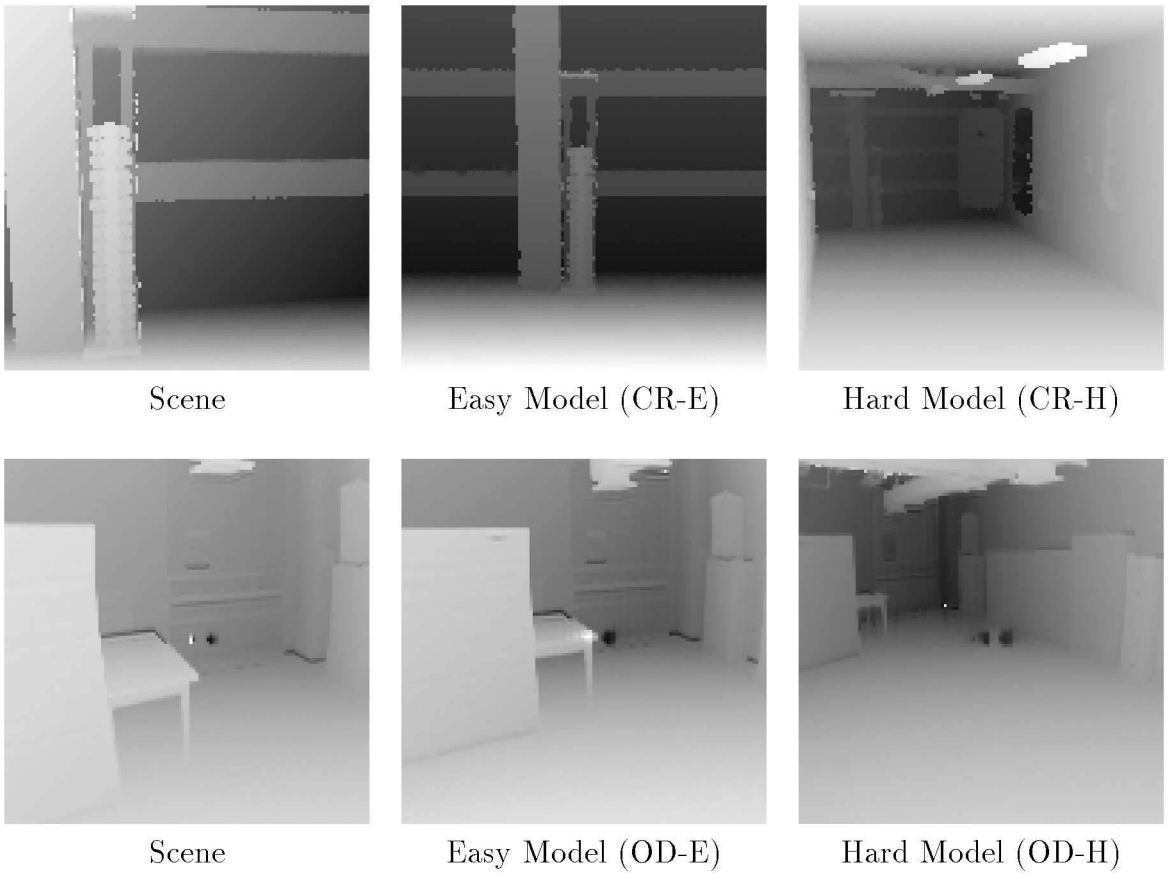


Fig. 7. Cruiser (top) and Odetics (bottom) test sets.

Finally, there are pitfalls in using the popular k-d tree [1] for performing nearest-neighbor search when $\alpha$ is scaled. The cells of a k-d tree are built by recursively splitting the dimension with the largest absolute spread distance, thereby reducing the required search radius for queries in that cell. If a dimension is scaled, its spread distance changes and so does the best split dimension. However, the k-d tree is a static structure. Two methods that may be used to counteract this effect: the tree may be rebuilt at a penalty of $O(n \log n)$, or the search at each ply of the tree must be adjusted to a wider radius, which is less efficient. When the rebuilding option is chosen, ICPIF has an average case complexity of $O(n \log n)$ per iteration, but always rebuilding the tree may be wasteful for small changes in $\alpha$. As a compromise, the tree may be rebuilt only after a sufficient decrease in $\alpha$, such as 10 percent.

### 5.4 ICPIF Algorithm

We now summarize the final version of the ICPIF algorithm.

**Algorithm 3 (Iterative Closest Point Registration Using Invariant Features, Final Algorithm).** Let $\mathcal{S}$ be a set of $N_s$ points, $\{\mathbf{s}_1, \cdots, \mathbf{s}_{N_s}\}$, and let $\mathcal{M}$ be the model.

1. Estimate the feature covariance and decorrelate features.
2. Let $\mathbf{T}_0$ be an initial estimate of the transformation.
3. $\alpha_0 = \sqrt{MSE}$.
4. Repeat for $k = 1..k_{\max}$ or until termination criteria is met
   a. Build up the set of correspondences
   $$\mathcal{C} = \bigcup_{\mathbf{s}_i \in \mathcal{S}} \{(\mathbf{T}_{k-1}(\mathbf{s}_i), \mathrm{CP}_{\alpha_{k-1}}(\mathbf{T}_{k-1}(\mathbf{s}_i), \mathcal{M}))\}.$$
   b. Compute the new transformation $\mathbf{T}_k$ that minimizes mean square error between point pairs in $\mathcal{C}$.
   c. $\alpha_k = \min(\alpha_{k-1}, \sqrt{MSE})$.
   d. Rebuild k-d tree if desired.
5. If $\alpha_k \neq 0$, assign $\alpha_0 = 0$ and goto 4.

### 5.5 Convergence

In [23], the ICPIF Algorithm 2 of Section 3.2 was shown to converge to a local minimum for a fixed value of $\alpha$. Here, we extend this result to the nonincreasing values of $\alpha$ of ICPIF Algorithm 3.

**Proposition 3.** *Iterative closest point registration using the distance function $d_\alpha(\mathbf{s}, \mathbf{m})$ with monotonically nonincreasing values of $\alpha$ will always converge monotonically to a local minimum.*

**Proof.** The convergence proof of Besl and McKay is generalized. For any given iteration $k$, let $T_k$ be our current estimate of the best scene transformation, let $\mathbf{s}_{i,k} = T_k(\mathbf{s}_i)$ be the current location of the scene point $\mathbf{s}_i$, let $\alpha_k$ be the current value of $\alpha$, and let $\mathbf{m}_{i,k} = \mathrm{CP}_{\alpha_k}(T_k(\mathbf{s}_i), \mathcal{M})$ be the nearest-neighbor correspondence of $\mathbf{s}_{i,k}$. We start with iteration $k-1$ and find the mean squared error over all correspondences:

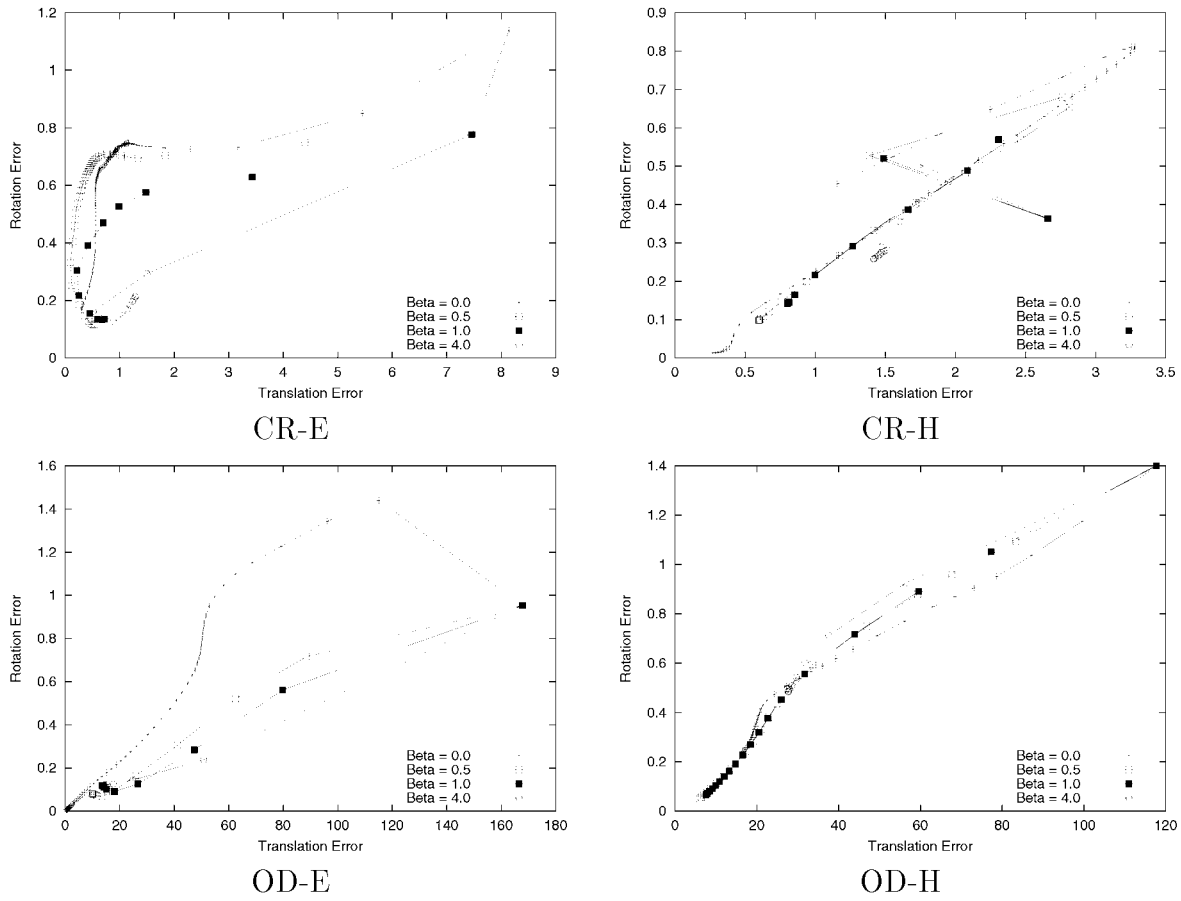


Fig. 8. Error from ground truth measured against each iteration for ICPIF using moment invariants.

$$e_{k-1} = \frac{1}{N_s}\sum_{i=1}^{N_s}[d_e(\mathbf{s}_{i,k-1}, \mathbf{m}_{i,k-1}) + \alpha_{k-1}^2 d_f(\mathbf{s}_{i,k-1}, \mathbf{m}_{i,k-1})].$$

After applying transformation $T_k$ to the scene, the error becomes:

$$e_{k-1} = \frac{1}{N_s}\sum_{i=1}^{N_s}[d_e(\mathbf{s}_{i,k}, \mathbf{m}_{i,k-1}) + \alpha_{k-1}^2 d_f(\mathbf{s}_{i,k}, \mathbf{m}_{i,k-1})].$$

Because $T_k$ minimizes the positional error between correspondences,

$$\sum_{i=1}^{N_s} d_e(\mathbf{s}_{i,k}, \mathbf{m}_{i,k-1}) \leq \sum_{i=1}^{N_s} d_e(\mathbf{s}_{i,k-1}, \mathbf{m}_{i,k-1}).$$

Because invariant values do not change as a result of rigid body transformation,

$$\sum_{i=1}^{N_s} \alpha_{k-1}^2 d_f(\mathbf{s}_{i,k}, \mathbf{m}_{i,k-1}) = \sum_{i=1}^{N_s} \alpha_{k-1}^2 d_f(\mathbf{s}_{i,k-1}, \mathbf{m}_{i,k-1}).$$

Therefore, $\tilde{e}_{k-1} \leq e_{k-1}$.

Next, the feature weights are updated to $\alpha_k$ and new correspondences are calculated using $\mathrm{CP}_{\alpha_k}$. The closest point function guarantees that

$$d_{\alpha_k}(\mathbf{s}_{i,k}, \mathbf{m}_{i,k}) \leq d_{\alpha_k}(\mathbf{s}_{i,k}, \mathbf{m}_{i,k-1})$$

for each point $s_i$ in $\mathcal{S}$ and since $\alpha$ is nonincreasing, $d_{\alpha_k}(\mathbf{s}_{i,k}, \mathbf{m}_{i,k-1}) \leq d_{\alpha_{k-1}}(\mathbf{s}_{i,k}, \mathbf{m}_{i,k-1})$. Therefore, $e_k \leq \tilde{e}_{k-1}$.□

## 6 Experimental Results

### 6.1 Methodology

To test the ICPIF algorithm, we have run experiments on the data sets shown in Fig. 7. The Cruiser data set is a synthetic data set constructed using the Z buffer output of the Radiance rendering software and the Odetics data set contains range images taken with an Odetics laser range finder atop a mobile platform. For the Odetics images, distortion and scaling have been corrected using the default sensor calibration parameters [16], but no filtering or noise reduction was performed. From each of these data sets we have selected both an "easy" test case, where the scene is registered to a model that is only slightly larger field of view than the scene and a "hard" test case, where the scene is registered to a model that captures a considerably larger field of view than the scene. We denote the easy Cruiser test set as "CR-E," the easy Odetics test set as "OD-E," the hard Cruiser test set as "CR-H," and the hard Odetics test set as "OD-H." Three moment invariants, three spherical harmonics invariants and two curvature invariants were computed using the method described in Section 2 at each point in all images. A $9 \times 9 \times 9$ uniform Cartesian sampling of the

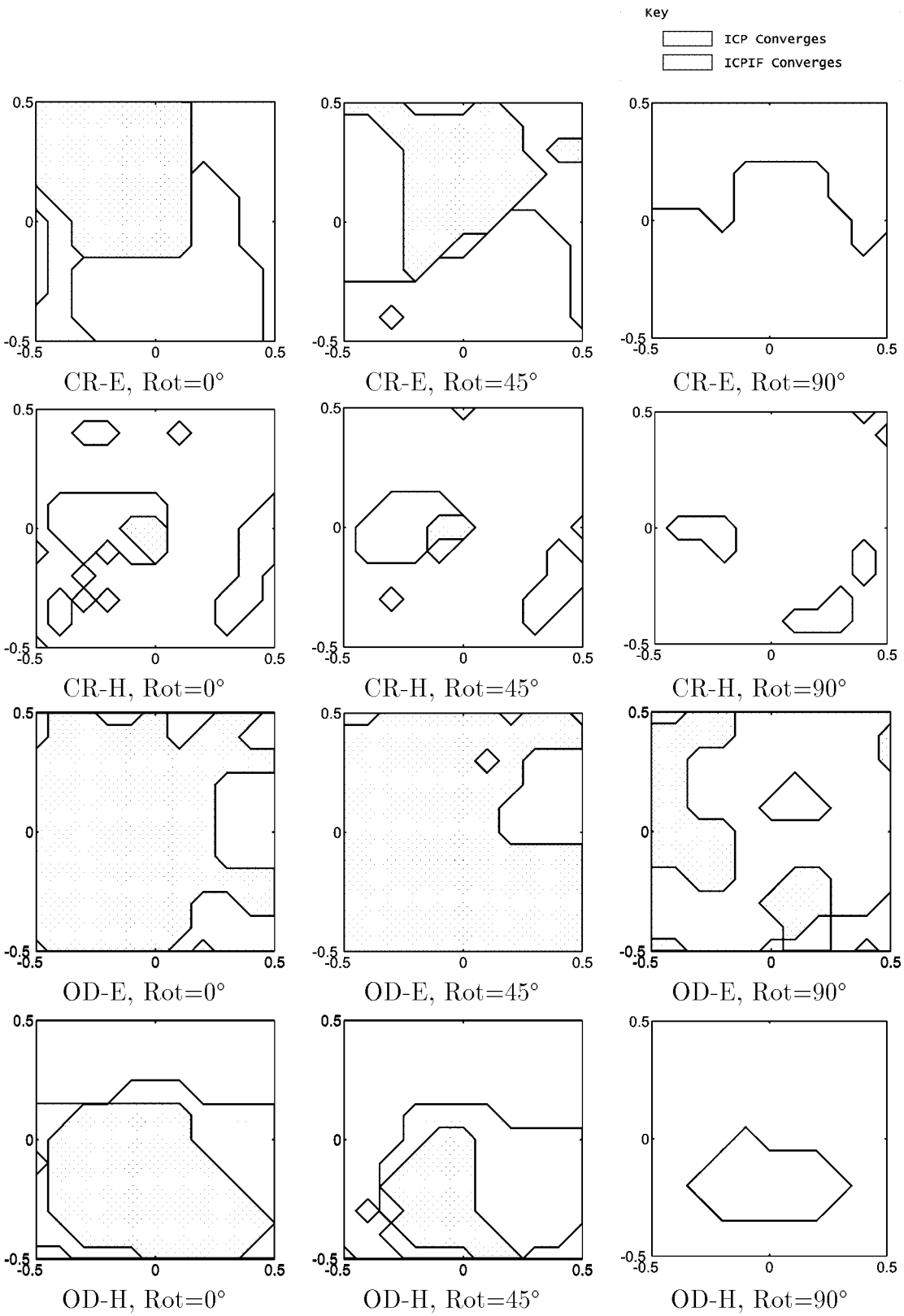


Fig. 9. Convergence regions of ICP and ICPIF using moment invariants with respect to $(x, y)$ translation of the initial condition.

space centered at the image point was used to perform the integration described in Section 2.24 and the size of the enclosing sphere was set to approximately 2 percent of the size of the easy model for each set.

The ICPIF algorithm was run on these data sets using both a fixed value of $\alpha$ as described in Algorithm 2 and, also, using adaptive weighting as described in Algorithm 3. For the adaptive weighting experiments, we introduced the

TABLE 1
Size of Convergence Region as Percentage of Random Initial Conditions that Converge to Ground Truth

| Weight | Cruiser (easy) | | | Cruiser (hard) | | | Odetics (easy) | | | Odetics (hard) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\alpha$ mult | M | S | C | M | S | C | M | S | C | M | S | C |
| 0 | 0.05 | 0.05 | 0.05 | 0.02 | 0.02 | 0.02 | 0.13 | 0.13 | 0.13 | 0.04 | 0.04 | 0.04 |
| 1 | 0.05 | 0.05 | 0.05 | 0.02 | 0.03 | 0.02 | 0.13 | 0.13 | 0.13 | 0.04 | 0.04 | 0.04 |
| 3 | 0.04 | 0.04 | 0.05 | 0.03 | 0.03 | 0.03 | 0.14 | 0.13 | 0.13 | 0.02 | 0.02 | 0.04 |
| 10 | 0.06 | 0.07 | 0.05 | 0.02 | 0.03 | 0.03 | 0.11 | 0.11 | 0.13 | 0.02 | 0.02 | 0.02 |
| 30 | 0.19 | 0.15 | 0.08 | 0.02 | 0.08 | 0.03 | 0.15 | 0.11 | 0.13 | 0.05 | 0.05 | 0.02 |
| 100 | 0.21 | 0.24 | 0.20 | 0.02 | 0.14 | 0.00 | 0.18 | 0.16 | 0.26 | 0.05 | 0.08 | 0.06 |
| 300 | 0.26 | 0.38 | 0.18 | 0.05 | 0.27 | 0.00 | 0.21 | 0.17 | 0.41 | 0.09 | 0.11 | 0.00 |
| 1,000 | 0.32 | 0.52 | 0.27 | 0.39 | 0.09 | 0.00 | 0.39 | 0.31 | 0.46 | 0.13 | 0.26 | 0.00 |
| 3,000 | 0.35 | 0.63 | 0.31 | 0.00 | 0.00 | 0.00 | 0.53 | 0.62 | 0.48 | 0.04 | 0.57 | 0.00 |
| 10,000 | 0.44 | 0.66 | 0.45 | 0.00 | 0.00 | 0.00 | 0.90 | 0.89 | 0.59 | 0.00 | 1.00 | 0.03 |
| 30,000 | 0.00 | 0.68 | 0.47 | 0.00 | 0.00 | 0.00 | 1.00 | 1.00 | 0.81 | 0.00 | 1.00 | 0.00 |
| 100,000 | 0.00 | 1.00 | 0.46 | 0.00 | 0.00 | 0.00 | 1.00 | 1.00 | 1.00 | 0.00 | 1.00 | 0.00 |
| 300,000 | 0.00 | 1.00 | 0.65 | 0.00 | 0.00 | 0.00 | 1.00 | 1.00 | 1.00 | 0.00 | 1.00 | 0.00 |
| 1,000,000 | 0.00 | 1.00 | 1.00 | 0.00 | 0.00 | 0.00 | 1.00 | 1.00 | 1.00 | 0.00 | 1.00 | 0.00 |
| $\beta$ mult | M | S | C | M | S | C | M | S | C | M | S | C |
| 0 | 0.05 | 0.05 | 0.05 | 0.02 | 0.02 | 0.02 | 0.13 | 0.13 | 0.13 | 0.04 | 0.04 | 0.04 |
| 0.1 | 0.08 | 0.33 | 0.18 | 0.03 | 0.21 | 0.00 | 0.16 | 0.16 | 0.24 | 0.09 | 0.10 | 0.05 |
| 0.2 | 0.15 | 0.36 | 0.24 | 0.05 | 0.27 | 0.00 | 0.19 | 0.20 | 0.34 | 0.08 | 0.14 | 0.03 |
| 0.3 | 0.16 | 0.45 | 0.25 | 0.03 | 0.27 | 0.00 | 0.24 | 0.28 | 0.46 | 0.09 | 0.15 | 0.02 |
| 0.5 | 0.24 | 0.50 | 0.29 | 0.05 | 0.18 | 0.00 | 0.31 | 0.38 | 0.46 | 0.11 | 0.27 | 0.01 |
| 0.7 | 0.25 | 0.54 | 0.30 | 0.17 | 0.08 | 0.00 | 0.37 | 0.49 | 0.49 | 0.14 | 0.44 | 0.00 |
| 1 | 0.28 | 0.53 | 0.34 | 0.21 | 0.06 | 0.00 | 0.44 | 0.59 | 0.51 | 0.27 | 0.58 | 0.00 |
| 2 | 0.31 | 0.61 | 0.39 | 0.04 | 0.01 | 0.00 | 0.56 | 0.76 | 0.51 | 0.13 | 0.78 | 0.00 |
| 3 | 0.39 | 0.65 | 0.42 | 0.04 | 0.00 | 0.00 | 0.60 | 0.84 | 0.50 | 0.09 | 0.92 | 0.00 |
| 5 | 0.49 | 0.68 | 0.46 | 0.04 | 0.00 | 0.00 | 0.76 | 0.91 | 0.57 | 0.00 | 0.97 | 0.00 |
| 10 | 0.41 | 0.73 | 0.47 | 0.01 | 0.00 | 0.00 | 0.95 | 0.98 | 0.64 | 0.00 | 1.00 | 0.06 |
| 20 | 0.14 | 0.86 | 0.48 | 0.00 | 0.00 | 0.00 | 1.00 | 1.00 | 0.80 | 0.00 | 1.00 | 0.05 |
| 30 | 0.03 | 0.95 | 0.49 | 0.00 | 0.00 | 0.00 | 1.00 | 1.00 | 0.94 | 0.00 | 1.00 | 0.02 |
| 50 | 0.00 | 1.00 | 0.54 | 0.00 | 0.00 | 0.00 | 1.00 | 1.00 | 0.98 | 0.00 | 1.00 | 0.03 |
| 100 | 0.00 | 1.00 | 0.68 | 0.00 | 0.00 | 0.00 | 1.00 | 1.00 | 1.00 | 0.00 | 1.00 | 0.00 |

*Features are based on moments (M), spherical harmonics (S), and curvature (C).*

additional parameter $\beta$ to increase or decrease overall feature weight contributions, with weights assigned according to the schedule $\alpha_k = \min(\alpha_{k-1}, \beta\sqrt{MSE})$ at Step 4c of Algorithm 3. Therefore, an $\alpha$ or $\beta$ value of 0 behaves exactly as the original ICP algorithm and a $\beta$ multiplier of 1 behaves exactly as the final ICPIF with default adaptive weight scheduling. Depending upon the initial estimate and geometry, a $\beta$ multiplier of 1 corresponds to an initial $\alpha$ value somewhere between about 100 and 1,000, decreasing as the alignment gets better.

### 6.2 Iterative Behavior

Fig. 8 shows the iterative behavior of the ICPIF algorithm using the moment invariants. Each curve displays iterative behavior for a single value of $\beta$, while each point on the curve displays the rotation and translation error for a single iteration. For nonzero weights, only the portion of the algorithm where features were used are shown; the final fit using ICP without features is not shown. Translation error, shown on the X axis, is measured as the real world distance between the ground truth camera location and computed camera location. Rotation error, shown on the Y axis, is measured as the angle in radians between the ground truth camera orientation and computed camera orientation. When the scene is not initially well aligned, ICP chooses incorrect correspondences, resulting in slow convergence. In contrast, the error declines rapidly when invariants are used and higher weights converge more quickly. However, the final registration is somewhat worse, suggesting that features are more useful for coarse registration than fine registration.

### 6.3 Global Convergence

Next, we examine the size of the convergence region. For a given initial condition, we say that the algorithm converges if the final rotation and translation error from a known approximate ground truth is under a certain threshold. For ICPIF, we include the final fit using ICP without features in deciding convergence. The plots in Fig. 9 show the size of the convergence regions for ICP and ICPIF for different $(x, y)$ translations of the initial conditions, with the $x$ axis being translation in the $x$ direction and the $y$ axis being translation in the $y$ direction. The $(x, y)$ dimensions are normalized by the size of the model and the ground truth location is set to $(0, 0)$. There is no translation in the $z$ direction. Three different rotations about the $z$ axis were used: $0°$, $45°$, and $90°$. ICPIF was run using moment

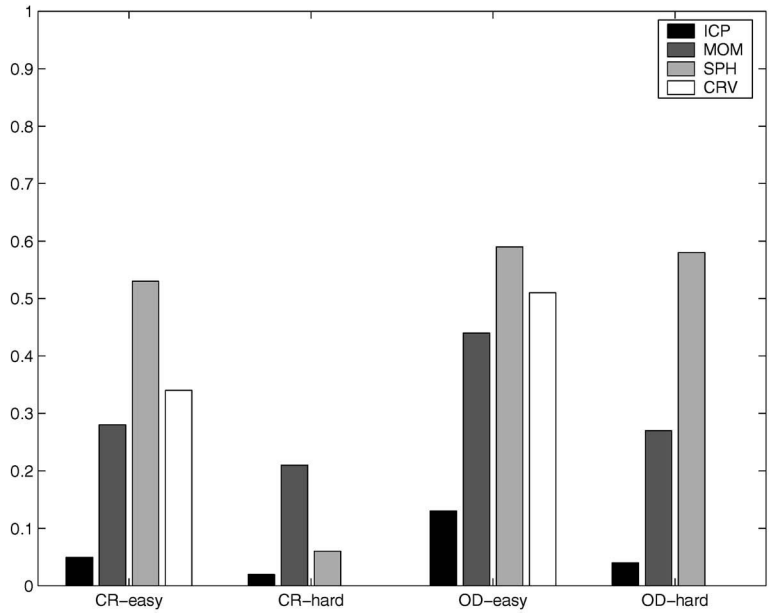


Fig. 10. Size of convergence region as percentage of random initial conditions that converge to ground truth using default weights ($\beta = 1$) and different feature types.

invariants and adaptive weight scheduling ($\beta = 1$). It can be seen from these experiments that ICPIF seems to provide a larger region of convergence. However, there are some initial conditions for which using features does not converge even when ICP does converge.

To better estimate the total size of the convergence region, ICP and ICPIF were run on 100 different random initial transformations. Random rotations were chosen as random quaternions on the unit 4-sphere, while random translations were chosen to place the center of mass of the scene within the bounding box of the model. Convergence was defined as above, using a threshold on the distance from the known ground truth and the same 100 initial transformations were used for each feature type and each feature weight value. The results of this experiment on different feature weights and feature types is shown in Table 1. At the highest feature weights, the algorithm either converges for all initial conditions, or none of the initial conditions, depending on the feature type and scene geometry. Further, we notice a relationship between convergence rate and model size, with the smaller "easy" models more likely to converge at high feature weights than the larger "hard" models. This suggests that the penalty for making an incorrect correspondence is greater with the larger models. However, it is also possible that within a smaller model the features are more geometrically distinct and are more spatially clustered, resulting in fewer correspondence errors. Therefore, we conclude that it may be possible to obtain better performance at higher feature weights for smaller models, while $\alpha = \sqrt{MSE}$ seems to work well for large models which induce many local minima.

A summary of the convergence region size of ICP and ICPIF ($\beta = 1$) for different feature types for this experiment is shown in Fig. 10. The spherical harmonics invariants appear to perform best, followed by moment invariants and finally curvature. Incidentally, the spherical harmonics invariants seem to be the most difficult to compute; our implementation computes spherical harmonics invariants for a 128x128 image in 32 seconds, moment invariants in nine seconds, and curvature in three seconds.

## 7 Summary

This paper presents ICPIF, a fully automatic range image registration method that uses shape features in conjunction with point positions to register range images without the need for a user supplied initial estimate. Theoretical results show that under noise-free conditions, ICPIF chooses the correct pointwise correspondences at least as well as ICP, and that monotonic convergence to a local minimum is preserved. Convergence to the ground truth registration occurs more often and in fewer iterations than traditional ICP. The relative weights of the feature and positional components can be controlled by trading off error in feature values caused by noise against error in positions caused by misalignment. This is accomplished using a calibration-time estimation of feature noise and fully automatic run-time estimation of misalignment. Experimental results on real and synthetic images suggest that for some alignment problems, matching can be performed using features alone, while for larger alignment problems, a blend of position and features may be better.

## Acknowledgments

The authors wish to thank Jerome Obermark for his many valuable contributions. This material is based upon work supported by Department of Energy (DOE) under Award No. DE-FG04-86NE37969. The Cruiser data set was created by Saba Rofchaei and Greg Ward under contract from the US Navy. Odetics laser range finder images are courtesy the CESAR lab at Oak Ridge National Laboratory in Tennessee, made available through the Web server of the University of South Florida [20].

## REFERENCES


[1] J.L. Bentley, "K-D Trees for Semidynamic Point Sets," *Proc. Sixth Ann. Symp. Computational Geometry,* pp. 187-197, 1990.

[2] R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau, "Towards a General Multi-View Registration Technique," *IEEE Trans. Pattern Analysis and Machine Intelligence* vol. 18, no. 5, pp. 540-547, May 1996.

[3] P.J. Besl and R.C. Jain, "Invariant Surface Characteristics for 3D Object Recognition in Range Images," *Computer Vision Graphics and Image Processing.* vol. 33, no. 1, pp. 33-80, Jan. 1986.

[4] P.J. Besl and N.D. McKay, "A Method for Registration of 3-D Shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 14, no. 2, pp. 239-256, Feb. 1992.

[5] G. Burel and H. Henocq, "Three-Dimensional Invariants and Their Application to Object Recognition," *Signal Processing,* vol. 45, no. 1, pp. 1-22, July 1995.

[6] C.S. Chen, Y.P. Hung, and J.B. Cheng, "Ransac-Based Darces: A New Approach to Fast Automatic Registration of Partially Overlapping Range Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, no. 11, pp. 1229-1234, Nov. 1999.

[7] Y. Chen and G.G. Medioni, "Object Modeling by Registration of Multiple Range Images," *Image and Vision Computing,* vol. 10, no. 3, pp. 145-155, 1992.

[8] C. Chua and R. Jarvis, "3D Free Form Surface Registration and Object Recognition" *Int'l J. Computer Vision* vol. 17, pp. 77-99, 1996.

[9] C. Dorai, J. Weng, and A.K. Jain, "Optimal Registration of Object Views Using Range Data," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 10, pp. 1131-1138, Oct. 1997.

[10] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis.* John Wiley & Sons, 1973.

[11] D.W. Eggert, A.W. Fitzgibbon, and R.B. Fisher, "Simultaneous Registration of Multiple Range Views for Use in Reverse Engineering of Cad Models," *Computer Vision and Image Understanding,* vol. 69, no. 3, pp. 253-272, Mar. 1998.

[12] O.D. Faugeras and M. Hebert, "The Representation, Recognition, and Locating of 3-D Objects," *Int'l J. Robotics Research,* vol. 5, no. 3, pp. 27-52, 1986.

[13] J. Feldmar and N.J. Ayache, "Rigid, Affine and Locally Affine Registration of Free-Form Surfaces," *Int'l J. Computer Vision,* vol. 18, no. 2, pp. 99-119, May 1996.

[14] K. Higuchi, M. Hebert, and K. Ikeuchi, "Building 3-D Models from Unregistered Range Images," *Graphical Models and Image Processing,* vol. 57, no. 4, pp. 315-333, July 1995.

[15] B.K.P. Horn, "Closed Form Solutions of Absolute Orientation Using Unit Quaternions," *J. the Optical Soc. Am.-A,* vol. 4, no. 4, pp. 629-642. Apr. 1987.

[16] *3-D Laser Imaging System User's Guide.* Odetics Inc., 1990.

[17] A.E. Johnson, "Surface Landmark Selection and Matching in Natural Terrain," *Computer Vision and Pattern Recognition,* vol. 2. pp. 413-420, 2000.

[18] A.E. Johnson and M. Herbert, "Surface Matching for Object Recognition in Complex 3-Dimensional Scenes," *Image and Vision Computing,* vol. 16, no. 9/10, pp. 635-651, July 1998.

[19] T. Masuda and N. Yokoya, "A Robust Method for Registration and Segmentation of Multiple Range Images," *Computer Vision and Image Understanding,* vol. 61, no. 3, pp. 295-307, May 1995.

[20] *USF Range Image Database,* Univ. of South Florida http://marathon.csee.usf.edu/range/DataBase. html. 2001.

[21] F.P. Preparata and M.I. Shamos, *Computational Geometry: An Introduction.* Springer-Verlag, 1985.

[22] F.A. Sadjadi and E.L Hall, "Three-Dimensional Moment Invariants," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 2, no. 2, pp. 127-136, Mar. 1980.

[23] G.C. Sharp, S.W. Lee, and D.K. Wehe, "Invariant Features and the Registration of Rigid Bodies," *Proc. IEEE Int'l Conf. Robotics and Automation,* pp. 932-937, 1999.

[24] G. Soucy and F.P. Ferrie, "Surface Recovery from Range Images Using Curvature and Motion Consistency," *Computer Vision and Image Understanding,* vol. 65, no. 1, pp. 1-18, Jan. 1997.

[25] A.J. Stoddart and A. Hilton, "Registration of Multiple Point Sets," *Int'l Conf. Pattern Recognition,* p. B6A.5, 1996.

[26] J.P. Thirion, "New Feature Points Based on Geometric Invariants for 3D Image Registration," *Int'l J. Computer Vision,* vol. 18, no. 2, pp. 121-137, May 1996.

[27] J. VandenWyngaerd, L. VanGool, R. Koch, and M. Proesmans, "Invariant-Based Registration of Surface Patches," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 301-306, 1999.

[28] R. Yang and P. Allen, "Registering, Integrating, and Building Cad Models from Range Data," *Proc. IEEE Int'l Conf. Robotics and Automation,* pp. 3115-3120, May 1998.

[29] Z.Y. Zhang, "Iterative Point Matching for Registration of Free-Form Curves and Surfaces," *Int'l J. Computer Vision,* vol. 13, no. 2, pp. 119-152, Oct. 1994.


**Gregory C. Sharp** received the BS degree in electrical engineering from the University of Wisconsin in 1991, and the MS degree in computer science from University of Wisconsin in 1996. He is currently pursuing the MS degree in mathematics and the PhD degree in electrical engineering and computer sciences at the University of Michigan. He is a student member of the IEEE, the IEEE Computer Society, and the ACM.

**Sang W. Lee** received the BS degree in electronic engineering from Seoul National University, Seoul, Korea, in 1981, the MS degree in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Seoul, Korea, in 1983, and the PhD degree in electrical engineering from the University of Pennsylvania in 1991. He is currently an associate professor of media technology at Sogang University, Seoul, Korea. He was an assistant professor in computer science and engineering at the University of Michigan (1994-2000), a postdoctoral research fellow and a research associate in computer and information Science at the University of Pennsylvania (1991-1994), a researcher at Korea Advanced Institute of Science and Technology (1985-1986), a research associate at Columbia University (1984-1985), and a researcher at LG Telecommunication Research Institute (1983-1984). Dr. Lee's major field of interest is computer vision, with an emphasis on physics-based vision, color indexing and recognition, range sensing, and range data registration. He received the outstanding paper award at European Conference in Computer Vision, Margherita Ligure, Italy, 1992, and was a program committee cochair of the IEEE Workshop on Physics-Based Modeling in Computer Vision, June, Cambridge, Massachusetts, 1995, and the IEEE Workshop on Photometric Modeling for Computer Vision and Graphics, June, Fort Collins, Colorado, 1999. He is member of the IEEE and the IEEE Computer Society.

**David K. Wehe** received the BSE degree in nuclear engineering, from the University of Michigan in 1973, the MSE degree in nuclear engineering from the University of Michigan in 1974, and the PhD degree in nuclear engineering and radiological sciences from the University of Michigan in 1984. During his career, he has served as a professor of physics at the US Naval Academy, a practicing engineer at Trident Engineering Associates where he developed analytic predictors for electric utilities to improve plant reliability, and a Eugene P. Wigner Fellow of the Oak Ridge National Laboratory where he performed basic R & D in instrumentation and advanced controls. In his current position on the University of Michigan's College of engineering faculty, Dr. Wehe teaches and performs research in robotics in hazardous environments for the Department of Energy (DOE). He also serves as the director of the Michigan Memorial Phoenix Project, which includes the Ford Nuclear Reactor. He is a member of the IEEE.