Geometric point matching of free-form 3D objects

Christian Schütz

THESE SOUMISE A LA FACULTE DES SCIENCES
DE L'UNIVERSITE DE NEUCHATEL POUR L'OBTENTION
DU GRADE DE DOCTEUR ES SCIENCES

1998

Summary

This work presents a contribution to 3D vision. It investigates a geometric point matching method and its application to the following three vision tasks: object recognition, object digitizing and object inspection.

The 3D vision of interest is characterized by an input of range images provided by 3D scanners which sense the real world objects and give direct access to the 3D object geometry.

The objects processed by the 3D vision system are represented by simple geometric primitives in order to deal with objects of free-form shape. Considered representations are: a cloud of points, triangle meshes or textured triangle meshes.

The main task of the 3D vision system is the matching of free-form surfaces performed by a geometric point matching algorithm. For this purpose, an iterative closest point (ICP) algorithm which matches surfaces represented by low-level primitives is applied. Several aspects of the ICP algorithm are investigated and analyzed. Different contributions extend the ICP algorithm in order to solve the specific problems encountered in the aimed applications.

A first contribution improves the matching performance by integrating several surface features such as surface color, texture and surface orientation in the ICP algorithm. Furthermore, the ICP algorithm is accelerated by fast search routines and data reduction methods.

Secondly, a special surface setup is defined in order to investigate the convergence behavior of the ICP algorithm for several free-form objects. A novel representation called SIC-map presents the convergence results in a comprehensive manner. The interpretation of the SIC-maps provides a key for the successful design of the vision applications.

Finally, the successful working of the extended ICP algorithm is demonstrated by its application to the following three vision tasks.

The object recognition system finds the type and the pose of 3D objects placed randomly on a workspace. To do so, the ICP algorithm matches the sensed test object to all the models in the knowledge database. In order to obtain successful recognition, a setup is proposed which places the test and model surfaces at several well-defined poses from where the ICP algorithm is launched. The successful working of this application is demonstrated for the automatic update of a virtual reality robotics environment.

The object digitizing application matches several views from an object and integrates them into a single model containing the entire object geometry and color information. Human perception provides rough surface registration and is followed by the precise registration performed by the ICP algorithm. The different surfaces do only partially overlap which implies a special adaptation of the ICP algorithm such as proposed in this work.

The object inspection system uses the ICP algorithm to match sensed data with models obtained from digitizing or CAD tools. The matching results of the ICP algorithm provide a measure to inspect the data quality. An augmented reality interface superimposes the matching error to the measured data and permits appealing visual inspection. This system has been applied for micropart inspection and virtual world construction.

Résumé

Ce travail présente une contribution à la vision 3D. Il étudie une méthode d'alignement géométrique de points de surfaces 3D et son application aux trois tâches suivantes: reconnaissance d'objets, numérisation d'objets et l'inspection d'objets.

La vision 3D considérée se caractérise par des données d'entrée sous forme d'images de profondeur fournies par un scanner 3D qui donne l'accès direct à la géométrie 3D de l'objet.

Les objets traités par le système de vision sont représentés par des primitives géométriques simples afin de pouvoir modéliser des objets de formes aléatoires. Les représentations considérées sont: des nuages de points, des maillages de triangle ou des maillages de triangles texturés.

La tâche principale du système de vision 3D est d'aligner des surfaces de formes aléatoires. Elle est réalisée par un algorithme d'alignement géométrique de points 3D qui procède itérativement par choix des plus proches voisins (ICP) et s'applique à des surfaces aux primitives bas-niveau. Plusieurs aspects de l'algorithme ICP sont étudiés et analysés. Différentes contributions étendent l'algorithme ICP afin de résoudre les problèmes spécifiques des applications visées.

Une première contribution améliore la performance de l'alignement de l'algorithme ICP en intégrant plusieurs caractéristiques tels que la couleur, la texture et l'orientation de la surface. En outre l'algorithme ICP est accéléré par des méthodes rapides de recherche du voisin le plus proche et la réduction de données.

Deuxièmement, une configuration spéciale pour positionner les surfaces est définie afin d'étudier le comportement de la convergence de l'algorithme ICP pour plusieurs objets de forme aléatoire. Une nouvelle représentation appelée carte SIC présente les résultats de convergence

d'une façon compréhensive. L'interprétation de la carte SIC fournit une clé pour une bonne conception d'applications de vision.

Finalement, le bon fonctionnement de l'algorithme ICP étendu est démontré par son application aux trois tâches de vision suivantes.

Le système de reconnaissance d'objet doit trouver le type et la pose des objets placés aléatoirement sur un espace de travail. Pour ce faire, l'algorithme ICP aligne l'objet à identifier avec tous les modèles de la base de données. Afin d'obtenir une reconnaissance rapide et correcte, on propose une configuration qui place la surface de test et le modèle à plusieurs positions bien définies d'où l'algorithme ICP est lancé. Le bon fonctionnement de cette application est démontré pour la mise à jour automatique d'un environnement virtuel de robotique.

L'application de numérisation d'objet doit aligner plusieurs vues d'un objet et les intégrer dans un modèle commun contenant l'information entière de la géométrie et de la couleur de l'objet réel. La perception humaine est utilisée pour trouver une estimation de l'alignement des surfaces dans une première étape. L'alignement précis des surfaces est ensuite exécuté par l'algorithme ICP. Le fait que les différentes surfaces ne se superposent que partiellement implique une adaptation de l'algorithme ICP telle que proposée dans ce travail.

Le système d'inspection d'objet emploie l'algorithme ICP pour aligner des objets mesurés avec des modèles obtenus par numérisation ou au moyen d'outils de DAO. Les résultats de l'alignement des points par l'algorithme ICP constituent une mesure de la qualité des objets. En superposant l'erreur d'alignement aux données mesurées, une interface de réalité augmenté permet leur inspection visuelle sous forme attrayante. Ce système a été appliqué pour l'inspection de micro-pièces et la construction de mondes virtuels.

Zusammenfassung

Diese Arbeit präsentiert einen Beitrag im Gebiet der 3D Bildverarbeitung. Sie untersucht eine Methode zur geometrischen Zuordnung von Punkten und ihre Anwendung auf die folgenden drei Aufgaben: Objekterkennung, Objektdigitalisierung und Objektinspektion.

Das angewandte 3D Bildverarbeitungssystem basiert auf Bildern mit Tiefendaten aufgenommen mit einem 3D Scanner, was den direkten Zugriff auf die dreidimensionale Objektgeometrie ermöglicht.

Die verarbeiteten Objekte werden mit einfachen geometrischen Elementen beschrieben, um auch Freiformflächen modellieren zu können. Folgende Oberflächenbeschreibungen werden in Betracht gezogen: Punktwolken, Dreiecknetze und texturierte Dreiecknetze.

Die Hauptaufgabe des 3D Bildverarbeitungssystems besteht in der Ausrichtung von Freiformflächen, welche durch einen geometrischen Punkt-Zuordnungs-Algorithmus ausgeführt wird. Zu diesem Zweck wird ein iterativer Algorithmus angewendet, der die nächsten Punkte (ICP) einander zuordnet und die durch einfachste Elemente beschriebenen Oberflächen aneinander ausrichtet. Mehrere Aspekte des ICP-Algorithmus werden untersucht und analysiert. Verschiedene Beiträge erweitern den ICP Algorithmus, um die spezifischen Probleme der anvisierten Anwendungen zu lösen.

Ein erster Beitrag verbessert die Qualität der Ausrichtung durch das Einbeziehen verschiedener Eigenschaften wie Farbe, Textur und Oberflächenorientierung in den ICP-Algorithmus. Ausserdem wird der ICP-Algorithmus durch einen schnellen Suchalgorithmus und Datenreduktionen beschleunigt.

Zweitens erlaubt die Definition einer speziellen Anordnung der Oberflächen, das Untersuchen des Konvergenzverhaltens des ICP-Algorithmus für verschiedene Freiformflächen. Eine neuartige Darstellungsart genannt SIC-Karte präsentiert die Konvergenzresultate in einer übersichtlichen Art und Weise. Die Interpretation der SIC-Karten ermöglicht den erfolgreichen Entwurf der Bildverarbeitungsanwendungen.

Schlussendlich erbringen die folgenden drei Anwendungen den Nachweis, dass der erweiterte ICP-Algorithmus erfolgreich funktioniert.

Das Objekterkennungssystem bestimmt den Typ und die Lage von dreidimensionalen Objekten, welche zufällig auf einen Arbeitsplatz gelegt werden. Um diese Aufgabe zu erfüllen, vergleicht der ICP-Algorithmus das gemessene Testobjekt mit allen Modellen aus der Wissensdatenbank. Um eine erfolgreiche Erkennung zu erhalten, werden die Test- und Modeloberflächen in verschiedenen Lagen speziell angeordnet und für jede dieser Konfigurationen der ICP-Algorithmus gestartet. Das erfolgreiche Funktionieren dieser Anwendung ist nachgewiesen anhand der automatischen Aktualisierung einer Roboterumgebung in einer virtuellen Welt.

In der Objektdigitalisierung werden verschiedene Objektansichten mit Hilfe des ICP-Algorithmus aneinander ausgerichtet und in ein einziges Model integriert. Dieses Model enthält nun die ganze Geometrie und Farbinformation des digitalisierten Objektes. Das menschliche Vorstellungsvermögen wird zu Hilfe genommen, um eine grobe Ausrichtung der Oberflächen abzuschätzen. Darauf folgt die präzise Ausrichtung, ausgeführt durch den ICP-Algorithmus. Die verschiedenen Flächen überlappen nur teilweise, was eine spezielle Anpassung des ICP-Algorithmus nötig macht, wie in dieser Arbeit beschrieben.

Das Objektinspektionsystem benützt den ICP-Algorithmus, um gemessene Daten an digitalisierten oder aus CAD-Programmen stammenden Modellen auszurichten. Die Resultate der Ausrichtung, durchgeführt durch den ICP-Algorithmus, stellen ein Mass zur Inspektion der Datenqualität zur Verfügung. Eine spezielle Schnittstelle visualisiert das Resultat der Inspektion nach dem Prinzip der angereicherten Realität, indem der Fehler der Ausrichtung über die gemessenen Daten gelegt wird. Dieses System wurde für die Inspektion von Mikroteilen und die Konstruktion von virtuellen Welten verwendet.

Table of contents

Chapter 1
Introduction

1.1 Motivation

Vision systems are of interest to industry since they allow fast and non-contact inspection. Previously, the processing of the image data needed expensive and specialized hardware. New computer generations permit the handling of the image processing on standard personal computers and to lower equipment costs significantly. Especially in 2D vision systems which are widely used in industry for implementing machine vision tasks.

3D vision is just emerging. With the availability of new and cheaper 3D scanners, 3D vision is becoming more accessible. This allows object recognition and inspection tasks to extend to 3D where they had previously only been done in 2D. Complete new systems integrating virtual reality environments are being developed for applications such as telemanipulation and multimedia.

1.2 Goals and applications

This work presents a contribution to 3D vision. It investigates a geometric matching method and its application to the following three vision tasks: object recognition, object digitizing and object inspection.

Object recognition: The type and pose of different objects placed on a working table have to be identified. Here, the term "pose" includes the object position and orientation. For every object, a model representation is stored in a database. The recognized objects are manipulated by a robot. Such a vision system can help to automate an assembly task.

Object digitizing: An object surface is sensed from different point of views. The aim is to assemble the different acquisitions in order to reconstruct a model of the complete object surface. Object digitizing is effectively used in reverse engineering applications to build CAD models from hand-made or historical objects where no digital data is available.

Object inspection: The surface geometry of a manufactured object is acquired and compared to a CAD model. The surface inspection allows the qualification of the production process.

All applications encounter a problem where two surfaces have to be matched. The following demands have to be observed:

Precise matching: The surfaces have to be matched with high accuracy to ensure the correct working of all applications.

Free-form objects: The processing of objects of arbitrary shape is necessary.

Access to geometry: Direct access to the surface geometry is a prerequisite for successful object inspection or digitizing.

This work presents a surface matching method which is used for all three applications and which fulfills the above demands.

Next, a short overview motivates the use of free-form surfaces and their shape acquisition.

Free-form objects

Modern industrial design, especially multimedia applications, deal a lot with smooth sculpted objects. The developed vision algorithms are designed especially to process free-form shapes that are much more complex than simple polyhedral objects (see Figure 1.1 for some examples).

Shape acquisition

3D scanners are well-suited for the acquisition of objects of complex shape since they can measure many surface points at high speed. In this work, optical 3D scanners are used to measure object surface geometry without contact. The acquired data is represented by absolute 3D coordinates. This gives direct access to the surface geometry. See [JIA] for an introduction to the processing of 3D scanner data.

1.3 The matching problem

The problem of matching two surfaces appears in all focused applications. Every task has to establish correspondences between two surfaces in order to register them. In this context, surface registration refers to the alignment of two surfaces. For example, the object recognition task matches a sensed real world scene and a set of model shapes stored in a model database on a computer.

Figure 1.1 illustrates the matching problem encountered in object recognition for some typical objects used in this work.

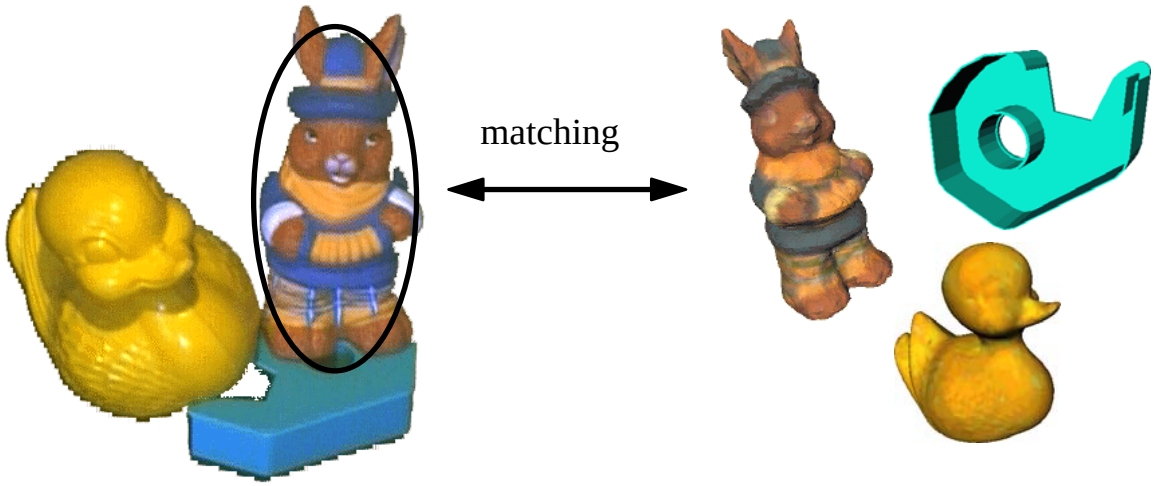


Figure 1.1 Illustration of matching problem

There are different levels of correspondence establishment. For some tasks, only the type of the object present in the scene has to be found. Other applications need to know the object type and pose, as is the case for the applications examined in this work.

Here, the term matching refers to type identification or shape localization or a combination of both of them. The matching establishes the correspondence of a sensed surface and a model surface database or two sensed surfaces.

The implementation of a matching system depends on the surface representation. Here, three matching classes are distinguished. The two surfaces to be matched are referred to as test and model:

Image matching: The test and the model are images containing for example intensity or depth information. Image correlation locates the model image in the test image. The result is the pose of the model in the 2D space of the test image.

Graph matching: The test data is segmented and labeled. The resulting abstract object representation is organized in a graph structure. Graph search methods compare the high-level test and model representations. Usually, only the object type is identified.

Geometric matching: The test and the model surface are represented by their geometry. The surfaces are aligned or registered by minimizing the measure of geometry disagreement. The resulting transformation corresponds to the pose of the matched surfaces in 3D space.

In this work, a geometric matching method working in 3D space is applied. It best fits the requirements of the aimed applications defined in Section 1.2.

1.4 Structure of the report

The following chapters are organized as follows:

Chapter 2 discusses several aspects of free-form object surfaces. Their geometry is defined and their representation and acquisition are discussed. An optimal low-level surface representation is selected for this work.

The geometric matching method chosen for this thesis is presented in Chapter 3. The basic working principle is explained and contributions which improve matching both in speed and quality are proposed. Special surface configurations are defined in order to measure the convergence behavior of the matching algorithm.

The three chapters Chapter 4, Chapter 5 and Chapter 6 present the implementations of the geometric matching for the three applications mentioned above. Each presentation starts with an introduction, a comparison with other work and finishes with the results and a conclusion. Furthermore, these chapters emphasize the different implementation aspects, which adapt the geometric matching algorithm for the particular needs of each application.

The last Chapter 7 concludes this work with a summary of the contributions of this work and a discussion of the limitations and of future work.

Chapter 2
Free-form objects in 3D vision

The objects considered in this work are three-dimensional and of free-form shape. This chapter starts with a definition of the term free-form. Then, follows a presentation of the two main approaches used to represent free-form objects in vision algorithms. Finally, the primitives which model free-form objects in this work are selected and the different principles with which free-form surfaces are acquired are discussed.

2.1 Definition of the free-form term

Since surfaces are what is seen by a vision sensor, the surface representations are important for computer vision. Therefore, the definition of the term "free-form" in this work is based on surface geometry. In addition, the terms free-form object or shape are used to tell that their surface is of free-form type.

The geometric surface modeling problem was treated in an exhaustive manner by Besl in the late eighties [BESc] and [BESd]. A qualitative definition of free-form surfaces is given in [BESc] and is summarized here. Additional restrictions are added to the free-form definition of Besl in order to adapt it for the objects used in the applications implemented in this work.

Definition

Some typical examples of free-form shapes are faces, sculptures and terrain as shown in Figure 2.1. Free-form objects are smooth, complex and their surfaces may be locally convex or concave. The objects considered in this work are solid and have no moving parts. A free-form surface is not limited to piecewise-planar, piecewise-quadric or piecewise-superquadric primitives. Having one point on the surface and heading in one direction, it is impossible to know how the surface geometry will evolve. However, surface normals are well-defined and continuous apart from isolated edges or vertices.

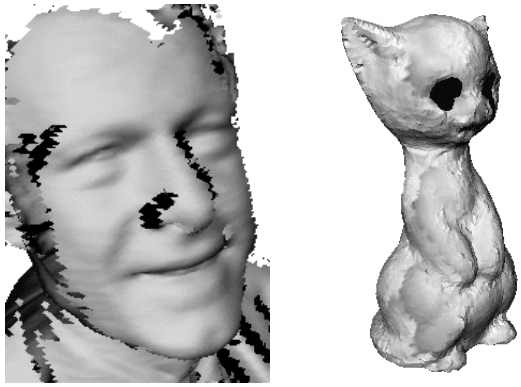


Figure 2.1 Human face and sculpture sensed with a 3D scanner

2.2 Surface representations

In general, different tasks have different demands on the object representation. A qualitative shape representation, for example a structural representation of object faces, may be sufficient for object recognition. Nevertheless, tasks such as industrial inspection need accurate quantitative surface measurements. Therefore, geometric primitives that approximate the object's surface geometry are more appropriate.

In the following, the classes of appearance-based and primitive-based approaches are presented and their suitability for the different tasks to be implemented is discussed. Appearance-based representations rely on a qualitative description of the object's surface whereas primitive-based approaches model the surface geometry quantitatively.

The following reflections are based on contributions of a recent vision workshop [HER].

2.2.1 Appearance-based representation

Appearance-based representations consist of a set of several object views. A set of simple images showing the object under different view angles is already sufficient to build a model database. The object information stored in the model database is usually an abstraction of the image collection and may be of geometric or also photometric nature.

Examples

Dorai defines a shape index based on curvatures [DOR]. An object is sensed from different views by 3D scanner data. An object's shape information is expressed by a shape spectrum, which is the histogram of all shape index values over the object's surface (see Figure 2.2). The set of shape spectrums for all views of the object represents the whole object.

Similarly, Besl proposes a crease angle histogram [BESa] to map a shape to a vector and Stein defines a "splash" representation which maps the orientations of the local normal vectors of a surface region to an index [STE].

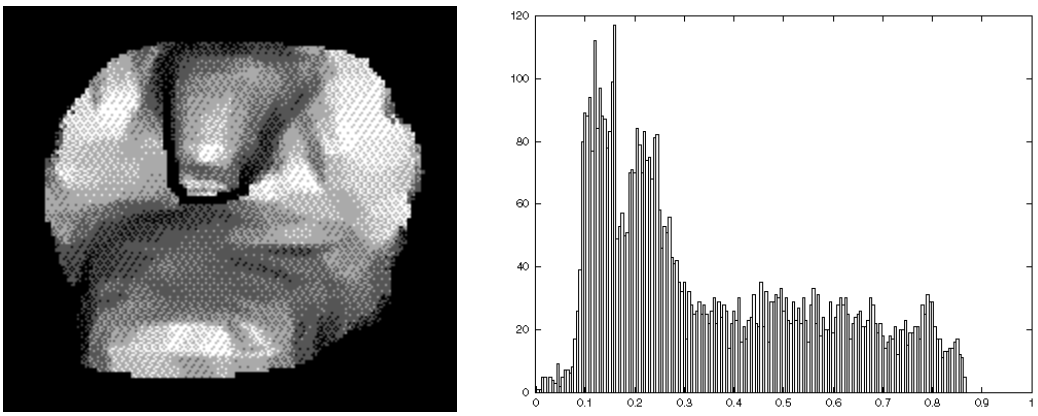


Figure 2.2 Appearance-based representation of a sculpture [DOR]

Suitability

Appearance-based methods do not contain an explicit description of object geometry and can therefore be applied to any object shape. However, they usually need a large number of views to represent the whole object shape. This results in a complex search when comparing a test view with the entire model database. An effective indexing scheme and the use of simple dissimilarity measures can reduce this search time.

Even if appearance-based approaches are well-suited to represent free-form objects, they are not appropriate for the applications aimed at in this work which need precise registration and direct access to surface geometry (see Section 1.2).

2.2.2 Primitive-based representation

Primitive-based approaches model an object surface by geometric primitives. Usually a single primitive, even when it has a large number of parameters, cannot capture the whole shape of a free-form object. Therefore, objects are in general decomposed into several parts, which are modeled separately by primitives.

The two main problems of primitive-based approaches are the selection of an appropriate primitive and the segmentation i.e. decomposition of the object shape into regions described by these primitives.

Primitive classes

Classical vision has focussed on particular classes of constrained geometric shapes as for example polyhedrons modeled by polygons (see [AMA] and [GIN] for some work done at our lab). Other examples for low-level primitives are set of points, lines or planar patches.

High-level primitives have more parameters and have been used in several object recognition systems [BESd]. Spheres, ellipsoids and cylinders can be described by second-degree algebraic surfaces called quadrics. Superquadrics define even more complex shapes such as tori. General cylinders use precise generative rules instead of numerical parameters for object representation. They move a 2D cross-section along a line while several parameters such as scale or rotation may change.

Such representations have been used to model human bodies in virtual reality applications.

Parametric primitives approach a surface by polynoms or B-Splines attached to a set of control points. For example, NURBS are parametric surfaces defined over a rectangular parameter domain, which again may be trimmed by some curve in order to represent discontinuities. In industry, the IGES (Initial Graphics Exchange Specification) standard uses NURBS (Non-Uniform Rational B-Splines) for free-form surface representation [BESc].

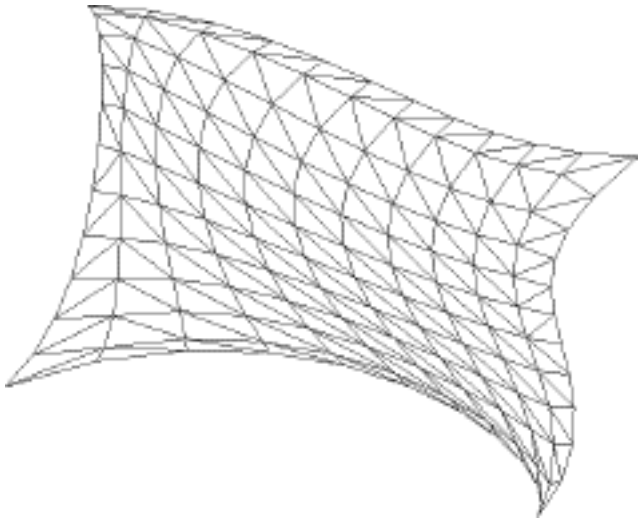


Figure 2.3 Example of a NURBS surface

Surface segmentation

Surface segmentation consists of the decomposition of an object surface into regions that can be precisely modeled by a given primitive. The decomposition of free-form surfaces is a difficult process and often does not lead to a stable representation. Especially, if there are several types of primitives available it is not obvious which one to select since this will again influence the decomposition (a chicken-and-egg problem, so to speak).

Promising 3D data segmentation techniques are often curvature-sign-based. The curvature signs are defined with respect to certain thresholds. Li shows that these techniques are not very reliable for free-form shapes especially if sensor noise is present in the data [LI]. It is difficult to define the curvature thresholds in advance. Usually, a

successful segmentation is only achieved after several trials and visual inspection of the resulting regions.

Figure 2.4 shows the segmentation results for a free-form object (a toy duck) scanned from two different view points. The surface geometry is represented by range images where the pixel intensity corresponds to the distance between the corresponding object point and the camera. The object surface is segmented into regions using a simple continuity criterion based on second order derivatives. The resulting segmentation is not stable and suggests the difficulty of the task.

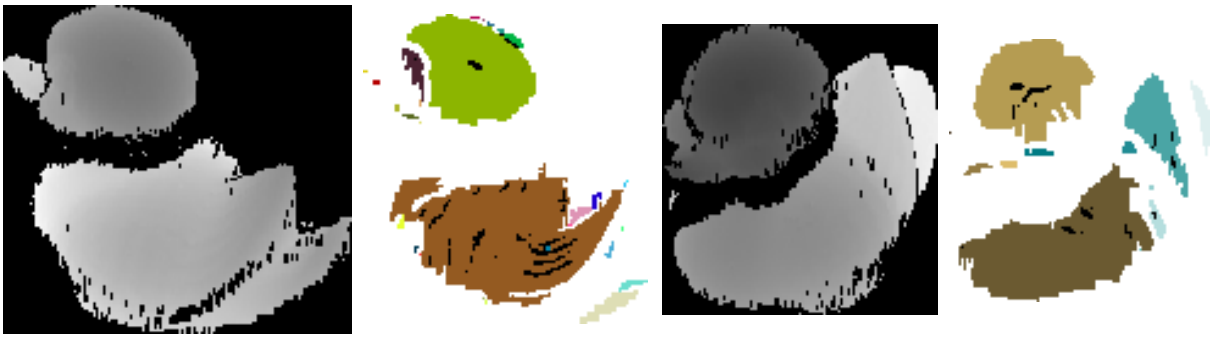


Figure 2.4 Surface segmentation with a surface continuity criterion

High-level versus low-level primitives

Because high-level primitives are controlled by a large number of parameters, they permit to model even complex surfaces with few primitives. However, it is difficult to obtain a stable and meaningful surface segmentation for high-level primitives.

Low-level primitives usually cover a small area and quite a large number of them is needed to represent a surface. On the other hand, surface segmentation is simpler. This segmentation does not have to be very stable since it is in general followed by an abstraction process, which groups different low-level primitives into a high-level representation.

Suitability

Primitive-based approaches model the surface geometry precisely and are therefore well suited for the tasks to be implemented in this work (compare Section 1.2).

2.3 Advantages of low-level primitives

Representation qualities

The low-level primitives considered in this work are points and triangles. The use of triangles, especially, has several advantages compared to high-level primitives as shown exhaustively in [BESa]. They are cited here:

- A cloud of points or a triangle mesh can both be obtained directly from the data measured by a 3D scanner as shown later in Section 2.4.
- If the object surface is sufficiently densely sampled, points or triangles can represent any shape.
- The surface segmentation into such primitives becomes superfluous.

On the other hand, the number of primitives gets very large since these low-level primitives cover only a small surface area. Efficient data structures and fast computers with much memory are needed to handle the large amount of primitives. Yet, these problems have become less important during recent years since memory prices have dropped drastically and workstation power is available on personal computer now.

Matching qualities

Low-level primitives are effective to compute quantitative shape similarity since they precisely represent free-form surface and the definition of a primitive dissimilarity can be a simple Euclidean distance.

Data exchange, display and reusability

The data exchange between different modules of a vision system is important to ensure effective work progress [BESa]. If there is a need for data converters, a digitized model built by a digitizing module may not be easily used in an object recognition or inspection module and needs further processing for on-screen visualization. Low-level primitives are very easily interchangeable since they do not result from a data abstraction process.

Graphic display processors can directly import triangle meshes or point clouds for display since they use the same primitives to calculate the image rendering.

Most CAD applications have converters to export models with a surface triangulated at different levels of detail [BESa]. This allows to integrate CAD data directly into the vision algorithms developed in this work.

2.4 Shape acquisition

Next follows a description of the most common optical 3D scanner principles and in more details the working principles of the 3D scanners used for the acquisitions done for this work. Different surface representations such as could of point, triangle mesh and textured triangle mesh are derived from the scanner data.

2.4.1 Working principles of optical 3D scanners

During the last few years optical 3D scanners have made substantial progress and several models have been commercialized [SCHa]. Several vision techniques exist to sense 3D object shapes without the need to make contact with the sample. Usually, the sensed 3D information referred to as depth corresponds to the set of vectors linking the sensor and the visible points on the object surface. Most of these systems can be classified according to one of the following techniques:

Stereo vision: A calibrated camera pair observes the sample. Subparts of both images are brought into correspondence and the depth is calculated by triangulation.

Active triangulation: Light spots or lines are projected on the sample and observed through a calibrated camera under a different view angle. The projected light is detected in the camera image and the depth is calculated by triangulation.

Focus/defocus: Depth from focus is calculated from the lens configuration corresponding to the best object focus. Depth from defocus is obtained from the optical camera and lenses parameters combined with an image blur measure.

Time of flight: A laser beam is pointed at the object and depth is obtained from time of flight or from phase shift measurements.

The 3D scanners used in this work belong to the class of active triangulation scanners and are described in more detail below.

Laser line triangulation

A laser beam is directed through a cylindrical lens, which then emits a sheet of light. A CCD camera observes the line, which results from the intersection of the sheet of light with the object surface. The vertical offset of the line in the camera image is calculated. The calibrated laser and camera configuration allow the determination of the depth information using triangulation methods. The use of coherent laser light and a corresponding camera filter makes these 3D scanners low sensitive to ambient illumination. The resulting information only represents a scene profile and some scanning like mechanical translation of the sensor is necessary to acquire a complete surface.

2D stripe projectors

This method is an extension of the previous laser line method to several sheets of light. This scanner works on the principle of space coding with a projected stripe pattern and triangulation. Here, a sequence of two dimensional stripe patterns is projected onto the sample. The different light sheets have a unique binary code defined by the 0 or 1 values of the corresponding column in the projected images. The light sheets are identified in the camera image by detecting sequentially over all

projections the binary code for each sheet (see Figure 2.5). This allows the direct acquisition of a complete scene view without the need to move the scanner. The 3D coordinates of a scene point are calculated using the identified sheet code and its camera image coordinates. The depth resolution of the installation at our laboratory is about 0.5 mm for a measurement space of 200 mm diameter (resulting in a precision of about 1:400).

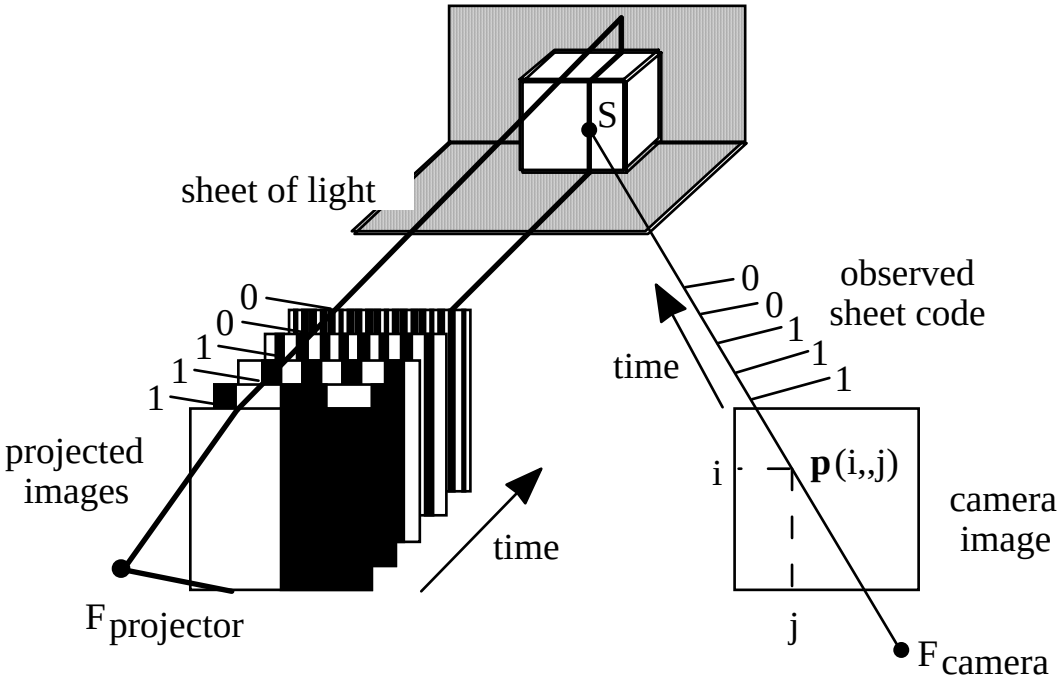


Figure 2.5 Working principle of a space coding 3D scanner

2.4.2 3D scanner output

Range image

The geometric data of a whole scene measured by a 3D scanner is usually organized in a 2D array called a range image. The pixel intensity in a range image corresponds to the depth information, for example the z coordinate of a surface point (see Figure 2.6). Also the x and y coordinates may be stored in images. The three images build a stack where the surface geometry is available in a sensor-centered coordinate system for every pixel: $\mathbf{p}(i,j) = (x(i,j), y(i,j), z(i,j))$.

Colored range image

Some scanners are able to acquire color or intensity information together with the 3D measurements because they employ white light to illuminate the object space. The depth (x, y, z) and color (red, green, blue) information is registered and available for every pixel since both informations are extracted from the same camera image. The incorporation of the color feature allows the extension of the information available for every pixel in the image stack to

$$\mathbf{p}(i,j) = \big(x(i,j), y(i,j), z(i,j), r(i,j), g(i,j), b(i,j)\big), \tag{2.1}$$

which represents a colored range image. Figure 2.6 shows the output of a 3D scanner from a sensed toy rabbit.

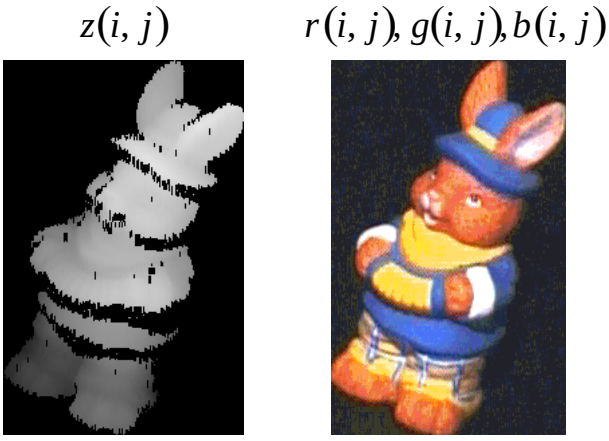


Figure 2.6 Range and color image of the 3D scanner output

Intrinsic color

When shapes are illuminated from different view points the color of corresponding points is not necessarily invariant. What surface rendering and matching tools need is the intrinsic surface color.

One method to obtain the intrinsic color is to use a reflection model. Already a simple lambertian model is useful if there is only one light source involved in the scanning process. Since the 3D scanner gives access to the object geometry, the direction of light incidence is known at every surface point. Assuming a diffuse object surface the sensed color $\mathbf{c}_s$ can be corrected as follows to obtain the intrinsic color $\mathbf{c}_i$:

$$\mathbf{c}_i = \frac{\mathbf{c}_s}{\cos(\theta)} \text{ with } \mathbf{c} = (r, g, b) \tag{2.2}$$

where θ is the incidence angle formed by the local surface normal vector and the direction vector of the light source. More complex illumination models are proposed in [GODb].

Although this method produces quite good results, experiments at our laboratory showed that better results are obtained if the color acquisition is decoupled from the 3D scanning. A color image is acquired in a separate step where an ambient light source uniformly illuminates the object.

Range image properties

The vector **p** of (2.1) is filled with a special value called NIL when the 3D scanner could not calculate a reliable 3D measurement. This happens for example, when a point on the object surface is not illuminated. Such invalid pixels are drawn in black in range images.

Since the range image is a perspective projection of the 3D scene onto an image plane, points which are next to one another in the range image are not necessarily neighbors on the object surface as illustrated by Figure 2.7.

However, points that are next on the object surface are still neighbors in the image space after perspective projection.

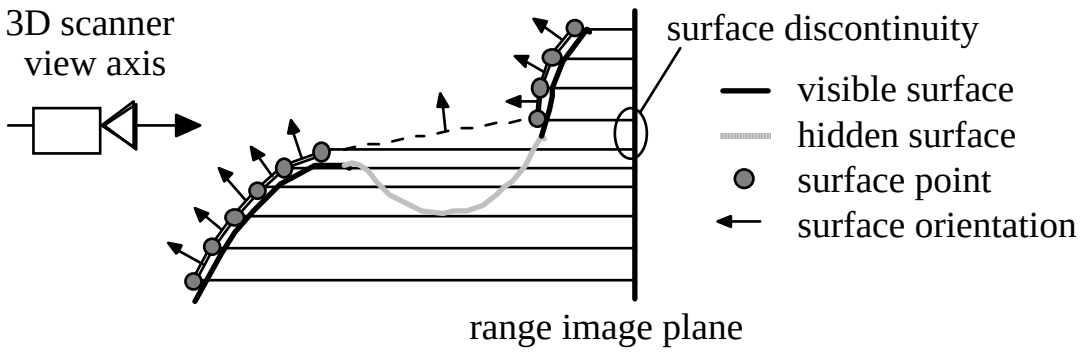


Figure 2.7 Surface discontinuity in a range image

These three properties have to be considered for the range image triangulation presented in the next section.

2.5 Surface representation

Points and triangles have been selected to represent free-form surfaces. This section presents methods how to extract such representations from the range image.

Several surface points build a cloud of points which is a set of feature vectors containing the Cartesian coordinates and possibly the color vector of a surface point.

A triangle mesh interconnects the surface points with non-overlapping triangles covering the whole surface. It is defined by a list of vertices and by a list of 3-tuples containing the coordinate indices of the triangle vertices. The color vectors may be assigned to each vertex or each triangle face and are stored in a separate list.

A textured triangle mesh differs from a triangle mesh in the sense that the color is stored in an image. Every triangle vertex obtains a pointer to the corresponding image pixel.

2.5.1 Cloud of points

This simple representation is directly obtained from the range image. The point vectors corresponds to the pixel information (2.1) of the image stack. Only pixels which are not NIL are considered for the cloud of points.

2.5.2 Triangle mesh

A cloud of points representing an object surface is inefficient for several applications since it is a discrete surface representation. Triangulated surfaces are preferable for the object modeling used for realistic object rendering, multimedia and reverse engineering applications.

The step from a cloud of measured surface points to a triangle mesh needs in general sophisticated triangulation methods, which are expensive in terms of computation time and memory storage [SCHb].

The fact that most 3D scanners provide the measurements in a regular grid, namely the range image, makes the triangulation of the surface become straightforward as proposed by [RUT]. Observing the discussed range image properties, a triangulated surface is obtained as follows.

Range image triangulation

The range image is explored from the upper left to the lower right corner and a local triangulation is performed for every pixel and its three neighbors "east", "south" and "south-east". A local triangulation algorithm creates two triangles covering the square grid mesh formed by the current pixel and the three pixels with column and row indices increased by one. If the four points represent valid 3D data ($\mathbf{p} \neq \text{NIL}$) two possible ways exist to triangulate the square depending on the diagonal which is selected. Following the principle of the Delaunay triangulation, the shortest diagonal is selected which in turn also creates triangles with a maximal size of the smallest angle. This results in a smooth surface approximation since there is a minimal number of triangles with long edges. If one of the four points is not valid then one triangle is constructed with the remaining three points and no triangle is built at all if more than one point is missing (see Figure 2.8).

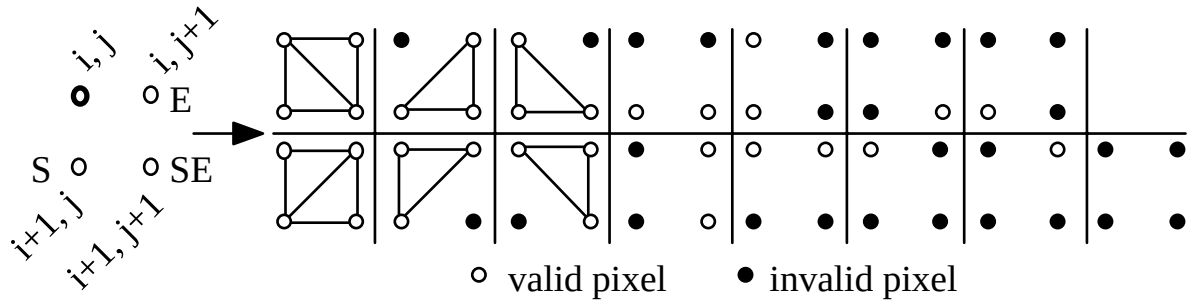


Figure 2.8 Range image triangulation

Range image filtering

Note that for several applications, the full range image resolution is not necessary and a subsampling of the rows and columns by a factor *r* allows

the reduction of the number of points and the processing of the data faster.

Furthermore, an additional step is necessary to ensure that occluded parts are not covered by triangles: the connection of triangles separated by a discontinuity step must be avoided [TUR] [RUT] (see Figure 2.7).

First, triangles with edges larger than $4 \cdot s \cdot r$ are rejected, where s is the sampling grid distance which is about 0.5 mm for the used 3D scanner and r the reduction factor introduced before. Secondly, triangles with an angle between the triangle normal vector and the sensor view axis exceeding 75° are also rejected.

Figure 2.9 shows the results of the triangulation performed on a subregion of the range image shown in Figure 2.6. The different representations include a cloud of points, a triangle mesh and a colored triangle mesh.

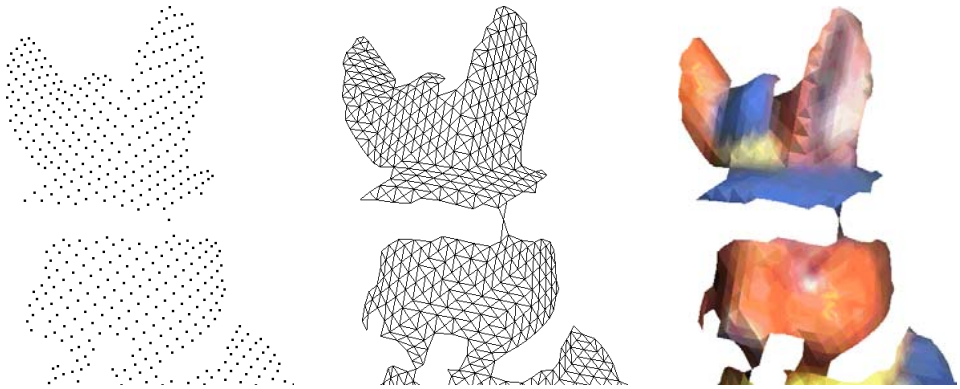


Figure 2.9 Cloud of points, triangle mesh and colored triangle mesh

2.5.3 Textured triangle mesh

For some applications such as multimedia, which usually exchange object data over a network with a limited transfer bandwidth, the full geometric shape resolution is not necessary. As much as possible points are discarded in order to keep the object model size small.

Surface point reduction methods successively reduce the number of points representing a shape thus conserving the geometry up to a certain error [CHE] [GAR] [SOUa]. Such algorithms preserve the surface geometry details but color details get lost. For example, the details of the eye of the rabbit in Figure 2.9 are lost and the color transitions are blurred.

Texture image mapping provides a solution to this problem. Instead of assigning the color information only to the triangle vertices, a color image of the object surface is applied onto the triangle faces. The display of texture mapped objects is fast since it is implemented in the hardware on modern graphics display cards.

Texture mapping functions need to know the corresponding coordinates (i, j) in the texture image for every 3D data point (x, y, z) in order to apply the color image onto the surface geometry. The 3D scanner used for this work acquires a color image registered with the geometric information (see Section 2.4.1). This allows it to easily determine the corresponding coordinates in the color image since pixels correspond with the pixel in the range image containing the 3D information.

The color information is now separated from the geometric data and stored in a color image. Images can be compressed very efficiently using algorithms with loss as JPEG. Practical experiments showed gains in model file size of up to a factor of ten for a compressed textured triangle mesh compared to a densely sampled and colored triangle mesh having both the same appearance. The texture images allows the detailed color information of an object to be stored. Figure 2.10 shows how a sampled object surface can be augmented by texture mapping of the color image. Note that the small stripes on the arm of the bear are present in the textured triangle mesh.

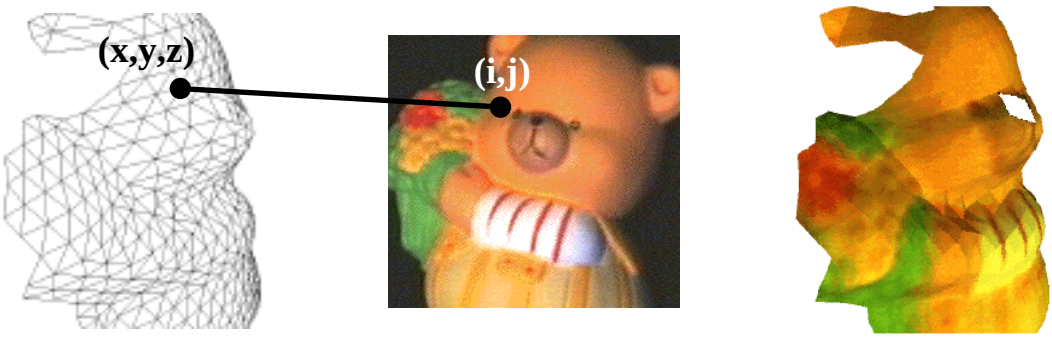


Figure 2.10 Color texture image mapping

Chapter 3
Geometric point matching

This chapter presents and analyzes the surface matching algorithm chosen for this work. Based on the discussions in the previous chapters, the selection of a geometric point matching algorithm is recommended.

First, the theoretical aspects of the algorithm are explained. Then follow several contributions that enhance the matching algorithm. They provide a faster execution time, better matching precision and more reliable convergence.

3.1 Motivation

Geometric matching registers two surfaces with the help of their geometry. It calculates the transformation, which brings one surface into correspondence with the other. This allows precise object localization in 3D and inspection of the surface geometry to be performed.

In order to represent the free-form surfaces, low-level primitives are selected: a cloud of points or a triangle mesh are well-suited.

The combination of geometric matching and low-level surface representations meets best the requirements for the focused applications. Geometric point matching combines both advantages and is chosen and implemented in this work.

3.2 Problem statement

The geometric point matching problem can be defined as follows: Given two sets of 3D points representing two surfaces called P and X, find the rigid transformation as defined by the rotation $\mathbf{R}$ and the translation $\mathbf{t}$, which minimizes the sum of the Euclidean square distances between the corresponding points of P and X. The sum of all square distances gives rise to the following surface matching error

$$e(\mathbf{R},\mathbf{t}) = \sum_k \left\| (\mathbf{R}\mathbf{p}_k + \mathbf{t}) - c(\mathbf{p}_k) \right\|^2, \quad \mathbf{p}_k \in P \tag{3.1}$$

which has to be minimized

$$e = \min_{\mathbf{R},\mathbf{t}} e(\mathbf{R},\mathbf{t}). \tag{3.2}$$

The function c associates with every point of P a corresponding point of X:

$$c : P \to X \,. \tag{3.3}$$

It is called the correspondence function c. If it is known, the rigid transformation ($\mathbf{R}$, $\mathbf{t}$) which minimizes (3.2) can be calculated in a closed-form (see Section 3.5). Practically, the function c is unknown because there is in general no available correspondence information between X and P. Corresponding points of different surface acquisitions are not labeled in the data obtained from a 3D scanner. Therefore, solving (3.2) for $\mathbf{R}$ and $\mathbf{t}$ is a complex minimization process since c is unknown.

3.3 Iterative geometric matching

Although it is difficult to find a direct solution for the geometric point matching problem, it may be solved by an iterative approach. The complex calculation of the rigid transformation ($\mathbf{R}$, $\mathbf{t}$), which solves (3.2), is decomposed into several steps that successively reduce the matching error.

At each iteration, the correspondence function c is approximated by an estimated function $\hat{c}$. As the terms of (3.1) are now defined the optimal rigid transformation, which minimizes the matching error at the current iteration, can be calculated.

The principal idea of the iterative approach is to use the value of $\mathbf{R}$ and $\mathbf{t}$ from the previous iteration step in order to build $\hat{c}$ at the current iteration. At every iteration i, a least-square method calculates the

optimal rigid transformation $(\mathbf{R}^{(i)}, \mathbf{t}^{(i)})$ which minimizes the current matching error. The total transformation $(\mathbf{R}, \mathbf{t})$, which combines the results of the different iterations, is updated as follows: $\mathbf{R} = \mathbf{R}^{(i)}\mathbf{R}$ and $\mathbf{t} = \mathbf{R}^{(i)}\mathbf{t} + \mathbf{t}^{(i)}$.

This method reduces the complex geometric point matching error of (3.1) to a simpler one, defined below and minimized at every iteration step.

$$e\left(\mathbf{R}^{(i)}, \mathbf{t}^{(i)}\right) = \sum_k \left\| \left[\mathbf{R}^{(i)} \left(\mathbf{R}\mathbf{p}_k + \mathbf{t} \right) + \mathbf{t}^{(i)} \right] - \hat{c}(\mathbf{p}_k) \right\|^2 \qquad (3.4)$$

The correspondence function $\hat{c}$ has to be chosen such that the iterative geometric matching converges towards a minimum. The matching error e can be minimized by a closed-form solution presented in Section 3.5.

3.3.1 History of iterative geometric point matching

Several authors have proposed algorithms to solve the geometric point matching problem. Early work using heuristic methods was done by Potmesil [POT] in 1983. About ten years later several authors published independent work based on the same idea [BESb] [CHE] [CHA] [MEN]. All the publications present an iterative geometric point matching algorithm. During one iteration, point correspondences of two surfaces are established in order to estimate c and the rigid transformation, which minimizes the matching error of the corresponding points, is calculated. The approaches differ mainly in how the point correspondence, i.e. the function $\hat{c}$, is established.

Chen and Medioni [CHE] calculate the corresponding point for a point from one surface by intersecting its surface normal with the second surface. Besl and McKay [BESb] minimize the Euclidean distance of a point from one surface to all the points from the second surface. The determination of the rigid transformation, which minimizes the distance between corresponding points, is done with a closed-form solution in [BESb] and with an iterative minimization in [CHE].

In [MEN] scanner data is matched with CAD models. Champleboux shows registration results of human face data [CHA].

In this work the approach of [BESb] has been chosen and extended. This iterative solution for the geometric point matching problem uses a closest point function for $\hat{c}$ as described in the next section.

3.3.2 Iterative closest point (ICP) matching

The distance error of the closest points of P and X is minimal, when P and X are registered. Assume that the transformation (**R**, **t**) which registers P and X is known, then c can be estimated by the function $\hat{c}$ which associates every point of P with the point with the smallest Euclidean distance of X:

$$\hat{c}(\mathbf{p}) = \underset{\mathbf{x} \in X}{\operatorname{argmin}} \left\| (\mathbf{R}\mathbf{p} + \mathbf{t}) - \mathbf{x} \right\|^2 . \tag{3.5}$$

The correspondence function $\hat{c}$ performs a closest point search for every point of P. It uses the square of the Euclidean distance in order to avoid the expensive square root computation.

The minimization of the error in (3.1) is still difficult even when using $\hat{c}$ defined in (3.5), since $\hat{c}$ itself is an optimization problem of **R** and **t**. However, the iterative approach which minimizes (3.4) in combination with (3.5) leads to an iterative closest point matching which provides a solution for the geometric matching problem.

ICP algorithm

The iterative closest point (ICP) matching method can be formulated in a procedural description as follows:

- input: Two sets of 3D data points P and X and the total transformation initialization $\mathbf{R} = \mathbf{I}$ and $\mathbf{t} = \mathbf{0}$.
- output: Transformation (**R**, **t**), which registers P and X.
- iteration i:

1. Use the following coupling distance definition

$$d(\mathbf{p}, \mathbf{x}) = \left\| \mathbf{p} - \mathbf{x} \right\|^2 \tag{3.6}$$

to build the set $C^{(i)}(\tilde{\mathbf{p}}_k, \mathbf{y}_k)$ of the N closest point pairs, N being the number of points in P. The closest point $\mathbf{y}_k$ is defined as follows

$$\mathbf{y}_k = \hat{c}(\tilde{\mathbf{p}}_k) = \underset{\mathbf{x} \in X}{\operatorname{argmin}} \; d(\tilde{\mathbf{p}}_k, \mathbf{x}) \text{ with } \tilde{\mathbf{p}}_k = \mathbf{R}\mathbf{p}_k + \mathbf{t} . \tag{3.7}$$

2. Define the matching error of the couplings in $C^{(i)}(\tilde{\mathbf{p}}_k, \mathbf{y}_k)$ as a function of $\mathbf{R}^{(i)}$ and $\mathbf{t}^{(i)}$

$$e\left(\mathbf{R}^{(i)},\mathbf{t}^{(i)}\right)=\sum_{k=1}^{N}\left\|\left(\mathbf{R}^{(i)}\tilde{\mathbf{p}}_k+\mathbf{t}^{(i)}\right)-\mathbf{y}_k\right\|^2 \tag{3.8}$$

and calculate the rigid transformation ($\mathbf{R}^{(i)}$, $\mathbf{t}^{(i)}$) such that

$$e^{(i)}=\min_{\mathbf{R}^{(i)},\mathbf{t}^{(i)}} e\left(\mathbf{R}^{(i)},\mathbf{t}^{(i)}\right). \tag{3.9}$$

3. Let $\mathbf{R}=\mathbf{R}^{(i)}\mathbf{R}$ and $\mathbf{t}=\mathbf{R}^{(i)}\mathbf{t}+\mathbf{t}^{(i)}$.
4. Stop if the matching has converged to a minimum or if the maximal number of iterations is reached.

Figure 3.1 shows the data flow graph of the iterative closest point matching together with an example of the matching of two L-shaped objects.

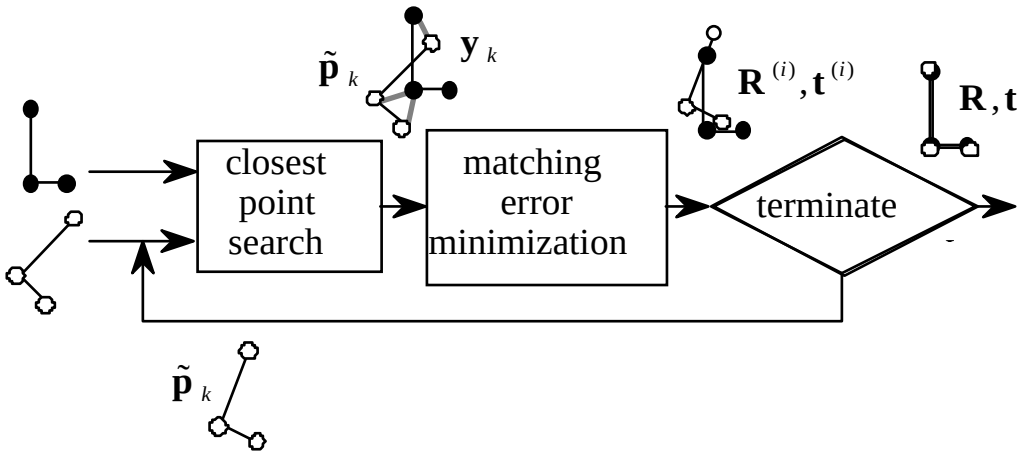


Figure 3.1 Data flow of ICP algorithm

The example in Figure 3.2 illustrates the different point sets P, X and $Y=\{\mathbf{y}_k\}$ of the closest point search.

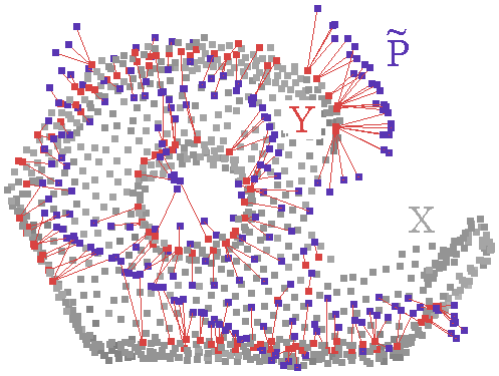


Figure 3.2 Closest point couplings *Y* of the ICP algorithm

3.3.3 Convergence of ICP algorithm

This section proves the convergence of the ICP algorithm. At iteration *i* the closest point search establishes the couplings $C^{(i)}$ having an overall coupling error $d^{(i)}$ defined by

$$d^{(i)} = \sum_{k=1}^{N} d_k \text{ with } d_k = \left\| \tilde{\mathbf{p}}_k - \mathbf{y}_k \right\|^2 . \tag{3.10}$$

Note that the matching error *e* of (3.8) uses the same distance measure as the coupling distance *d* of (3.6).

The matching error minimization calculates the optimal rigid transformation $(\mathbf{R}^{(i)}, \mathbf{t}^{(i)})$. This leads to the matching error $e^{(i)}$ with $e^{(i)} \le d^{(i)}$ by definition of (3.9). If $e^{(i)} > d^{(i)}$ would be true, then the identity transformation would yield a smaller error than the least-square minimization which is impossible.

The next iteration of the ICP algorithm starts again with the closest point search generating the new couplings $C^{(i+1)}$. By definition of the closest point search, the new couplings $C^{(i+1)}$ are closer or equal to the previous ones in $C^{(i)}$ and therefore $d^{(i+1)} \le e^{(i)}$ must be true. The least-square minimization yields again $e^{(i+1)} \le d^{(i+1)}$.

Finally, the statement

$$0 \le e^{(i+1)} \le d^{(i+1)} \le e^{(i+1)} \le d^{(i+1)} \tag{3.11}$$

is true. It ensures a non-increasing and bounded matching error. The ICP algorithm therefore converges monotonically to a (in general local) minimum. Its behavior is necessarily affected by a number of factors and some of them are investigated in Section 3.7.

3.3.4 Enhancements

The following sections describe some of the contributions of this work. Methods to integrate several features which enhance the correspondence establishment function $\hat{c}$ are discussed in Section 3.4. The closed-form solution for the matching error minimization is presented in Section 3.5. Finally, different iteration termination criteria are proposed in Section 3.6.

3.4 Closest point search

The closest point search is the first step of the ICP algorithm. It establishes correspondences between two surfaces at every iteration. To every point of the surface P a closest point of the surface X is assigned. Different methods which improve the quality and the speed of the closest point search are discussed in this section.

1) Section 3.4.1 discusses special matching cases where the surface P does not completely match X.

2) The better the calculated closest points reflect the real correspondences, the faster the ICP algorithm will converge, since once the best point correspondences are found, the matching error will not change any more. As the geometric distance is sometimes not sufficient to establish good correspondences, Section 3.4.2 proposes the integration of further features into the closest point search such as surface orientation and color in order to better distinguish the different points.

3) The closest point search is expensive in computation time. Section 3.4.3 presents methods to avoid an exhaustive search over all points of X and to accelerate the closest point search considerably.

4) Until now the surfaces used in the ICP algorithm have been represented by a set of 3D points. Actually, any geometric primitive for which a closest point distance definition exists could be used. Section 3.4.5 shows how the point to triangle distance can be calculated in an efficient way and how color texture can be integrated into the closest point search.

3.4.1 Surface dispositions for the closest point search

Given two surfaces, the ICP algorithm behaves differently if the role of P and X is assigned to one or the other surface. The matching result will not be the same since the closest point couplings are different. The ICP algorithm is not symmetrical in this sense. Different surface dispositions and their impact on the ICP algorithm behavior are discussed here.

If P is a subset of the points of X the coupling error of the correct matching is zero. However, usually the surface discretization of two surfaces differs and even when the surface P matches a subarea of the surface X the points do not coincide. Therefore, the error of the correct matching is small but not zero.

A potential closest point in X must exist for every point in P else the correspondence estimate $\hat{c}$ of (3.5) does not apply. This property is satisfied for the case of the object recognition application where P represents a partial region of the surface X.

The digitizing and inspection applications are different. Here, the surface P contains points, which do not match any point in X. For example, the object digitizing task matches surfaces representing object views that only partially overlap. Points in P which do not overlap X have therefore to be excluded from the error minimization. A method to solve this problem is presented in Section 3.5.3.

Figure 3.3 visualizes the discussed cases for the different applications. The digitizing and inspection application differ in the size of the surface P. Both surfaces have about the same size in the digitizing application where P may represent a complete scene in the inspection task and is usually much larger than X which represents one object present in the scene.

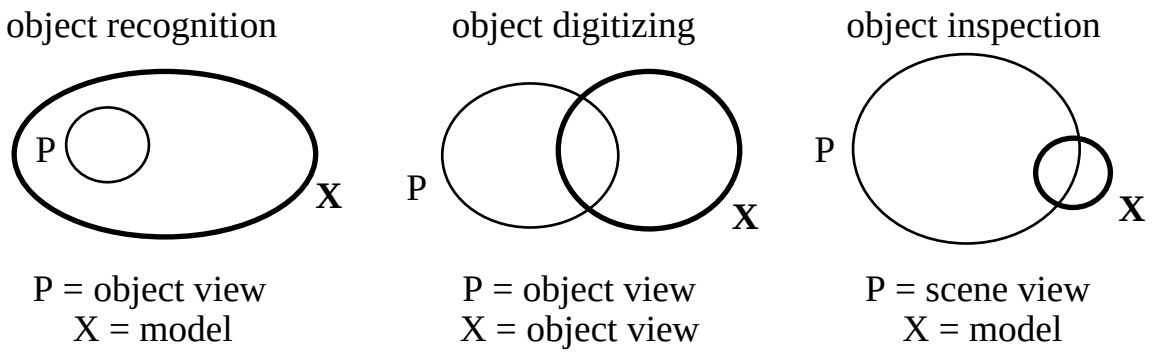


Figure 3.3 Different surface dispositions for the closest point search

The association of P and X to the two matched surfaces is defined for the object recognition and inspection application as shown in Figure 3.3. In the object digitizing application, the roles of P and X are interchangeable since the surface setup is symmetric.

3.4.2 Closest point features

The closest point search of the standard ICP algorithm uses the square Euclidean distance of the Cartesian 3D coordinates as defined in (3.6) and called geometric feature in this work. This geometric distance is not always sufficient to obtain successful matching for the ICP algorithm. For example, the matching of objects like a ball or a cylinder is not defined since two views of such objects do not differ in shape and therefore no unique matching solution exists. The geometry of two surfaces can match but the coupled points may differ in color or orientation.

Better correspondences are found if the surface representations contain more discriminative characteristics. Barequet and Sharir [BAR] assign a feature vector called footprint to every sensed point. Such a feature vector may contain data available from other sensors such as color or calculated features for example surface normal, curvature or multi dimensional characteristics such as spin images [JOHb].

Various criteria apply for the choice of additional closest point features:

1) The quality of a feature depends not only on its potential to distinguish different surface parts, but also on the time needed to calculate the dissimilarity of two different features.

2) Furthermore, the calculated or sensed characteristics should be insensitive to noise since corresponding points on different shapes should have the same feature values.

3) Finally, the different feature distances have to be integrated in one total coupling distance. This requires a normalization of the different distance values since they do not necessarily have the same value range. The distance normalization is necessary to level out the influence of the different features in the total coupling distance.

In this work, color and surface orientation information are integrated into the coupling distance d of (3.6). The following sections discuss the different distance definitions and their normalization.

Color

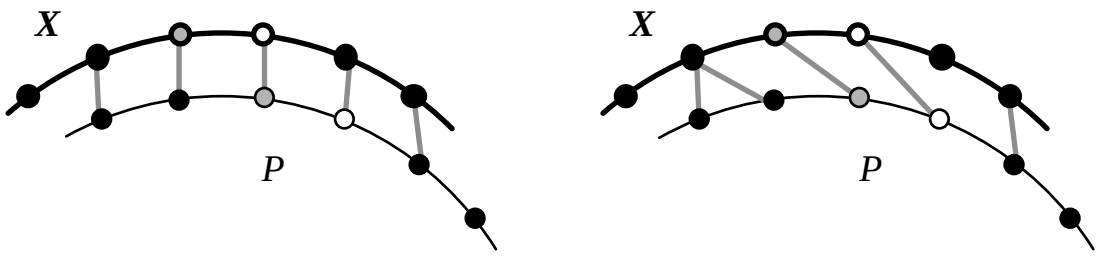
Several authors report encouraging results for modified ICP algorithms with the color feature integrated into the closest point search [JOHa] [MAR] [GODb]. Surface color helps to avoid ambiguous cases where the surface geometry is not sufficient for a successful matching; this assumes however, that the object surface is of varying color.

Colored range images are available for some 3D scanners where the surface geometry information is registered with the corresponding surface color for every point (see Section 2.4.2).

In order to be able to define a simple distance measure, the color feature is represented by the three color components red, green and blue which build the feature vector $\mathbf{c} = (r,g,b)$. There is no need to use perceptually-based color systems since the coupling distance is an objective measure. The different color components r, g and b have the same range as for example from 0 to 255. The coupling distance between two colors on P and X is defined by the square Euclidean distance of the color vectors:

$$d_c(\mathbf{c}_p, \mathbf{c}_x) = \left\| \mathbf{c}_p - \mathbf{c}_x \right\|^2 \tag{3.12}$$

Figure 3.4 shows the matching of two arcs with different colored surface points. Correct couplings are established if the color feature is integrated into the coupling distance.



without color with color

Figure 3.4 Coupling improvement with color feature

Surface orientation

Several interesting features can be derived from the local surface geometry. Here follows a list presenting three possible features:

Normal vector: Most frequently used and not very sensitive to noise since its calculation implies only first-order derivatives [BAR] [FELa] [STOa].

Curvatures: Insensitive to rigid transformations and scale but need advanced data filtering to remove noise as second-order derivatives are implied in the calculation [GODa].

Spin image: Two-dimensional histogram of the polar coordinates of all surface points. The polar coordinates are measured relatively to a tangential plane going through a selected surface point. This feature allows good correspondences to be established. However, feature matching is a time expensive correlation operation [JOHb].

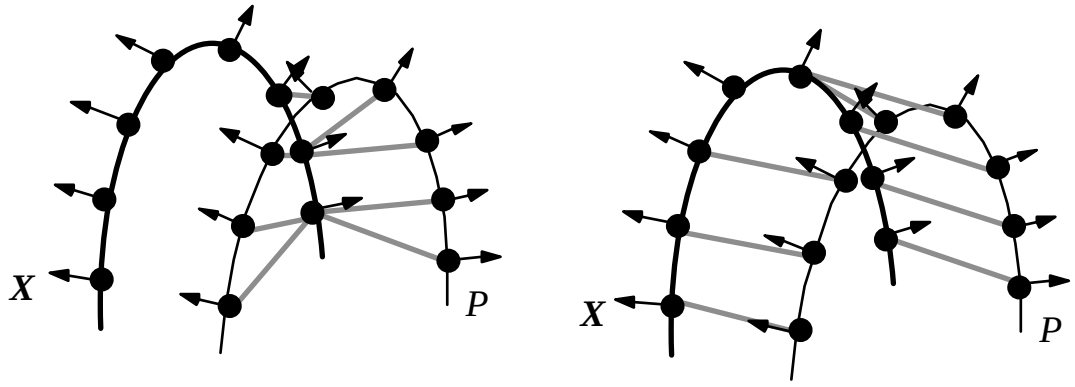
In this work, the surface normal vector has been chosen for the following reasons. It is low sensitive to noise and the difference between two normal vectors can be calculated easily.

The surface orientation feature vector is defined by the surface normal coordinates $\mathbf{n}$ = (u,v,w). The coupling distance between two orientation vectors on *P* and *X* is therefore defined as follows:

$$d_n(\mathbf{n}_p, \mathbf{n}_x) = \|\mathbf{n}_p - \mathbf{n}_x\|^2 \text{ with } \|\mathbf{n}_p\| = \|\mathbf{n}_x\| = 1 \qquad (3.13)$$

Of course, two parameters as for example two angles are sufficient to define the orientation of a normal vector. Nevertheless, the Euclidean distance of two surface normal vectors of unit length reflect the difference in orientation better since there is no angle modulo problem.

Figure 3.5 shows an example where the surface orientation feature helps to avoid a local minimum of the ICP matching.



without orientation with orientation

Figure 3.5 Coupling improvement with surface orientation feature

Combination of multiple features

Different approaches exist with which to consider several features in the closest point search of the ICP algorithm.

Godin finds the closest point by separately minimizing every feature distance [GODa], [GODb]. First, the subset of the closest points which have a color distance below a certain threshold is built. Then, among these candidates the point with minimal geometric distance is chosen as the final closest point. This decoupled distance minimization introduces considerable effort since the closest point search is performed several times.

In this work, a single distance measure, which combines all features, is selected. This allows the closest point search to be performed in one step. It permits also a simple integration of acceleration methods (see Section 3.4.3).

The selected features are surface geometry **g**, normal **n** and color **c**. The different features are integrated into one feature vector. Previously **p** and **x** contained the Cartesian coordinates of the surface points which are now represented by the geometric feature vector **g**. The original point vectors of *P* and *X* are extended as follows:

$$\begin{aligned}\mathbf{p} &= \left(\mathbf{g}_p, \mathbf{n}_p, \mathbf{c}_p\right) = \left(x_p, y_p, z_p, u_p, v_p, w_p, r_p, g_p, b_p\right) \\ \mathbf{x} &= \left(\mathbf{g}_x, \mathbf{n}_x, \mathbf{c}_x\right) = \left(x_x, y_x, z_x, u_x, v_x, w_x, r_x, g_x, b_x\right)\end{aligned} \tag{3.14}$$

This leads to a new formulation of the coupling distance in (3.6) as follows

$$d(\mathbf{p},\mathbf{x}) = \frac{1}{\alpha_g} d_g(\mathbf{g}_p, \mathbf{g}_x) + \frac{1}{\alpha_n} d_n(\mathbf{n}_p, \mathbf{n}_x) + \frac{1}{\alpha_c} d_c(\mathbf{c}_p, \mathbf{c}_x) \text{ with}$$

$$d_g(\mathbf{g}_p, \mathbf{g}_x) = \|\mathbf{g}_p - \mathbf{g}_x\|^2, \quad d_n(\mathbf{n}_p, \mathbf{n}_x) = \|\mathbf{n}_p - \mathbf{n}_x\|^2 \tag{3.15}$$

$$\text{and} \quad d_c(\mathbf{c}_p, \mathbf{c}_x) = \|\mathbf{c}_p - \mathbf{c}_x\|^2$$

where α_g, α_n and α_c represent the weights used to normalize the different feature value ranges.

Since the different features used for the total coupling distance do not have the same value range, there is a need to normalize the different feature distances before they are added. Otherwise, one of these features may dominate the other in the closest point distance. The selection of an appropriate size for the normalization factor is discussed in Section 5.7.

The coupling distance d in (3.15) can be any combination of d_g, d_n and d_c but practically d_g should always be included since the ICP algorithm is based on it (see (3.5)). Experiments in Section 5.8 show that the surface orientation or color alone are not discriminative enough to estimate the correspondence function c (see Section 5.8 for an exception).

3.4.3 Fast closest point search

The closest point search is the most expensive step in time of the ICP algorithm. Having N points in P and M points in X the resulting complexity is $N \cdot M$ since the coupling distance d is calculated for every point in P and for all the points in X.

The magnitude of N and M is usually several thousand points; this is necessary in order to cover all details of a surface sufficiently. Therefore, the calculation of the closest point couplings includes millions of computer operations and may last some seconds even on a fast workstation.

The following sections present methods to reduce the closest point calculation complexity of $N \cdot M$ for the worst case represented by the exhaustive search.

Subsampling of P and X

A simple method to reduce the complexity of the closest point search is to subsample the points of P and X. A straightforward approach is to perform a linear subsampling of the point sets.

More elaborate methods use adaptive subsampling which keeps few points in flat surface areas and more points when the geometry varies. Some authors use curvatures [GAR] or surface smoothness criteria [CHE] [SOUa] to select the points to be removed. These methods allow the significant compression of the number of points needed to represent a surface geometry. However, the distance between the remaining points may be locally large and the geometric point matching fails to establish reliable closest point couplings.

To avoid this problem, a linear subsampling method which works in the range image space is proposed. The point reduction is performed using the range image structure as shown in Section 2.5. Points that are close in the range image are in general also close in 3D. Therefore, a subsampling of the range image ensures that the resulting points are distributed homogeneously over the object surface. The subsampling distance, which defines the mean distance between two surface points, is proportional to the range image subsampling factor and fixes an upper limit for the closest point distance when both surfaces are matched.

However, details smaller than the subsampling distance get lost. Multi-resolution approaches such as those presented in Section 6.3 provide an escape for this problem.

Search structures

The closest point search can be fast even for large data sets if special structures are used to organize the data. Different search structures such as trees, projections and clusters are summarized in [FRI]. Here, two methods are discussed:

Distance maps	Distance maps voxelize the feature space [DAN] and allow direct access to the closest points. They are however not suited for high dimensional vectors as defined in (3.14) since the memory requirement for the map explodes.
kD tree	A kD tree is a binary tree used to search data with k keys. Each non-terminal node represents a

partitioning of the data set according to one key. The memory requirement for this structure grows linearly with the number of points and is independent of the number of used features.

Since the aim of this work is to integrate several features in the closest point search, the kD tree algorithm is selected. Several authors proposed this structure to implement a fast closest point search for the ICP algorithm [BESb] [SIM] [ZHA]. A novel aspect of this work is the integration of three different features in a kD tree search.

kD tree construction

The construction of a kD tree is shown using the 2D data point set of Figure 3.6. Point A is selected as the root of the kD tree. Every level of the kD tree divides the feature space according to one of the feature space dimensions. Usually, the dimensions are used in a cyclic order. Therefore, point B is inserted as left child of A since it has a smaller x coordinate than A. B now divides the feature space in y direction. The construction continues iteratively by inserting one point after the other. At every node, the new point falls on either side of the hyperplane defined by the key used as discriminator at the current node. If the node has no child on the side where the new point falls, the new point is introduced as child. Figure 3.6 presents the final kD tree and the feature space partitioning for an example point set in 2D.

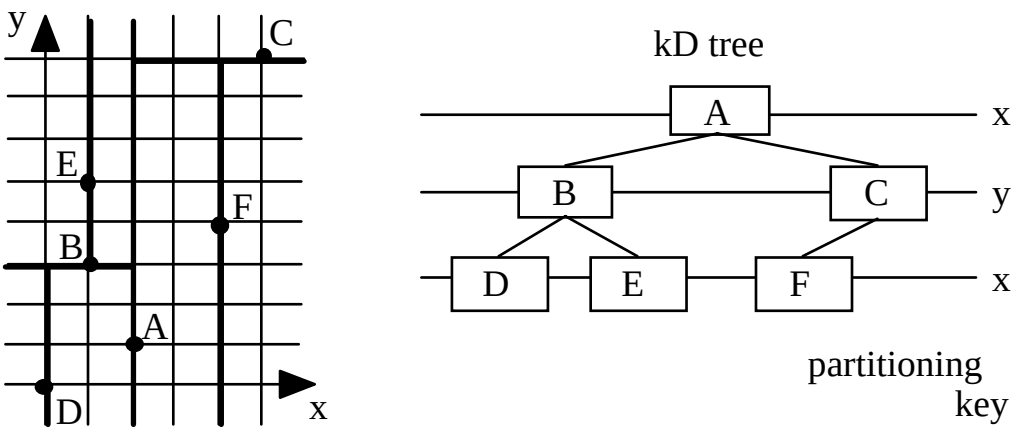


Figure 3.6 Example of a 2D tree construction

Since each non-terminal node represents a partitioning of the data set according to one key, it allows the exclusion of one side of the hyperplane separating the data space during the closest point search. Therefore, the better the kD tree is balanced the faster is the closest point search.

The following factors influence the construction of the kD tree: the selection of the next point inserted in the kD tree and the key used to separate the feature space at the node where the point is inserted. Here, the partitioning key is obtained by cycling through the keys in order. An alternative is to select the partitioning key at each node in a more flexible way. An interesting choice is to use the key which at each node provides the largest spread of values and to use the median of the key as discriminator value [FRI]. This leads to a balanced binary tree and a theoretically optimal search performance.

Closest point kD tree search

Recursive algorithms that use the kD tree for the closest point search have been proposed in [FRI] and [ZHA]. The principal idea is to observe all points in the kD tree of X, which fall inside a k dimensional sphere centered on the point **p** for which the closest point is searched.

At the initialization, the sphere radius $D_{\max}$ is set to D_{init} and the current node is the root of the kD tree. If the sphere overlaps the hyperplane defined by the partitioning key of the current node, the search algorithm is called recursively for both children, otherwise only for the one lying on the side of the point **p**. Furthermore, if the current node to point **p** distance is smaller than $D_{\max}$, $D_{\max}$ is reduced to this distance and the intermediate closest point is set to the current node. The recursive search stops if a node has no children. Finally, the intermediate closest point corresponds to the correct closest point and $D_{\max}$ is equal to the closest point distance.

The following procedure formulates the above recursive search algorithm:

- input: a point **p**, a kD tree T of a point set X, an intermediate closest point $\mathbf{y} = root$ and $D_{\max} = D_{\text{init}}$
- output: the closest point **y** and the corresponding distance $D_{\max}$
- procedure: search(*node*)

if(*node* == NIL) return;

dim = get_partitioning_key(*node*);

$d_{dim} = \mathbf{p}[dim] - X[node][dim]$;

if($|d_{dim}| < D_{\max}$) then

$\mathbf{y} = X[node]$, $D_{\max} = d(\mathbf{p}, X[node])$;

if($\mathbf{p}[dim] - D_{\max} < X[node][dim]$) then

search(leftchild(*node*));

if($\mathbf{p}[dim] + D_{\max} \geq X[node][dim]$) then

search(rightchild(*node*));

The expected search time complexity for the above routine is of $O(\log(M))$ with M the number of points in X [FRI]. The worst case complexity is of $O(M^{2/3})$ [ZHA].

3.4.4 kD tree search performance

In order to assess the performance of the recursive search routine using a kD tree, the following experiment is performed. A closest point search is done for 7000 points of a surface P and a varying number of points representing a second surface X. The calculation is performed on a SGI Indigo2 R10000 workstation and the results are shown in Figure 3.7.

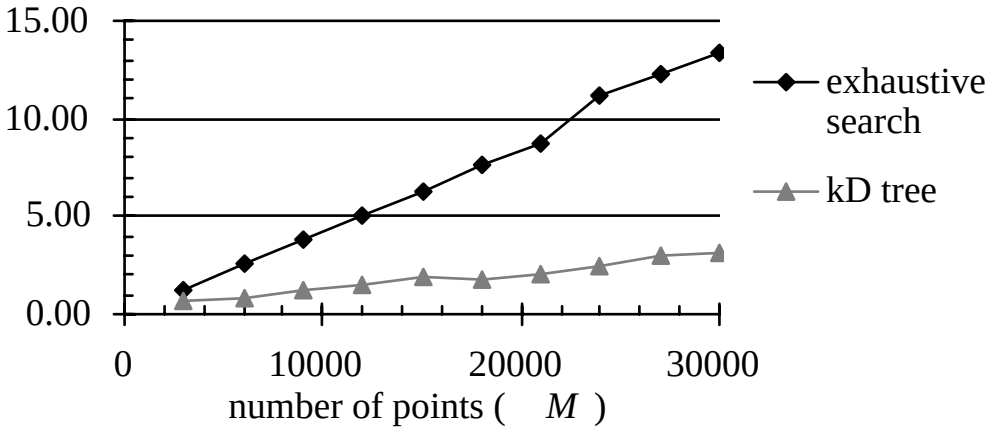


Figure 3.7 Search time using a kD tree

The obtained results show that the kD tree provides as expected a significant reduction of the computation time of the closest point search compared to the exhaustive search.

Performance factors

Different parameters allow further improvement to the performance of the recursive search routine. First, the construction of the kD tree is of major importance. The more balanced the kD tree is, the faster the closest point search is.

The estimate of the size of D_{init} is a second important factor. The smaller D_{init} the faster the recursive search since fewer branches are inspected. In a general case D_{init} is set to infinity which is not very efficient. Section 5.6.1 shows that a priori knowledge allows the setting of D_{init} to a smaller upper bound.

Furthermore, the closest point of the previous search done for $\mathbf{p}_{k-1}$ can be used to estimate D_{init} for the next closest point search of $\mathbf{p}_k$. Since most 3D scanners store the data points in the scanning order, consecutive points are close in 3D space and are likely to have close closest points. This method corresponds to a spatial closest point cache. A temporal closest point cache where the closest point of the previous iteration is used to estimate D_{init} is proposed in [SIM].

Limitations

The closest point search is most efficient if the data of *P* is inside the volume spanned by the data of *X* for which the kD tree is constructed. If the two shapes are far apart, which may be the case for the first iterations of the geometric point matching, the search time increases. This fact is measured in the following experiment. The *P* data is successively moved away from the *X* data and a closest point search is performed for every configuration. The results are presented in Figure 3.8.

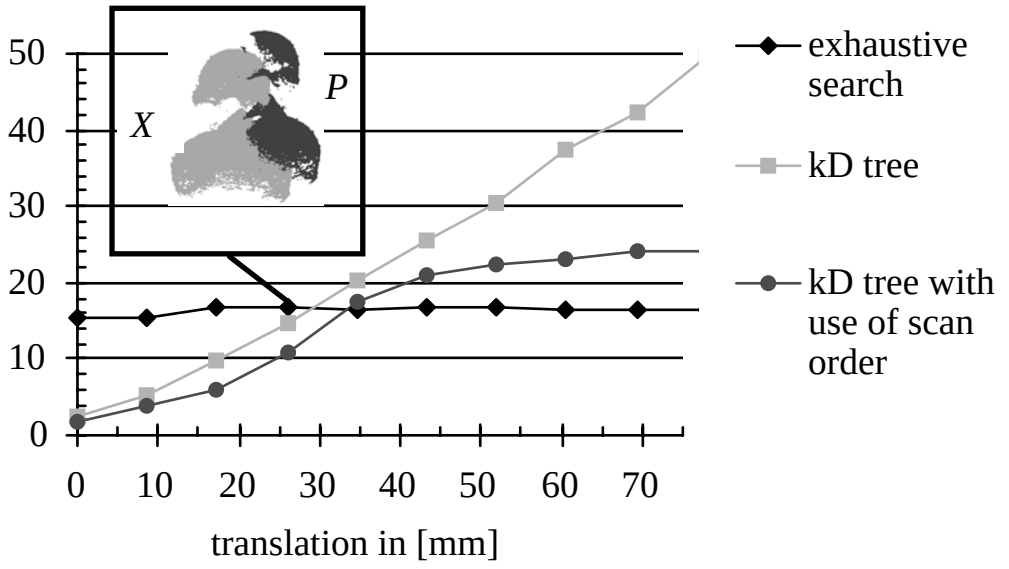


Figure 3.8 Limits of kD tree search routine

As soon as the two shapes do not overlap any more which is the case for a translation of more than 20 mm for this object, the performance of the recursive search routine degrades. This increase of the search time at larger distances is explained by the fact that all coupling distances are of similar length. The discrimination power of single keys falls short of separate points with similar Euclidean distances.

A solution to limit this problem is to use the scan order of the points in P as proposed above using the closest point found in the previous search to estimate the next D_{init} .

Note however that this unfavorable case appears very seldom in the ICP algorithm because a possible large distance disappears after the first iteration. Then, the surfaces P and X overlap and the kD tree can work optimally.

3.4.5 Closest point on a continuous surface

So far, the surfaces P and X are represented by sets of points, which result from a discretization of the object surface. To insure precise registration dense surface sampling is needed since the sampling grids of different acquisitions do not necessarily correspond and the closest points do not necessarily overlap. Even after several iterations, a matching error remains in the order of the sampling grid distance.

This final error is much smaller (usually in the order of the scanner resolution) if the object surface is represented continuously by two-dimensional primitives (see Figure 3.9). Such primitives cover the whole surface area.

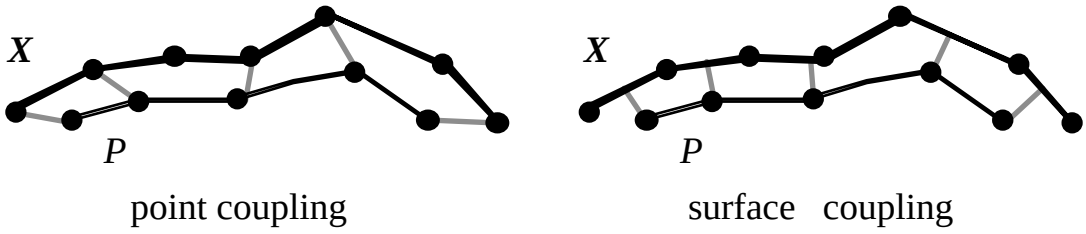


Figure 3.9 Smaller matching error with continuous surface coupling

In fact, the surface *X* in the ICP algorithm can be of any representation and the algorithm remains valid as long as a point to primitive distance definition exists to replace (3.6) in the closest point search [BESb].

Section 2.3 motivates the selection of triangle meshes to represent free-form surfaces. The following sections present methods to calculate efficiently the distance from a point to a triangle. The closest point search calculates this distance for a point of *P* for every triangle of *X* and selects the triangle of *X* with the smallest distance. A method based on the kD tree is presented to accelerate the calculation of the distance from a point to a triangle mesh.

Point to triangle distance

The distance of a point $\mathbf{p}$ to a triangle $\mathbf{t}_x = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$ can be defined in an implicit form as follows:

$$d(\mathbf{p}, \mathbf{t}_x) = \min_{a,b} \left\| \mathbf{p} - \left(\mathbf{x}_1 + a(\mathbf{x}_2 - \mathbf{x}_1) + b(\mathbf{x}_3 - \mathbf{x}_1)\right) \right\|^2 \quad \text{with} \quad a \geq 0 \cap b \geq 0 \cap a + b \leq 1 \tag{3.16}$$

This distance definition replaces the definition (3.6) in the search for the closest point $\mathbf{y}$ performed according to (3.7).

Since there is not a direct solution to evaluate the distance of (3.16), the minimization over the factors *a* and *b* is decomposed into three steps. Actually, the evaluation of (3.16) corresponds to the search for the point $\mathbf{x}$ on the triangle $\mathbf{t}_x$ which is closest to $\mathbf{p}$. The point $\mathbf{x}$ can be inside the

triangle, on a triangle edge or on one of the triangle vertices. These three cases are checked as follows:

1) If the projection $\mathbf{p}_t$ of $\mathbf{p}$ onto the plane defined by the triangle vertices is inside the triangle, then $\mathbf{x} = \mathbf{p}_t$, else continue:

2) If one of the projections $\mathbf{p}_e$ of $\mathbf{p}$ on one of the triangle border rays is inside a triangle edge, then $\mathbf{x} = \mathbf{p}_e$ taking the closest one, else continue:

3) Use the closest triangle vertex $\mathbf{x}_i$ as point $\mathbf{x}$.

The different conditions are implemented as follows (see also Figure 3.10):

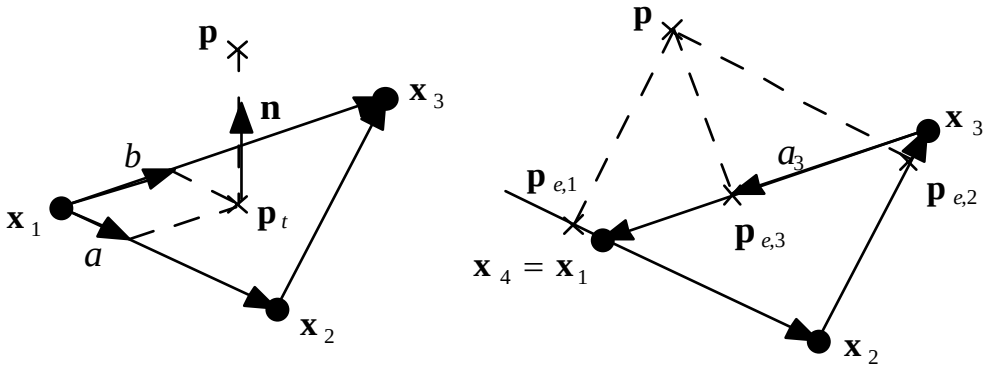


Figure 3.10 Closest point to triangle distances

1) The projection $\mathbf{p}_t$ of $\mathbf{p}$ is inside the triangle if the following condition is satisfied:

$$
\begin{aligned}
&a \geq 0 \cap b \geq 0 \cap a + b \leq 1 \text{ with} \\
&a = \frac{(\mathbf{p} - \mathbf{x}_1) \cdot (\mathbf{n} \times (\mathbf{x}_3 - \mathbf{x}_1))}{(\mathbf{x}_2 - \mathbf{x}_1) \cdot (\mathbf{n} \times (\mathbf{x}_3 - \mathbf{x}_1))} \\
&b = \frac{(\mathbf{p} - \mathbf{x}_1) \cdot (\mathbf{n} \times (\mathbf{x}_2 - \mathbf{x}_1))}{(\mathbf{x}_3 - \mathbf{x}_1) \cdot (\mathbf{n} \times (\mathbf{x}_2 - \mathbf{x}_1))} \\
&\mathbf{p}_t = \mathbf{x}_1 + a(\mathbf{x}_2 - \mathbf{x}_1) + b(\mathbf{x}_3 - \mathbf{x}_1)
\end{aligned}
\tag{3.17}
$$

2) The projection $\mathbf{p}_e$ is on an edge if:

$$\begin{aligned} &0 \le a_l \le 1 \text{ with } \; l \in [1,3] \\ &a_l = \frac{(\mathbf{x}_{l+1} - \mathbf{x}_l)\cdot(\mathbf{p} - \mathbf{x}_1)}{\|\mathbf{x}_{l+1} - \mathbf{x}_l\|^2} \\ &\mathbf{p}_{e,l} = \mathbf{x}_l + a_l(\mathbf{x}_{l+1} - \mathbf{x}_l) \end{aligned} \tag{3.18}$$

If the coupling distances include the normal and/or the color feature, linear interpolation can again be used to obtain these feature values for the point **x**. The factors *a* and *b* calculated as above can be used to weight the vertex feature vectors in order to approximate the normal or color vector for **x**. However, the point **x** is not necessarily closest in color or surface orientation.

The Lagrange technique finds the point **x** inside a triangle (step 1) which minimizes the sum of all feature distances as defined in (3.15) at once. The triangle vertices include all features in one vector and the following distance has to be minimized.

$$\begin{aligned} &\mathbf{t}_x = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\} \text{ with } \; \mathbf{x}_l = (\mathbf{g}_{x,l}, \mathbf{n}_{x,l}, \mathbf{c}_{x,l}) \\ &\mathbf{x} = (a\mathbf{x}_1 + b\mathbf{x}_2 + c\mathbf{x}_3) \text{ and } \; \mathbf{v} = \mathbf{p} - \mathbf{x} \\ &d = \min_{a,b,c} \|\mathbf{v}\|^2 = \min_{a,b,c} (\mathbf{v}\cdot\mathbf{v}) \text{ with } \; a + b + c = 1 \end{aligned} \tag{3.19}$$

Applying the technique of Lagrange multipliers, this minimization problem is equivalent to solve

$$\begin{aligned} &\text{find } \min_{a,b,c} L = [\mathbf{v}\cdot\mathbf{v} + \lambda(a + b + c - 1)] \\ &\text{subject to the constraint} \qquad a + b + c = 1. \end{aligned} \tag{3.20}$$

This is solved by setting the partial derivatives of the criterion *L* to zero giving the following system of equations for *a*, *b* and *c*:

$$\begin{aligned} &a + b + c - 1 = 0 \\ &\frac{\partial L}{\partial a} = 0, \frac{\partial L}{\partial b} = 0, \frac{\partial L}{\partial c} = 0 \end{aligned} \tag{3.21}$$

If all factors *a*, *b* and *c* are in the range [0, 1], the point **x** is inside the triangle and can be calculated with (3.19). If this is not the case, the same steps 2 and 3 of the above routine have to be performed. The point **x** may lie on a triangle edge (check condition (3.18) with extended feature vectors) or correspond to the feature vector of a triangle vertex.

Fast calculation of closest point on triangle mesh

This section presents a fast closest point to triangle search. As proposed by Simon in [SIM], a kD tree is used to select only a few triangles in which the closest point is searched for. The basic idea is to first find the closest vertex on *X*. This search can be done efficiently with a kD tree. Then, the more complex point to triangle distance is calculated only for the triangles attached to the closest triangle vertex.

This approach assumes that the object surface is sufficiently densely sampled to ensure that the triangle with the correct closest point is attached to the closest triangle vertex. Critical regions are thin convex or concave object parts where this assumption may not hold (see Figure 3.11).

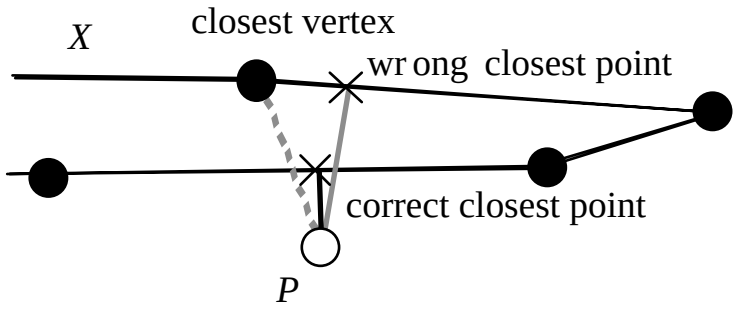


Figure 3.11 Limitations of the fast closest point on triangle mesh method

3.4.6 Closest point on a textured surface

Textured triangle meshes provide realistic object rendering even for sparsely sampled surfaces (see Section 2.5). Apart from visualization, texture images can also enhance the surface registration quality. The following method describes a closest point search based on texture image correlation.

The basic idea is to project the color texture of two 3D surfaces onto an image plane. 2D image correlation matches subregions of the two texture images. The displacement vectors calculated in the image plane allows the 3D surface registration to be updated for small displacements.

As the surface texture of both surfaces has to be projected onto the same image plane, a previous rough registration of the two surfaces is necessary. This registration task can be performed by the ICP algorithm. Now, the surface textures of the overlapping parts can be projected and correlated in a common image plane. Since the image correlation is sensitive to scale and rotation, the correlation of the complete two texture images will fail in most cases. A better approach is to correlate small subregions of the textures.

This work proposes to correlate several small texture windows. A texture window is attached to a closest point coupling pair. The basic idea is to correlate the texture window attached to **p** with the texture around the closest point **y** in order to update the position of **y**. The texture images of the surfaces P and X have in general not the same scale and orientation because they have been acquired from different points of view. If the texture image of the surface P is transformed such that it looks as it would have been taken from the same view point as the texture image of X then the image correlation can work successfully. The closest point couplings of the ICP algorithm help to perform this image transformation and to calculate the corresponding warping function.

Once the texture images are in the same reference frame, the texture windows of the closest points are correlated and the position of the closest point **y** is updated in the image space. The resulting 2D displacement vector allows the position of **y** in the 3D space to be updated. The modified closest points are used in the ICP matching error minimization to find a new optimal rigid transformation.

The following points describe the closest point search using texture images in more details:

1) Since the image correlation is a computation intensive task, only the vertices **p** on P which have a closest point on X and a texture of high contrast are selected. A low contrast is not pertinent and is inadequate for correlation. The contrast of the texture window w_t with N_t pixels attached to a vertex **p** is defined as the standard deviation of the grayscale pixel values:

$$s = \sqrt{\frac{1}{N_t}\left(\sum_{k=1}^{N_t} w_{t,p}(k)^2 - N_t \overline{w}_{t,p}{}^2\right)} \text{ with } \overline{w}_{t,p} = \frac{1}{N_t}\sum_{k=1}^{N_t} w_{t,p}(k) \qquad (3.22)$$

2) The texture images of P and X have to be mapped onto the same reference frame before the image correlation can be launched. The P image space is mapped to the X image space with an image transform defined by two 2-nd degree polynomials (one for each spatial

coordinate). This corresponds to an affine transformation. This warping function can be defined by a set of corresponding points in both images. Since the closest point couplings of the 3D shape geometry are available from the ICP algorithm, corresponding pixels on both texture images are known (see Figure 3.12).

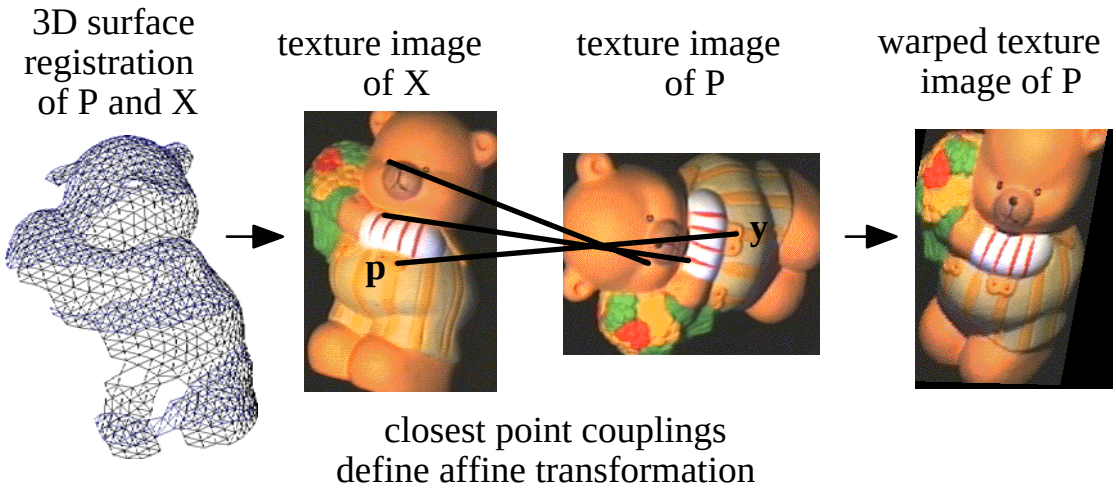


Figure 3.12 Texture image warping using the ICP closest point couplings

3) For every selected vertex **p**, the closest point **y** in X is updated as follows. The texture window centered at a point **p** from the warped P texture image is correlated with texture image of X. The correlation is only calculated in the neighborhood of the closest point **y**. The correlation factor is defined as follows:

$$r = \frac{\sum_{k=1}^{N_t} w_{t,p}(k) w_{t,x}(k) - N_t \overline{w}_{t,p} \overline{w}_{t,x}}{\sqrt{\left(\sum_{k=1}^{N_t} w_{t,p}(k)^2 - N_t \overline{w}_{t,p}{}^2\right)\left(\sum_{k=1}^{N_t} w_{t,x}(k)^2 - N_t \overline{w}_{t,x}{}^2\right)}} \qquad (3.23)$$

$$\text{with} \quad \overline{w}_{t,x} = \frac{1}{N_t} \sum_{k=1}^{N_t} w_{t,x}(k)$$

The closest point **y** in the X image is moved to the location with the highest correlation factor as shown in Figure 3.13.

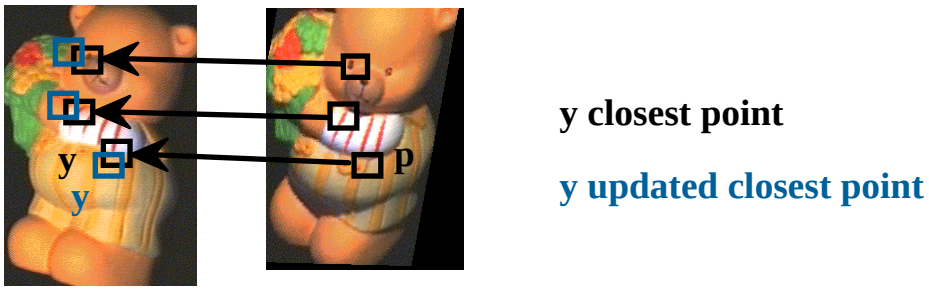


Figure 3.13 Texture image correlation updating the closest point couplings

4) The displacement vector in the texture image space is used to update the closest point **y** in 3D space. The displacement vector is expressed as a linear combination of two triangle edges in the image space. The calculated factors are applied to the triangle edges in 3D space, which assumes that the new closest point lies on the triangle plane. This restriction is acceptable if the object surface is locally smooth and the displacement vector small.

5) The matching error minimization of the ICP algorithm is launched again with the updated closest point couplings.

3.5 Optimal rigid transformation

Given the surfaces *P* and *X* and the point couplings $C^{(i)}(\mathbf{p}_k, \mathbf{y}_k)$ find the rigid transformation that minimizes the matching error $e^{(i)}(\mathbf{R}^{(i)}, \mathbf{t}^{(i)})$ as stated in (3.9). The following sections show how the rotation and translation, which minimize the matching error, can be calculated. The matching error of (3.8) is expanded into different sums. This allows the best translation and rotation to be found independently as proposed in [HOR].

To make the explanations more readable, the iteration index *i* in (3.8) is dropped and the total transformation is set to the identity transformation and therefore $\tilde{\mathbf{p}}_k = \mathbf{p}_k$.

$$e = \sum_{k=1}^{N} \left\| (\mathbf{R}\mathbf{p}_k + \mathbf{t}) - \mathbf{y}_k \right\|^2 \qquad (3.24)$$

Here, the feature vectors $\mathbf{p}_k$ and $\mathbf{y}_k$ contain only the geometric feature **g** representing the Cartesian point coordinates. The integration of other features in the matching error minimization is discussed in Section 3.5.4.

3.5.1 Finding the best translation

The error of (3.24) can be split up into the following sums

$$e = \sum_{k=1}^{N} \left\| \mathbf{R}\mathbf{p}_k - \mathbf{y}_k \right\|^2 + 2\mathbf{t} \cdot \sum_{k=1}^{N} \left(\mathbf{R}\mathbf{p}_k - \mathbf{y}_k \right) + \sum_{k=1}^{N} \left\| \mathbf{t} \right\|^2 . \tag{3.25}$$

The first and the third term depend only on **R** or **t** whereas the second term is mixed. If the mixed term can be set to zero, the error *e* can be minimized successively for **R** and **t** since the other two terms are positive. The mixed term disappears if the data points $\mathbf{p}_k$ and $\mathbf{y}_k$ are referred to their centroids

$$\begin{gathered} \overline{\mathbf{p}} = \frac{1}{N} \sum_{k=1}^{N} \mathbf{p}_k , \quad \overline{\mathbf{y}} = \frac{1}{N} \sum_{k=1}^{N} \mathbf{y}_k \quad . \\ \mathbf{p}' = \mathbf{p} - \overline{\mathbf{p}}, \qquad \mathbf{y}' = \mathbf{y} - \overline{\mathbf{y}} \end{gathered} \tag{3.26}$$

Now the error of (3.24) can be rewritten as

$$\begin{gathered} e = \sum_{k=1}^{N} \left\| \left(\mathbf{R}\mathbf{p}'_k + \mathbf{t}' \right) - \mathbf{y}'_k \right\|^2 \\ \text{with} \quad \mathbf{t}' = \mathbf{R}\overline{\mathbf{p}} + \mathbf{t} - \overline{\mathbf{y}} \end{gathered} \tag{3.27}$$

and expanded into the following sums

$$e = \sum_{k=1}^{N} \left\| \mathbf{R}\mathbf{p}'_k - \mathbf{y}'_k \right\|^2 + 2\mathbf{t}' \cdot \sum_{k=1}^{N} \left(\mathbf{R}\mathbf{p}'_k - \mathbf{y}'_k \right) + \sum_{k=1}^{N} \left\| \mathbf{t}' \right\|^2 . \tag{3.28}$$

The second term which sums up all the data points is now equal to zero since the measurements are referred to their centroids. The remaining first and third term cannot be negative and can therefore be minimized individually. The total error is obviously minimal if the last term is zero which leads to the best translation vector.

$$\mathbf{t} = \overline{\mathbf{y}} - \mathbf{R}\overline{\mathbf{p}} \tag{3.29}$$

The first term in (3.28) is the last one not yet equal to zero and depends only on **R**. The best rotation is therefore the one which minimizes the term:

$$e = \sum_{k=1}^{N} \left\| \mathbf{R}\mathbf{p}'_k - \mathbf{y}'_k \right\|^2 . \tag{3.30}$$

Once the rotation matrix is obtained from the minimization of (3.30), the translation vector is easily calculated with (3.29), which corresponds to the difference of the centroid of $\mathbf{y}_k$ and the rotated centroid of $\mathbf{p}_k$.

3.5.2 Finding the best rotation

Several methods to find the rotation matrix **R** exist which minimize the matching error of (3.30). Among them, the ones having a closed-form are the most interesting (see Zhang for an extensive list [ZHA]).

One of the fastest implementations is the singular value decomposition (SVD) algorithm proposed by [ARU]. Here, the rotation matrix **R** is equal to the multiplication of the two orthonormal matrices obtained from the singular value decomposition of the following matrix

$$\mathbf{H} = \sum_{k=1}^{N} \mathbf{p}'_k \mathbf{y}'^{T}_k \,. \tag{3.31}$$

The SVD method has the disadvantage of the resulting matrix possibly representing a rotation or a reflection (mirroring).

Faugeras proposed the use of unit-quaternions to solve (3.24) for **R** and **t** [FAU]. Unit-quaternions have several advantages. The resulting matrix is always a rotation matrix. It is much simpler to enforce the constraint that a quaternion is of unit magnitude than to ensure that a matrix is orthonormal. In this work, a combination of the work of Faugeras [FAU] and Horn [HOR] is implemented.

Quaternions

Several authors used quaternions for computer vision tasks and give a good introduction and definition of their calculus [FAU] [HOR] [AMA].

A quaternion consists of a vector with four components. Quaternions can be interpreted as complex numbers where the first component is a scalar and the second component an imaginary vector as stated in (3.32). Quaternions are denoted as vectors with dots above them.

$$\dot{\mathbf{q}} = (q_0, q_x, q_y, q_z) = (q_0; \mathbf{q}) \tag{3.32}$$

A real number s is identified by the quaternion (s, **0**) and a real vector **v** by the purely imaginary quaternion (0, **v**). The conjugate $\dot{\mathbf{q}}^*$ of a quaternion $\dot{\mathbf{q}}$ is defined similar to the one for complex numbers:

$$\dot{\mathbf{q}}^* = (q_0; -\mathbf{q}). \tag{3.33}$$

The product of two quaternions $\dot{\mathbf{q}}$ and $\dot{\mathbf{r}}$ is defined as follows (· dot product, × cross product)

$$\begin{aligned}\dot{\mathbf{q}}\otimes\dot{\mathbf{r}} &= \left(q_0 r_0 - \mathbf{q}\cdot\mathbf{r}, q_0\mathbf{r} + r_0\mathbf{q} + \mathbf{q}\times\mathbf{r}\right)\\ &= \left(q_0 r_0 - q_x r_x - q_y r_y - q_z r_z, q_0 r_x + q_x r_0 + q_y r_z - q_z r_y,\right.\\ &\quad \left. q_0 r_y - q_x r_z + q_y r_0 + q_z r_x, q_0 r_z + q_x r_y - q_y r_x + q_z r_0\right)\end{aligned} \tag{3.34}$$

and can also be expressed in terms of the product of an orthogonal matrix with a quaternion vector. Note that the quaternion multiplication is not commutative, this leads to two different matrices depending if the first or second factor is expanded in a matrix.

$$\begin{aligned}\dot{\mathbf{q}}\otimes\dot{\mathbf{r}} &= \begin{bmatrix} q_0 & -q_x & -q_y & -q_z \\ q_x & q_0 & -q_z & q_y \\ q_y & q_z & q_0 & -q_x \\ q_z & -q_y & q_x & q_0 \end{bmatrix}\dot{\mathbf{r}} = \mathbf{Q}\dot{\mathbf{r}},\\ \dot{\mathbf{r}}\otimes\dot{\mathbf{q}} &= \begin{bmatrix} q_0 & -q_x & -q_y & -q_z \\ q_x & q_0 & q_z & -q_y \\ q_y & -q_z & q_0 & q_x \\ q_z & q_y & -q_x & q_0 \end{bmatrix}\dot{\mathbf{r}} = \overline{\mathbf{Q}}\dot{\mathbf{r}} \text{ and}\\ \dot{\mathbf{r}}\otimes\dot{\mathbf{q}}^* &= \overline{\mathbf{Q}}^T\dot{\mathbf{r}}\end{aligned} \tag{3.35}$$

The definition of the magnitude of a quaternion is

$$\begin{aligned}|\dot{\mathbf{q}}|^2 &= \dot{\mathbf{q}}^*\otimes\dot{\mathbf{q}} = \dot{\mathbf{q}}\otimes\dot{\mathbf{q}}^*\\ &= \left(q_0{}^2 + \|\mathbf{q}\|^2, \mathbf{0}\right) = \left(\|\dot{\mathbf{q}}\|^2, \mathbf{0}\right) = (\dot{\mathbf{q}}\cdot\dot{\mathbf{q}}, \mathbf{0})\end{aligned} \tag{3.36}$$

where $|$ represents the quaternion magnitude, $\|$ the usual Euclidean norm and $\cdot$ the dot product.

Rotations and quaternions

As stated before, a data vector can be represented by a purely imaginary quaternion. So, a rotation of a data vector can be represented by an unit-quaternion if there is a way to transform purely imaginary quaternions to purely imaginary quaternions. The next development shows that the composite product

$$\begin{aligned}\dot{\mathbf{r}}' &= \dot{\mathbf{q}}\otimes\dot{\mathbf{r}}\otimes\dot{\mathbf{q}}^*\\ \text{with } \dot{\mathbf{r}} &= (0, r_x, r_y, r_z)\end{aligned} \tag{3.37}$$

is purely imaginary where **r** is the data point to be rotated and $\dot{\mathbf{q}}$ an unit-quaternion. This can be proven if the quaternions are expanded as follows using (3.35).

$$\dot{\mathbf{r}}' = \dot{\mathbf{q}} \otimes \dot{\mathbf{r}} \otimes \dot{\mathbf{q}}^* = (\mathbf{Q}\dot{\mathbf{r}}) \otimes \dot{\mathbf{q}}^* = \overline{\mathbf{Q}}^T (\mathbf{Q}\dot{\mathbf{r}}) = \left(\overline{\mathbf{Q}}^T \mathbf{Q}\right)\dot{\mathbf{r}},$$

$$\overline{\mathbf{Q}}^T \mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \begin{matrix} q_0^2 + q_x^2 \\ -q_y^2 - q_z^2 \end{matrix} & 2(q_x q_y - q_0 q_z) & 2(q_x q_z + q_0 q_y) \\ 0 & 2(q_x q_y + q_0 q_z) & \begin{matrix} q_0^2 - q_x^2 \\ +q_y^2 - q_z^2 \end{matrix} & 2(q_y q_z - q_0 q_x) \\ 0 & 2(q_x q_z - q_0 q_y) & 2(q_y q_z + q_0 q_x) & \begin{matrix} q_0^2 - q_x^2 \\ -q_y^2 + q_z^2 \end{matrix} \end{bmatrix} \tag{3.38}$$

Since the three upper-right elements of $\overline{\mathbf{Q}}^T \mathbf{Q}$ are zero $\dot{\mathbf{r}}'$ is purely imaginary. Horn proves that the lower-right-hand 3×3 submatrix of $\overline{\mathbf{Q}}^T \mathbf{Q}$ represents a valid rotation matrix which is defined by the unit-quaternion $\dot{\mathbf{q}}$ [HOR].

If a rotation is defined by an axis **u** (a vector of length 1) and a rotation angle ϕ then the corresponding quaternion $\dot{\mathbf{q}}$ is calculated as follows [HOR]:

$$\dot{\mathbf{q}} = \left(\cos\frac{\phi}{2}, \sin\frac{\phi}{2}\mathbf{u} \right). \tag{3.39}$$

Finding the best rotation

As stated before, the remaining matching error consists only of the first term in (3.28) since the second is zero by definition and the computation of the translation as shown in (3.29) sets the third to zero for any **R**.

The remaining error can be rewritten using the quaternion calculus and the representation of (3.37) for the rotation **R**.

$$\dot{\mathbf{e}} = (e, \mathbf{0}) = \left(\sum_{k=1}^{N} \left\| \mathbf{R}\mathbf{p}'_k - \mathbf{y}'_k \right\|^2, \mathbf{0} \right) = \sum_{k=1}^{N} \left| \dot{\mathbf{q}} \otimes \dot{\mathbf{p}}'_k \otimes \dot{\mathbf{q}}^* - \dot{\mathbf{y}}'_k \right|^2 \tag{3.40}$$

$$\text{with } \dot{\mathbf{p}}'_k = (0, \mathbf{p}'_k) \text{ and } \dot{\mathbf{y}}'_k = (0, \mathbf{y}'_k)$$

The following developments give a way to find the unit-quaternion $\dot{\mathbf{q}}$ that minimizes *e* by calculating the eigenvector corresponding to the smallest eigenvalue of a matrix **A**.

First, the error in (3.40) is multiplied with the square magnitude of the unit-quaternion $\dot{\mathbf{q}}$ which does not change its value but will allow the regrouping of the different terms into one multiplication:

$$e = \sum_{k=1}^{N} \left| \dot{\mathbf{q}} \otimes \dot{\mathbf{p}}'_k \otimes \dot{\mathbf{q}}^* - \dot{\mathbf{y}}'_k \right|^2 |\dot{\mathbf{q}}|^2 , \tag{3.41}$$

which can be factorized and simplified into

$$\begin{aligned} e &= \sum_{k=1}^{N} \left| \dot{\mathbf{q}} \otimes \dot{\mathbf{p}}'_k \otimes \dot{\mathbf{q}}^* \otimes \dot{\mathbf{q}} - \dot{\mathbf{y}}'_k \otimes \dot{\mathbf{q}} \right|^2 \text{ with } |\dot{\mathbf{q}} \otimes \dot{\mathbf{n}}|^2 = |\dot{\mathbf{q}}|^2 |\dot{\mathbf{n}}|^2 \\ &= \sum_{k=1}^{N} \left| \dot{\mathbf{q}} \otimes \dot{\mathbf{p}}'_k - \dot{\mathbf{y}}'_k \otimes \dot{\mathbf{q}} \right|^2 \text{ with } \dot{\mathbf{q}}^* \otimes \dot{\mathbf{q}} = |\dot{\mathbf{q}}|^2 = (1, \mathbf{0}) \end{aligned} \tag{3.42}$$

Since the imaginary part of the error quaternion $\dot{\mathbf{e}}$ is a zero vector only the real part of $\dot{\mathbf{e}}$ is observed and rewritten using (3.36)

$$\begin{aligned} e &= \sum_{k=1}^{N} \left\| \dot{\mathbf{q}} \otimes \dot{\mathbf{p}}'_k - \dot{\mathbf{y}}'_k \otimes \dot{\mathbf{q}} \right\|^2 = \sum_{k=1}^{N} \left(\dot{\mathbf{q}} \otimes \dot{\mathbf{p}}'_k - \dot{\mathbf{y}}'_k \otimes \dot{\mathbf{q}} \right) \cdot \left(\dot{\mathbf{q}} \otimes \dot{\mathbf{p}}'_k - \dot{\mathbf{y}}'_k \otimes \dot{\mathbf{q}} \right) \\ &= \sum_{k=1}^{N} \left(\dot{\mathbf{q}} \otimes \dot{\mathbf{p}}'_k \right) \cdot \left(\dot{\mathbf{q}} \otimes \dot{\mathbf{p}}'_k \right) - 2 \left(\dot{\mathbf{q}} \otimes \dot{\mathbf{p}}'_k \right) \cdot \left(\dot{\mathbf{y}}'_k \otimes \dot{\mathbf{q}} \right) + \left(\dot{\mathbf{y}}'_k \otimes \dot{\mathbf{q}} \right) \cdot \left(\dot{\mathbf{y}}'_k \otimes \dot{\mathbf{q}} \right) \end{aligned} \tag{3.43}$$

and gives together with the matrix representations of (3.35)

$$\begin{aligned} e &= \sum_{k=1}^{N} \left(\overline{\mathbf{P}}'_k \dot{\mathbf{q}} \right) \cdot \left(\overline{\mathbf{P}}'_k \dot{\mathbf{q}} \right) - 2 \left(\overline{\mathbf{P}}'_k \dot{\mathbf{q}} \right) \cdot \left(\mathbf{Y}'_k \dot{\mathbf{q}} \right) + \left(\mathbf{Y}'_k \dot{\mathbf{q}} \right) \cdot \left(\mathbf{Y}'_k \dot{\mathbf{q}} \right) \\ &= \sum_{k=1}^{N} \left(\overline{\mathbf{P}}'_k \dot{\mathbf{q}} \right)^T \overline{\mathbf{P}}'_k \dot{\mathbf{q}} - 2 \left(\overline{\mathbf{P}}'_k \dot{\mathbf{q}} \right)^T \mathbf{Y}'_k \dot{\mathbf{q}} + \left(\mathbf{Y}'_k \dot{\mathbf{q}} \right)^T \mathbf{Y}'_k \dot{\mathbf{q}} \\ &= \sum_{k=1}^{N} \dot{\mathbf{q}}^T \overline{\mathbf{P}}'^T_k \overline{\mathbf{P}}'_k \dot{\mathbf{q}} - 2 \dot{\mathbf{q}}^T \overline{\mathbf{P}}'^T_k \mathbf{Y}'_k \dot{\mathbf{q}} + \dot{\mathbf{q}}^T \mathbf{Y}'^T_k \mathbf{Y}'_k \dot{\mathbf{q}} \\ &= \sum_{k=1}^{N} \dot{\mathbf{q}}^T \overline{\mathbf{P}}'^T_k \overline{\mathbf{P}}'_k \dot{\mathbf{q}} - 2 \dot{\mathbf{q}}^T \overline{\mathbf{P}}'^T_k \mathbf{Y}'_k \dot{\mathbf{q}} + \dot{\mathbf{q}}^T \mathbf{Y}'^T_k \mathbf{Y}'_k \dot{\mathbf{q}} \\ &= \sum_{k=1}^{N} \dot{\mathbf{q}}^T \left(\overline{\mathbf{P}}'^T_k \overline{\mathbf{P}}'_k - 2 \overline{\mathbf{P}}'^T_k \mathbf{Y}'_k + \mathbf{Y}'^T_k \mathbf{Y}'_k \right) \dot{\mathbf{q}} \end{aligned} \tag{3.44}$$

The minimization problem can now be stated as follows:

$$e_{\min} = \min_{\hat{\mathbf{q}}} \sum_{k=1}^{N} \hat{\mathbf{q}}^T \mathbf{A}_k \hat{\mathbf{q}} = \min_{\hat{\mathbf{q}}} \hat{\mathbf{q}}^T \mathbf{A} \hat{\mathbf{q}}$$
$$\text{with } \mathbf{A}_k = \overline{\mathbf{P}}_k'^T \overline{\mathbf{P}}_k' - 2\overline{\mathbf{P}}_k'^T \mathbf{Y}_k' + \mathbf{Y}_k'^T \mathbf{Y}_k' \text{ and } \mathbf{A} = \sum_{k=1}^{N} \mathbf{A}_k \tag{3.45}$$

where $\hat{\mathbf{q}}$ is an unit-quaternion. If the following sums of cross-correlations are introduced

$$S_{xx} = \sum_{k=1}^{N} \mathbf{p}'_{k,x} \cdot \mathbf{y}'_{k,x}, \quad S_{xy} = \sum_{k=1}^{N} \mathbf{p}'_{k,x} \cdot \mathbf{y}'_{k,y} \text{ and so on,} \tag{3.46}$$

then $\mathbf{A}$ can be rewritten in a more expressive manner as

$$\mathbf{A} = \sum_{k=1}^{N} \|\mathbf{p}'_k\|^2 \mathbf{I} - 2\mathbf{B} + \sum_{k=1}^{N} \|\mathbf{y}'_k\|^2 \mathbf{I}$$
$$\text{with } \mathbf{B} = \begin{bmatrix} S_{xx}+S_{yy}+S_{zz} & S_{yz}-S_{zy} & S_{zx}-S_{xz} & S_{xy}-S_{yx} \\ S_{yz}-S_{zy} & S_{xx}-S_{yy}-S_{zz} & S_{xy}+S_{yx} & S_{zx}+S_{xz} \\ S_{zx}-S_{xz} & S_{xy}+S_{yx} & -S_{xx}+S_{yy}-S_{zz} & S_{yz}+S_{zy} \\ S_{xy}-S_{yx} & S_{zx}+S_{xz} & S_{yz}+S_{zy} & -S_{xx}-S_{yy}+S_{zz} \end{bmatrix} \tag{3.47}$$

Note that $\mathbf{B}$, hence $\mathbf{A}$, are symmetric matrices. Applying the technique of Lagrange multipliers, the problem of (3.45) is equivalent to

$$\text{find } \min_{\hat{\mathbf{q}}} L = \left[\hat{\mathbf{q}}^T \mathbf{A} \hat{\mathbf{q}} + \lambda\left(1 - \|\hat{\mathbf{q}}\|^2\right)\right]$$
$$\text{subject to the constraint } \|\hat{\mathbf{q}}\| = 1 \tag{3.48}$$

This is solved by setting the partial derivatives of the criterion L to zero.

$$\frac{\partial L}{\partial \hat{\mathbf{q}}} = \mathbf{A}\hat{\mathbf{q}} + \left(\hat{\mathbf{q}}^T \mathbf{A}\right)^T - 2\lambda\hat{\mathbf{q}} = \mathbf{A}\hat{\mathbf{q}} + \mathbf{A}^T\hat{\mathbf{q}} - 2\lambda\hat{\mathbf{q}} = \mathbf{0} \tag{3.49}$$

Since $\mathbf{A}$ is a symmetric matrix (see (3.47)) the expression of (3.49) becomes

$$2\mathbf{A}\hat{\mathbf{q}} - 2\lambda\hat{\mathbf{q}} = \mathbf{0}$$
$$\mathbf{A}\hat{\mathbf{q}} = \lambda\hat{\mathbf{q}} \tag{3.50}$$

which expresses the fact that $\dot{\mathbf{q}}$ is an eigenvector of matrix **A**. Actually, the unit-quaternion which minimizes the matching error *e* is equal to the normalized eigenvector associated with the smallest eigenvalue because the matching error is equal to the eigenvalue (see (3.51)). All eigenvalues are positive since the error is a sum of square distances.

$$\begin{aligned} e &= \dot{\mathbf{q}}^T \mathbf{A} \dot{\mathbf{q}} \text{ and } \mathbf{A}\dot{\mathbf{q}}_{\min} = \lambda_{\min} \dot{\mathbf{q}}_{\min} \\ e_{\min} &= \dot{\mathbf{q}}_{\min}^T \mathbf{A} \dot{\mathbf{q}}_{\min} = \dot{\mathbf{q}}_{\min}^T \lambda_{\min} \dot{\mathbf{q}}_{\min} \\ &= \lambda_{\min} \dot{\mathbf{q}}_{\min}^T \dot{\mathbf{q}}_{\min} = \lambda_{\min} \left\| \dot{\mathbf{q}}_{\min} \right\|^2 = \lambda_{\min} \end{aligned} \tag{3.51}$$

The presented method to calculate the optimal rigid transformation, which minimizes the matching error, can be summarized as follows:

1) refer all data points to their centroid (3.26)
2) build the matrix **A** using the definitions of (3.46) and (3.47)
3) calculate the smallest normalized eigenvector and use it as quaternion $\dot{\mathbf{q}}$
4) use $\dot{\mathbf{q}}$ to calculate the best rotation matrix **R** with (3.38)
5) find the best translation **t** by calculating (3.29)

3.5.3 Weighted couplings

In several applications there is some interest in attributing different weights to the couplings in $C^{(i)}(\mathbf{p}_k, \mathbf{y}_k)$ used for the matching error minimization. This allows, for example, to weight the couplings according to their reliability or to consider the influence of outliers.

Every coupling in $C^{(i)}(\mathbf{p}_k, \mathbf{y}_k)$ gets a weight w_k and the matching error *e* of (3.24) becomes

$$e = \sum_{k=1}^{N} w_k \left\| \left(\mathbf{R}\mathbf{p}_k + \mathbf{t} \right) - \mathbf{y}_k \right\|^2 . \tag{3.52}$$

The principle of the error minimization showed in the previous section does not change. However, the substitution made in (3.26) is replaced by

$$\begin{aligned} \overline{\mathbf{p}} &= \frac{1}{W}\sum_{k=1}^{N} w_k \mathbf{p}_k, \quad \overline{\mathbf{y}} = \frac{1}{W}\sum_{k=1}^{N} w_k \mathbf{y}_k, \\ \mathbf{p}' &= \mathbf{p} - \overline{\mathbf{p}}, \qquad \mathbf{y}' = \mathbf{y} - \overline{\mathbf{y}}, \\ W &= \sum_{k=1}^{N} w_k \end{aligned} \tag{3.53}$$

and the calculation of the matrix **A** is done with

$$S_{xx} = \sum_{k=1}^{N} w_k \left(\mathbf{p}'_{k,x} \cdot \mathbf{y}'_{k,x}\right), \; S_{xy} = \sum_{k=1}^{N} w_k \left(\mathbf{p}'_{k,x} \cdot \mathbf{y}'_{k,y}\right) \text{ and so on,} \tag{3.54}$$

and

$$\mathbf{A} = \sum_{k=1}^{N} w_k \|\mathbf{p}_k\|^2 \mathbf{I} - 2\mathbf{B} + \sum_{k=1}^{N} w_k \|\mathbf{y}_k\|^2 \mathbf{I}. \tag{3.55}$$

A meaningful use of the coupling weights w_k is to make them vary with the coupling distance d_k of (3.6) or to use a reliability measure calculated for every point during the scanning process. For example points with a surface normal parallel to the scanner view direction can be sensed more accurately. Different functions, which define w_k using d_k, are presented in the Section 4.5.5 and the Section 5.6.1.

3.5.4 Convergence of the modified ICP algorithm

The use of several features in the coupling distance *d* (3.15) or the introduction of weights in the matching error *e* (3.52) can modify the convergence of the ICP algorithm.

So far, the monotonical convergence of the ICP algorithm is proven if both the coupling distance *d* and the matching error *e* use the geometric feature and the couplings are not weighted (see Section 3.3.3).

In the following, a modified ICP algorithm is discussed. First, methods are discussed to include the color and the surface orientation features in the matching error minimization. Then the following sections investigate the influence on the ICP convergence when these features are not observed for the matching error minimization. Finally, the convergence behavior of the ICP algorithm with coupled weights in the error minimization is discussed.

Color

The matching error of (3.24) including both the geometric feature and the color feature results in

$$e = \sum_{k=1}^{N} \left\| \left(\mathbf{R}\mathbf{g}_{p,k} + \mathbf{t} \right) - \mathbf{g}_{x,k} \right\|^2 + \sum_{k=1}^{N} \left\| \mathbf{c}_{p,k} - \mathbf{c}_{x,k} \right\|^2 . \qquad (3.56)$$

Since the color information of a surface point is not affected by the calculated rigid transformation (**R**, **t**) there is no need to include it in the matching error minimization [GODb]. This allows the use of the closed-form solution based on quaternions without modifications.

The proof of convergence of Section 3.3.3 still holds since $e^{(i)} \leq d^{(i)}$ and $d^{(i+1)} \leq e^{(i)}$ are true because the color term in (3.56) does not change with (**R**, **t**).

Surface orientation

If the surface orientation feature expressed by the surface normal vector is included in the minimization process the matching error changes as follows

$$e = \sum_{k=1}^{N} \left\| \left(\mathbf{R}\mathbf{g}_{p,k} + \mathbf{t} \right) - \mathbf{g}_{x,k} \right\|^2 + \sum_{k=1}^{N} \left\| \mathbf{R}\mathbf{n}_{p,k} - \mathbf{n}_{x,k} \right\|^2 . \qquad (3.57)$$

The normal feature is affected by the rotation of the rigid transformation which minimizes *e*. No substitution as in (3.26) exists to bring (3.57) in a form which could be solved with the closed-form solution presented in Section 3.5.2. Iterative least-square methods as for example extended Kalman filtering have to be applied to minimize (3.57) (see [CHA] and [FELa]). Such methods usually need more computation time [ARU].

Another approach proposed in this work minimizes only the first term of (3.57) using the closed-form solution based on quaternions. Then, the convergence of the ICP algorithm is not guaranteed any more. Because the features in the coupling distance *d* and the matching error *e* are no longer the same and the normal vectors are changed by **R**, the convergence condition $d^{(i+1)} \leq e^{(i)}$ of Section 3.3.3 is not necessarily true.

However, experimental work presented in the Section 5.8 investigate the practical use and convergence of this approach.

Coupling weights

As stated in Section 3.4.1 and Section 3.5.3 there is an interest to attribute different weights to the closest point couplings used in the matching error minimization.

The convergence condition of (3.11) still holds when the value of the coupling weights w_k does not change from one iteration to another. This is for example the case if the weights w_k represent some fixed reliability measure that is associated to the data points $\mathbf{p}_k$.

In the object digitization application, the weights are used to exclude points in P which do not overlap with X (see Section 5.6.1). Couplings, which have a coupling distance d larger than a certain threshold, are not used. As the data points are moved during the ICP iterations, the points which are excluded change and therefore the convergence proof of Section 3.3.3 can not be applied any more [GODb] [ZHA].

However, experimental results based on two different weighing functions presented in Section 4.5.5 and Section 5.6.1 show that the ICP algorithm converges successfully even if the weights change during the iterations.

Conclusions

The modified ICP algorithm using only the geometric distance for the matching error minimization converges to a minimum when the following conditions are met:

1) The additional features used for the closest point search are not affected by the rigid transformation

2) The coupling weights w_k associated to the $\mathbf{p}_k$ are constant.

3.6 Iteration termination

Geometric point matching performed with the ICP algorithm is intensive in computation. This provides the motivation to keep the number of needed iterations as low as possible. Different measures are proposed to detect the moment when the ICP algorithm has reached a minimum:

Absolute error: the coupling error $d^{(i)}$ falls below a certain threshold

Error change: the coupling error change $d^{(i-1)} - d^{(i)}$ of two successive iterations falls below a certain threshold [BESb]

Pose change: the calculated rigid transformation $(\mathbf{R}^{(i)}, \mathbf{t}^{(i)})$ at iteration i is small [CHA] which is the case when the length of the translation vector $\mathbf{t}^{(i)}$ and the rotation angle corresponding to $\mathbf{R}^{(i)}$ are both small

The different thresholds are determined empirically. The sampling density of the surface points can be used to define an absolute error threshold since it allows the mean coupling distance of two matched surfaces to be estimated.

Each of the above termination criteria has its own advantages. If an application as object digitizing needs accurate matching, the ICP algorithm should not be stopped before the absolute coupling error falls below the fixed accuracy.

In other applications such as object recognition a short system response is more important. The error or pose change allows quick detection if the matching will continue to change or if it is locked in a local minimum.

For some applications the coupling error $d^{(i)}$ representing the mean square distance of all ICP couplings is showed to be insufficient to measure the quality of the surface registration. Different error statistics are proposed to replace $d^{(i)}$ for the absolute error or error change measure (see Section 4.5.4).

3.7 Successful convergence

Section 3.3.3 assessed the existence of convergence for the standard ICP algorithm. However, the reached minimum does not necessarily coincide with the absolute minimum. Successful convergence is obtained when the matching finds the absolute minimum.

The absolute minimum of the matching error is only obtained for a subset of all the possible pose configurations of two surfaces. The closest point search which approximates the correct but unknown correspondence function c is valid only if both surfaces are roughly registered. The ICP algorithm needs an estimate of the correct matching solution or else it misses the absolute minimum.

The following section presents a method that investigates the nature of the pose configurations which lead to successful convergence and means to display it.

3.7.1 SIC-range

The rigid transformation, which brings the surface *P* into a certain pose expressed relatively to the fixed surface *X*, is called an initial configuration. Considering the 6D space of all possible initial configurations, successful matching is obtained only from a subrange of it, named the successful initial configuration range or SIC-range. Note, that this range does not have to be connected.

Preferred setup

The following setup is defined in order to measure the SIC-range empirically. The dimension of the 6D (rotation and translation parameters) space including all possible initial configurations is reduced to a 3D space as proposed in Figure 3.14. The surfaces *P* and *X* are orientated with the help of the a priori knowledge of the scanner view direction.

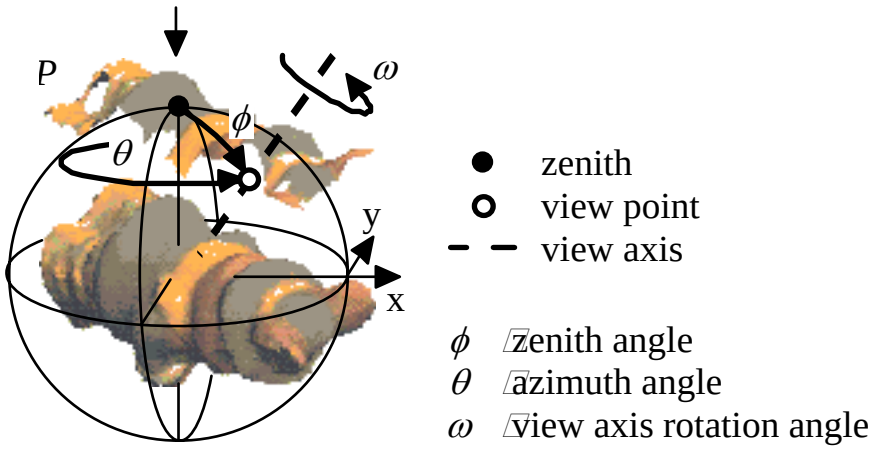


Figure 3.14 SIC-range measurement setup

The surface *P* is first correctly registered with the surface *X*. This can be done for example with an ICP matching supervised by an human

operator. Then, the center of mass of the surface P is moved along the scan view direction until it crosses the circumsphere of the surface X with the center of the circumsphere being the center of mass of X. This intersection point is referred to as the zenith.

The reduced space of initial configurations is defined by the triple (ϕ, θ, ω) where ϕ and θ are respectively zenith and azimuth angles in the spherical reference system of X and ω designates the rotation angle around the view axis. A point on the circumsphere defined by the pair (ϕ, θ) is called a view point. A view axis is defined by the center of mass of X and a view point.

The surface P is placed at several view points and rotated around the corresponding view axis. This setup is well suited for studying successful convergence and motivated by the following facts:

- The surface P is placed at an optimal distance near the center of mass of X so that during the first ICP iterations meaningful couplings can be established.

- During the ICP matching, the surface P is moved towards the surface X in a way that discourages any match of P with the invisible surface parts of X. Only surfaces with similar orientation get coupled.

SIC-range of (ϕ, θ, ω) space

Thus, the SIC-range is measured in the (ϕ, θ, ω) space. In order to do so, the (ϕ, θ, ω) parameter space of all possible initial configurations has to be inspected for successful matching. The set of all successful configurations forms the SIC-range.

Starting from the zenith configuration $(\phi, \theta, \omega) = (0°, 0°, 0°)$, the complete (ϕ, θ, ω) parameter space is explored in steps varying the azimuth angle θ from 0° to 360°, the zenith angle ϕ from 0° to 180° and the view axis rotation angle ω from 0° to 360°. More formally, the transformation, which orientates P on a view point, rotates P by zenith angle ϕ around the axis which is perpendicular to the great circle of azimuth angle θ.

For every initial configuration defined by (ϕ, θ, ω), the ICP algorithm is launched. If the ICP converges towards the absolute minimum the initial configuration is labeled as successful and its pose is added to the SIC-range.

The absolute minimum is determined practically by one of the following two methods:

– The convergence is successful if the matching error falls below a fixed threshold. The threshold is selected small enough.

– If the pose of the successfully registered surface P is known in advance, the following method applies. Calculate the difference of the pose found by the ICP algorithm and the correct one. If the resulting translation vector and rotation angle are both small, the absolute minimum is reached.

3.7.2 SIC-map

The following representation permits the visualization of the 3D SIC-range of the (ϕ, θ, ω) space in a 2D graph.

The view points on the circumsphere of X are projected on a plane tangential to the zenith. This plane represents the (ϕ, θ) space where ϕ and θ correspond to the polar coordinates which define the view point positions. Therefore, view points having the same zenith angle lie on circles.

For every view point, a small circle represents the range of the parameter ω. A successful initial configuration of the SIC range is drawn as a black sector in the circle of the corresponding view point (see Figure 3.15). The sum of all black sectors for a specific view point is also referred as the SIC-range of ω..

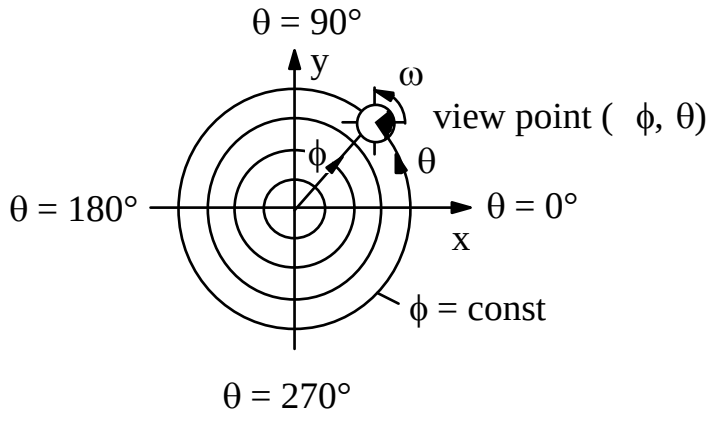


Figure 3.15 Definition of the SIC-map

Chapter 4
Object recognition

The object recognition task is the first of three applications implemented in this work. All three applications use the geometric point matching algorithm presented in Chapter 3. Since the nature of the matching problem changes from one application to another (see Section 3.4.1) the geometric point matching is adapted as required.

The considered objects have free-form surfaces which are represented by points or triangles as motivated in Chapter 2.

The presentation of the different applications in this and the following two chapters is structured as follows. After an introduction to the problem to be solved has been provided, a comparison to other work is given, then the implementation and adaptation of the ICP algorithm follows and finally results are presented.

The presented work has been published in [HUGa] [HUGb] [HUGc] [HUGd] [SCHc] [SCHd] [SCHg] [SCHh] [SCHi].

4.1 Introduction

The term object recognition covers a large range of computer vision applications. The work described in this chapter is carried out in the context of knowledge based 3D object recognition which is outlined as follows.

The object recognition system has to find the type and the pose of 3D objects placed randomly on a workspace. The objects have free-form

surfaces as defined in Section 2.1. The vision system has full knowledge about the shape and color of the objects and the working environment.

Before the object recognition system is described in more detail, the terms "test" and "model" need to be defined. The test refers to the real world object information acquired by the 3D scanner. The model encompasses the object knowledge stored in the database. The object recognition system tries to find the corresponding model for a sensed test (see Section 1.3).

4.1.1 Hybrid 3D vision system

A hybrid object recognition system has been developed together with the vision group of the Institute of Microengineering of the Swiss Federal Institute of Technology in Lausanne. It combines range and intensity images to generate and verify object hypotheses. Figure 4.1 gives an overview of the different modules which build this system.

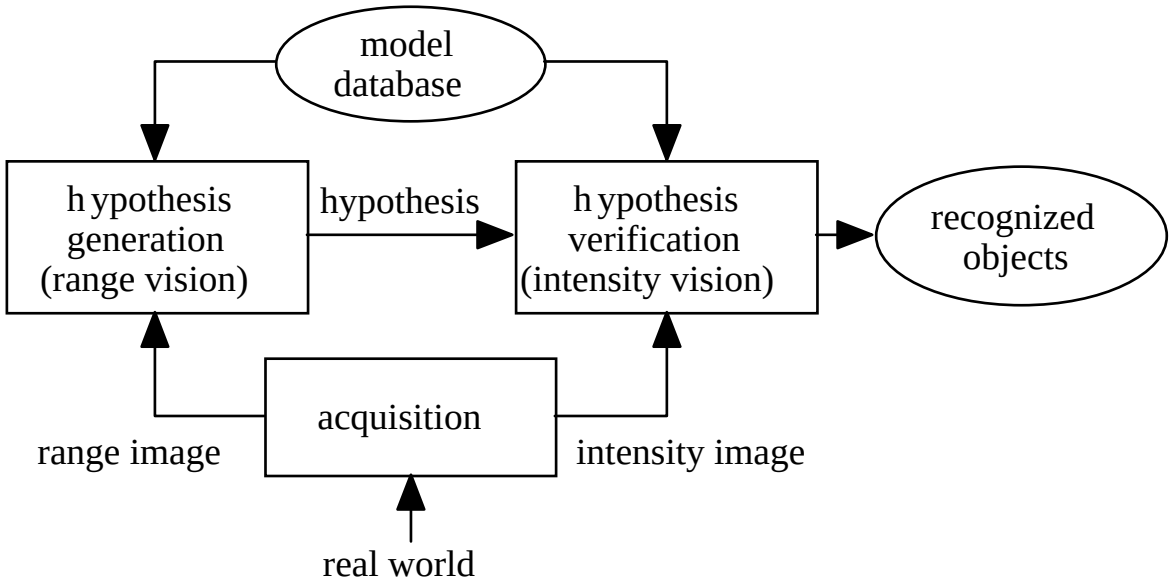


Figure 4.1 Hybrid 3D vision system

The scene acquisition is performed by a 3D scanner. It acquires the range and intensity images. The 3D information from the range image is used in the hypothesis generation module whereas the hypothesis verification uses the intensity information.

The hypothesis generation module generates a hypothesis consisting of the type and the pose of the objects present in the scene. This object

recognition task uses the 3D geometry information from the object models in the database.

The hypothesis verification module uses the generated hypotheses to render a synthetic image of the scene. This module operates on the same model representation as the hypothesis generation module. Additionally, the stored photometric information of the objects and a priori knowledge about the workspace and camera position are used for the image rendering. The synthetic image is compared to the real intensity image in order to verify the validity of the object hypotheses.

Finally, the hybrid 3D vision system returns all recognized objects to build a complete representation of the sensed real world.

This work presents the development of the hypothesis generation module whereas the hypothesis verification module is described in [NATb].

4.2 Comparison with other work

In previous work at our lab, the object recognition task has been performed according to the classical segmentation approach [AMA] [GIN]. The range image of the scene is first segmented into planar faces. These are then grouped and compared to polyhedral models. This approach had to be abandoned because the segmentation of objects becomes unstable when dealing with free-form objects.

Therefore, geometric point matching has been selected. The ICP algorithm matches the object surface of the test with all models in order to find the correct one.

Unfortunately, the ICP algorithm does not always find the correct matching for a random pose configuration of two surfaces representing the same shape. The ICP algorithm needs a rough pose estimate in order to converge to the absolute minimum (see Section 3.7). This problem has to be solved before the ICP algorithm can be applied to object recognition.

To our knowledge, only Chua and Jarvis [CHU] proposed the ICP algorithm for object recognition. They find a pose estimate for the test and the model surface by first establishing 3-tuples with corresponding principal curvatures on both surfaces. Other authors propose the use of features as edges or triangles to find a pose estimate for surface registration tasks [BERa] [CHS] [FELb]. This usually results in many possible candidates which have to be verified [CHS].

One or several pose estimates can be derived from contextual knowledge about the 3D scanner setup [BLA] [CHE] or from scan sequences as for example in object tracking [SIM] [ZHA].

In this work a priori knowledge of the model pose is not available and feature extraction, especially when using second-order derivatives as for example curvatures is not reliable for free-form surfaces (see Section 2.2.2).

Here, a novel approach is proposed which is to apply the ICP algorithm for an object recognition task. Instead of looking for a rough pose estimate of the test and model surface, the presented object recognition system systematically launches the ICP algorithm for several well-conditioned test and model poses referred to as initial configurations. The number and setup of these initial configurations is based on the experimental inspection of the size of the range of successful convergence of the ICP algorithm as introduced in Section 3.7. This knowledge provides the assurance that the ICP algorithm finds a correct match to the test and the corresponding model for at least one initial configuration.

The following sections discuss the different methods which have been used to implement the ICP algorithm in the hypothesis generation module for the object recognition task.

4.3 Model database

The model database supports both hypothesis modules as shown in Figure 4.1. It encompasses the geometric and the photometric information of all objects which can appear in the real world. Both vision modules use this common database to ensure a consistent hypothesis representation. This simplifies the hypothesis generation since both systems use the same coordinate reference system.

The surfaces of the object models are represented by low-level primitives as cloud of points or colored triangle meshes. This allows the use of the same model representation for the vision system and for image rendering (see Section 2.3).

Model construction

In order to generate an object model one needs to take into account all the properties of the real object. Models are twofold. The first part reflects

the geometry of the object. The second part defines the attributes of the object, among which there are the photometric attributes, which describe the way the object interacts with light.

Three different techniques are used to construct the model database. The first uses the conventional technique of defining objects by entering the surface coordinates manually. This is generally considered as tedious.

The second and so far the most interesting technique uses 3D geometric databases from different commercial CAD packages. The advantage of this solution, especially in the case of automated assembly, is that precise and complete databases of technical drawings can be reused. This is possible since most CAD packages now allow the export of their data as cloud of points or triangle meshes.

A third technique relies on the use of 3D scanners which are used when the outer surface dimensions of free-form objects have to be determined. Simple cloud of points models have been generated by fusing different views of an object exposed on a turn table to the 3D scanner [HOU]. A more sophisticated technique for object digitizing is implemented in Chapter 5.

The digitizing system based on the 3D scanner acquires object geometry and color at the same time. This is one of its advantages compared to the other two model construction methods where color is usually not present.

4.4 Preprocessing of sensed data

The acquisition module feeds both the hypothesis generation and the hypothesis verification with the necessary data. Two different 3D scanners, which are presented in Section 2.4.1 and Section 4.7.2, acquire the range and intensity image.

The following sections describe the preprocessing of the range image used for the hypothesis generation.

4.4.1 Scene segmentation

The purpose of scene segmentation is to separate different objects in a range image. This is necessary since the applied object recognition can only match one object at a time.

If the depth of the background is known a simple threshold of the range image allows the location of the objects. This also works with textured backgrounds since the color information is not present in the range image.

If the different objects touch one another, segmentation methods based on region homogeneity can be used to separate them. However, this segmentation criterion is in general too weak to segment a scene of free-form objects.

This work proposes a change detection method to segment complex scenes. Rather than analyzing each image separately, it proceeds iteratively by detecting objects as they are moved into the observed scene. Change detection refers to the possibility of subtracting two successive images in order to find changes in the scene, like added or removed objects. Change detection of intensity images is widely used in quality control and motion detection and often applies a simple difference threshold of two following images [FRE] [HER] [RUSa].

Range change detection

Despite a wide use in intensity imaging, change detection has not found its way to range imaging. To our knowledge, the problem of change computation in the presence of real range images has not been addressed in the literature. In fact, there is an interest in detecting changes in range imaging. The motivation is related to the geometric nature of range imaging which results in measurements which reflect the intrinsic property of an object. Range differences are believed to be more stable than differences in the related intensity images. Obviously change detection can only be applied to images where changes happen. To isolate an object from a complex background at least two scene acquisitions are needed; before and after the object is moved.

The difference calculation of two range images is more complex than for intensity images since range images may contain invalid data represented by NIL values (see Section 2.4.2). The black regions in Figure 4.2 indicate such NIL values and indicate points where the 3D scanner could not measure the surface. Special rules have to be defined to interpret the difference of a NIL and a valid pixel or two NIL pixels. Methods which solve this problem are discussed in more details in [SCHc]. The following Figure 4.2 shows an example of the detection of both an added and a removed object in a complex scene using a sequence of two range images.

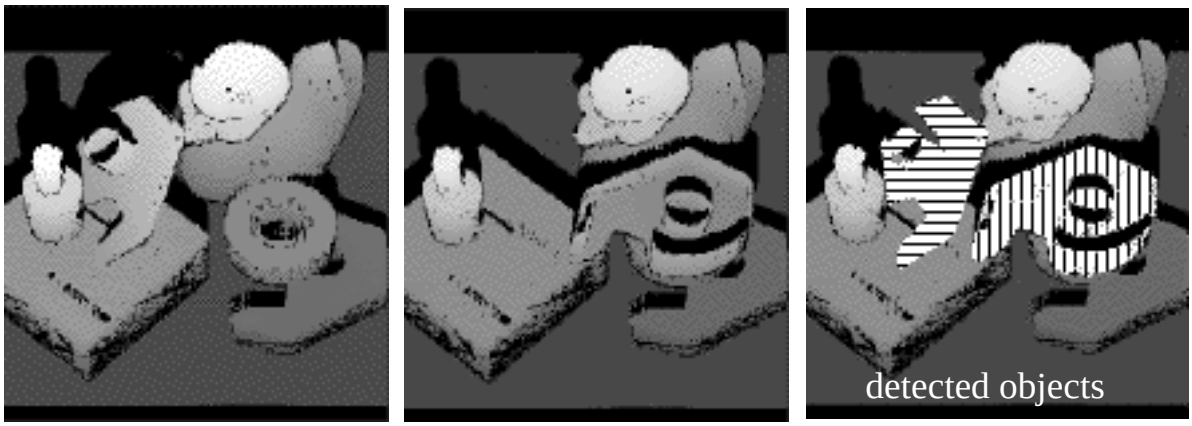


Figure 4.2 Segmentation of complex scenes with range change detection

4.4.2 Data reduction

A reduction of the number of measured points is usually performed in order to reduce the ICP computation complexity as discussed in Section 3.4.3. To obtain a successful matching it is necessary for the point reduction to result in a homogenous point distribution on the object surface [SCHd]. This is achieved by the two following methods which apply to different data arrangements.

When the data measurements are ordered in a range image which is a projection of the 3D scene in a 2D plane, a simple linear subsampling which takes every i-th point gives good results.

If the cloud of points is in random order, as for example for the models obtained from several fused point clouds, an iterative point grouping method can be applied [SCHd]. A point is chosen at random to center a sphere of a certain radius. All surface points which fall inside this sphere are identified. These points are then removed from the point set and replaced by the point which corresponds to their center of mass. The iterative grouping continues with another point taken from the remaining points. The size of the sphere controls the density of the reduced point cloud.

4.5 Hypothesis generation

The hypothesis generation module uses the range image and the models from the database as input. The range image segmentation and point reduction methods provide the 3D geometry for every object to be

recognized. A test consists of a cloud of points which represents the visible part of an object of the scene. Every test is matched with all models in order to find the type and the pose of the model corresponding to the test.

The following sections present the integration of the ICP algorithm in the object recognition system.

4.5.1 Recognition setup for the ICP algorithm

In a 3D object recognition system, the matching algorithm is used to compare a test with the models in the database. This should allow the test to be of any view of the corresponding model placed in any pose. As stated in Section 4.2, the ICP algorithm needs a rough pose estimate in order to successfully match the two surfaces. This work proposes to define several test and model poses called initial configurations. If enough initial configurations are used, the ICP algorithm will find the correct matching for at least one of them assuming that the test and the model represent the same object.

An exhaustive inspection of all possible configurations would take to much time since there are six degrees of freedom which may be changed, three for the rotation and three for the translation. Note that the model pose is fixed and only the test is moved. The a priori knowledge of the 3D scanner position helps to constraint the recognition search. This allows the setup for the test and the model to be defined, which is similar to the one used for the SIC-range establishment in Section 3.7.

First, given a view axis defined by the 3D scanner position and the center of mass of the test, the model is centered on this view axis so that the test lies in-between the model and the 3D scanner (see Figure 4.3). This placement ensures that the test surface not visible from the 3D scanner always faces the model. This also excludes the test surfaces from being compared with invisible model surfaces.

Second, V view points are distributed uniformly on the sphere, circumscribing the model as drawn in Figure 4.3. Every view point is defined by the two spherical coordinates ϕ and θ. Now, the model is rotated around its center of mass so that every view point lies once on the view axis. Furthermore, the model is rotated in S steps around the view axis for each of these configurations. Every step corresponds to a view axis rotation angle ω.

The ICP algorithm is then launched for every initial configuration as defined by the three coordinates ϕ, θ and ω. The use of different initial

configurations and an appropriate selection of the numbers *V* and *S* will ensure that the matching converges at least once towards a successful matching.

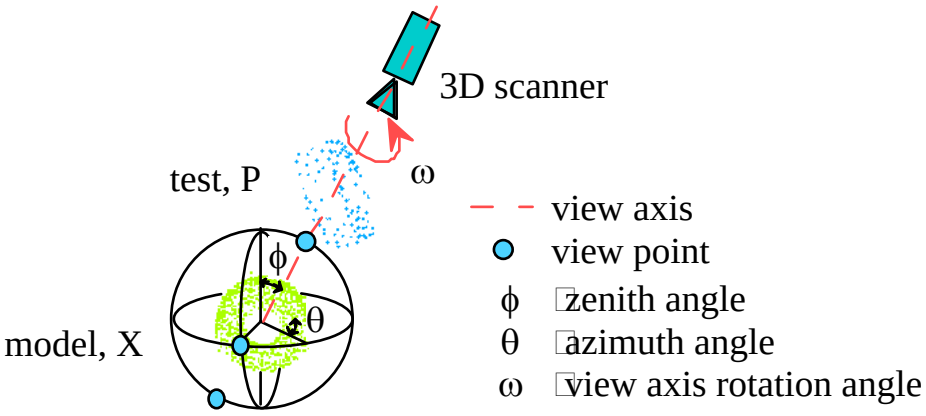


Figure 4.3 Recognition setup for the geometric point matching

If the database consists of several models, the procedure described above has to be applied to every model. Finally, the model with the smallest matching error is selected.

4.5.2 Recognition of scotch dispenser parts

Figure 4.4 shows some of the first successful recognition results obtained for three scotch dispenser parts. The recognized models are represented by a cloud of points and projected into the image space using the camera model obtained from the 3D scanner calibration. This allows to visually verify the recognition result.

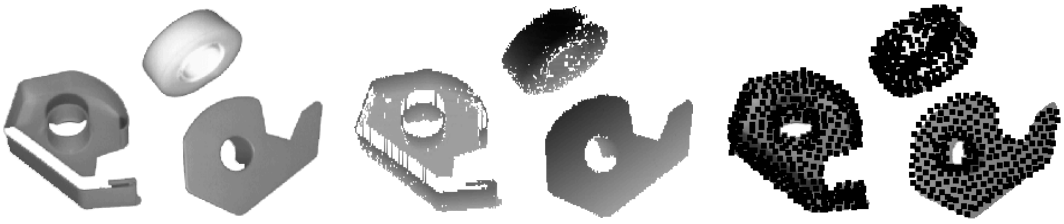


Figure 4.4 Recognition results for scotch dispenser parts

A test contains about 200 points whereas a model has about 500 points. The ICP algorithm performs 30 iterations for every initial configuration defined according the setup of the previous section. Here, the number of initial configurations is set to 24 with $V = 6$ and $S = 4$. These numbers have been obtained by testing several configurations and selecting the optimal one.

The next section presents experiments to asses the number of initial configurations in a more systematic way.

4.5.3 Optimal recognition setup

There is an interest in knowing the recognition performance as a function of the number of initial configurations. The less initial configurations are used the faster the object recognition is executed. However, the ICP algorithm may not find the correct matching if too few initial configurations are defined. This section proposes a method to build an optimal recognition setup with the knowledge of the range of successful convergence (SIC-range) of the ICP algorithm for a specific object.

The SIC-range defined in Section 3.7.1 uses the same definition of the pose of two surfaces as the object recognition setup defined in Section 4.5.1. Since both setups work in the (ϕ, θ, ω) parameter space, the results of the SIC-maps can be used to design an optimal recognition setup.

This process is shown here for the recognition of three toy objects presented in Figure 4.5. They represent injected plastic toys and are called duck, fish and swan.

Figure 4.5 Free-form objects: duck, fish and swan

First, a SIC-map is established for every object. Then, a simplified rectangular SIC-range is derived from the SIC-map and used to find a recognition setup which minimizes computation costs on the condition that successful convergence is obtained.

SIC-range

For each toy one typical view is selected since the SIC-map construction for every possible view is much too laborious. The SIC-range is elaborated as defined in Section 3.7.1. Figure 4.6 shows for all toy objects the zenith configuration on which the SIC-map is centered.

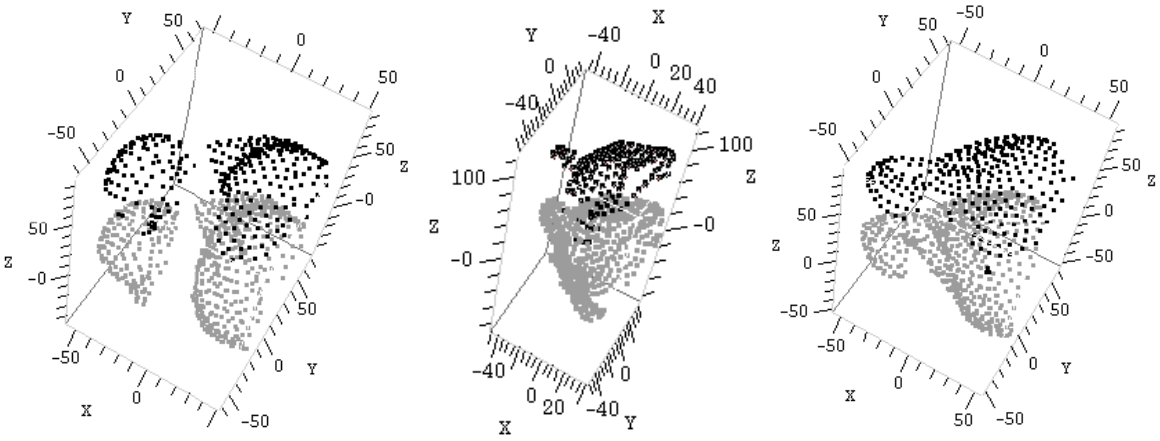


Figure 4.6 Zenith configuration of the three toy objects showing the model in gray and the test in black

Starting from the zenith configuration, the complete (ϕ, θ, ω) parameter space is traversed in steps of 10° varying azimuth angle θ from 0° to 360°, zenith angle ϕ from 0° to 180° and view axis rotation angle ω from 0° to 360°. The resulting SIC-maps of the three toy objects are presented in Figure 4.7 for the zenith angles ranging from 0° to 60°.

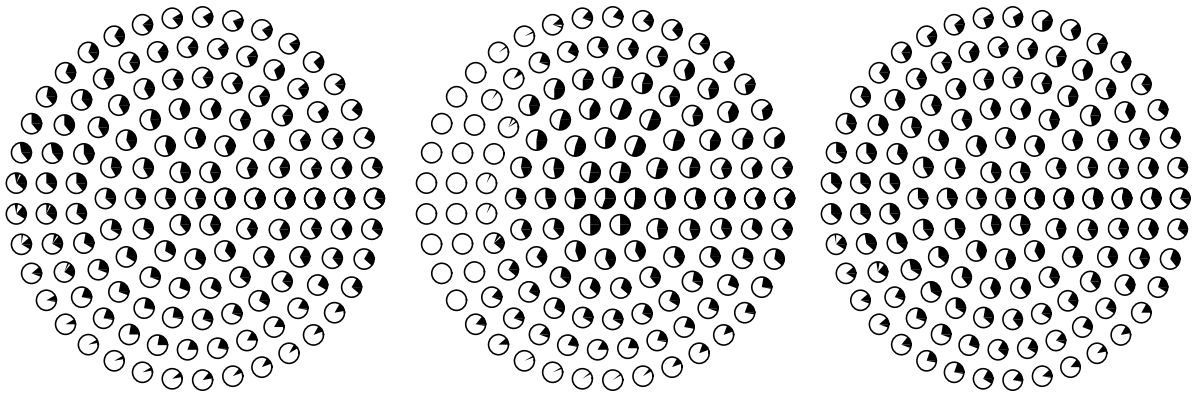


Figure 4.7 SIC-map for free-form toy objects

One can observe that the SIC-range of ω (black sectors at every view point) decreases for view points with growing zenith angle ϕ. While this decrease depends on the azimuth for the fish it is nearly azimuth independent for the duck and swan.

In any case, it is interesting to analyze how the size of the SIC-range of ω referred as $\Delta\omega$ varies with increasing zenith angle ϕ. Therefore, the following two statistical indicators are considered:

- $\Delta\overline{\omega}$, the mean of $\Delta\omega$ taken over all view points having a maximal zenith angle ϕ and any azimuth angle θ.
- $\Delta\omega_{\min}$, the minimum of $\Delta\omega$ taken over all view points having a maximal zenith angle ϕ and any azimuth angle θ.

The decreasing functions of $\Delta\overline{\omega}(\phi)$ and $\Delta\omega_{\min}(\phi)$ are reported in Figure 4.8.

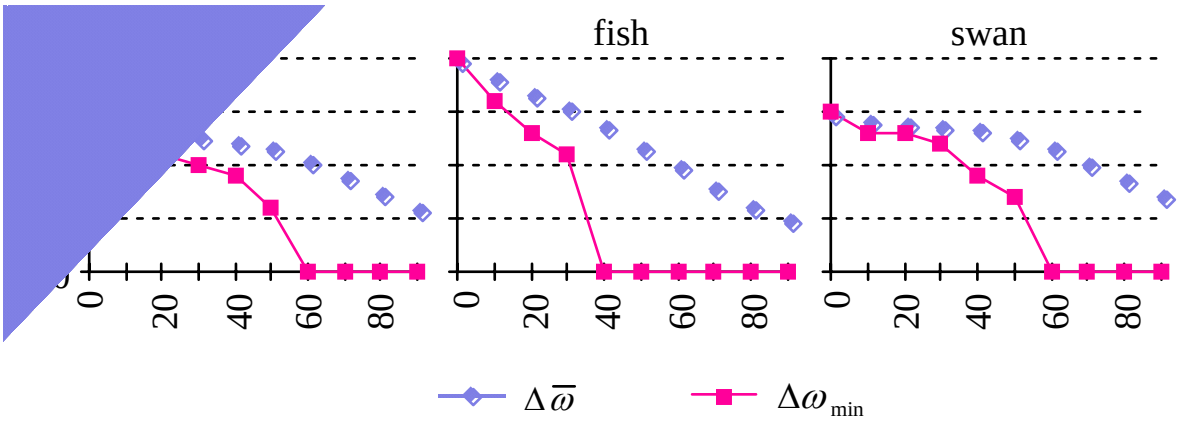


Figure 4.8 SIC-range statistics for ω

A $\Delta\omega_{\min}$ of 150° falls to 0° at a zenith angle ϕ_0 of 60° for the duck and the swan. For the fish, a $\Delta\omega_{\min}$ of 200° falls to 0° at a zenith angle ϕ_0 of 40°.

Rectangular SIC-range

Considering the indicator $\Delta\omega_{\min}$ a simple rectangular SIC-range description can be defined as follows:

$$\begin{gathered}\mathrm{SIC}(\phi_0, \Delta\omega_0) \\ \text{means that for } 0 \le \phi \le \phi_0 \text{ one has } \Delta\omega_{\min}(\phi) \ge \Delta\omega_0\end{gathered} \qquad (4.1)$$

In order to cover at least one successful configuration for any zenith angle, the value of ϕ_0 has to be chosen small enough so that the resulting $\Delta\omega_0$, which is defined as $\Delta\omega_0 = \Delta\omega_{\min}(\phi_0)$, is larger than zero. An approximate and conservative SIC-range modeling for the statistics in Figure 4.8 could be for example:

SIC-range of duck and swan: SIC(40°, 90°)

SIC-range of fish: SIC(35°, 60°)

A rectangular SIC-range estimates the range of successful convergence and can be used for the design of a reliable recognition system as described next.

Required initial configurations

On the one hand, the computational cost of recognition is proportional to $V \cdot S$, calling for small values of V (number of view points) and S (number of view axis rotations). On the other hand, V and S must be sufficiently large to ensure reliable recognition. This section proposes a method with which to define a set of required initial configurations (RIC) such that reliable recognition is obtained at minimal cost.

Considering an object with an hypothetical uniform and rectangular SIC-range given by SIC(ϕ_0, $\Delta\omega_0$), reliable recognition is obtained if the covering of the (ϕ, θ, ω) space with the set of RIC is dense enough to fulfill the conditions of a maximum spacing in rotation angle

$$\frac{360°}{S} < \Delta\omega_0 \tag{4.2}$$

and a maximum spacing in zenith angle

$$\phi^*(V) < \phi_0 \tag{4.3}$$

The function $\phi^*(V)$ is a basic function that relates the number of evenly distributed points on a sphere with their maximum spacing. More formally, it expresses the maximum angle distance of any point on the sphere to the set of V points distributed on it. Some values of $\phi^*(V)$, computed from a distribution obtained by a point repulsion algorithm [RUSb], are reported in Figure 4.9.

V	$\phi^*(V)$
4	70°
6	54°
8	46°
10	42°
12	37°
14	34°

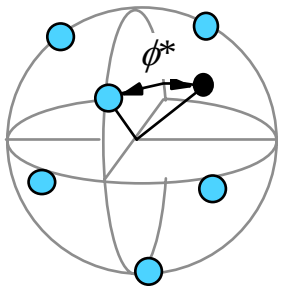


Figure 4.9 Maximum angular distance $\phi^*(V)$ to V view points evenly distributed on a sphere

Applying now the above conditions (4.2) and (4.3) now to the SIC-ranges of the toy objects, example choices of V and S are obtained as follows:

duck and swan: SIC(40°, 90°) $\rightarrow V = 12, S = 4$

fish: SIC(35°, 60°) $\rightarrow V = 14, S = 6$

In a further step, the computational cost $V \cdot S$ is minimized by varying the parameters $\Delta\omega_0$ and ϕ_0.

With these two conditions, the cost expresses as function of $\Delta\omega_0$ and ϕ_0 as

$$C(\phi_0, \Delta\omega_0) = const \cdot \frac{360°}{\Delta\omega_0} \cdot \phi^{*-1}(\phi_0) \approx S \cdot V, \qquad (4.4)$$

where $\Delta\omega_0$ and ϕ_0 are bound by the object specific functions $\Delta\omega_{\min}(\phi)$ of Figure 4.8. Under this constraint, optimizing the costs leads to an optimal solution $C_{opt}(\phi_0, \Delta\omega_0)$. Figure 4.10 draws the cost function $C(\phi_0, \Delta\omega_0)$ together with the $\Delta\omega_{\min}(\phi)$ function of the duck object.

Reliable recognition is ensured for every pair $(\phi_0, \Delta\omega_0)$ with a smaller ϕ_0 or $\Delta\omega_0$ than the frontier delimited by $\Delta\omega_{\min}(\phi)$. The minimal cost is obtained for $C_{opt}(\phi_0, \Delta\omega_0)$ = (50°, 60°) corresponding to the RIC defined by $V = 7$, $S = 6$.

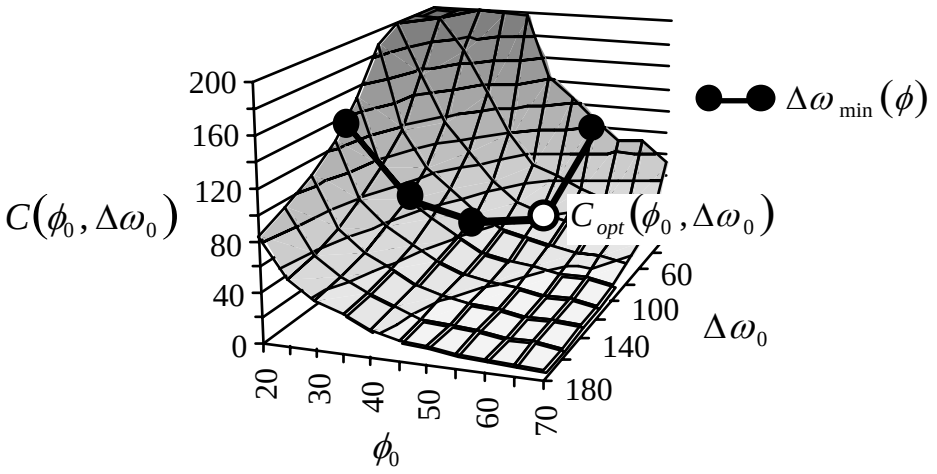


Figure 4.10 RIC assessment for duck object

Such RIC are defined individually for every model. When the ICP algorithm matches a test with a model it uses the corresponding RIC setup defined for the model.

Note that, the obtained numbers *V* and *S* are only estimates since the SIC-range elaboration has only been performed with one single view per object. Section 4.6 presents recognition experiments on a scene database in order to verify the performances of the designed setup.

Optimal model-to-test distance

The optimal size of the radius *r* of the sphere on which the view points are placed is inspected now. In the implemented system, *r* is equal to the circumradius r_c of the model.

The results in Figure 4.11 show that if the test is placed too far away from the model, the SIC-range of ω quickly decreases to zero. Because only few couplings are established in the first iteration, the error minimization results in bad rigid transformations.

In order to keep the recognition cost low the number of view points should be low. Therefore, the zenith angle ϕ_0 for which the SIC-range of ω reaches the minimal size $\Delta\omega_0$ should be large. Setups with $r \leq r_c$ are therefore better suited since the SIC-range of ω drops to zero for a higher ϕ.

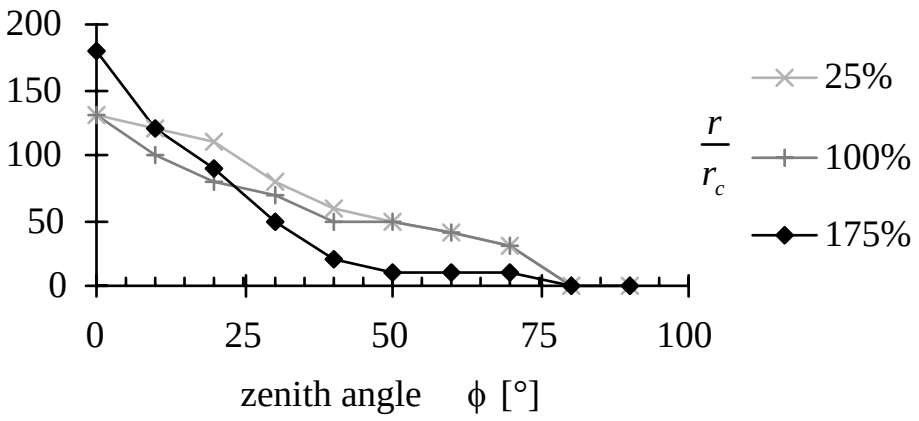


Figure 4.11 SIC-range of ω for different model-to-test distances

4.5.4 Search space pruning

If precise matching is requested, the ICP algorithm has to perform several iterations (about 30) for every initial configuration. Actually, precise matching is only needed for the correct model. Therefore, the search space containing all ICP matching iterations can be pruned and only few iterations are performed in the first step for each initial configuration. The initial configurations of the models which show promising matching are used for further matching iterations to finish the search for precise matching.

Promising configurations can be recognized with the quality measure defined below. A termination criterion based on the error change, allows the termination of the ICP algorithm as soon as it has reached a minimum.

Matching quality

Since the ICP matching algorithm minimizes the coupling error d of the closest points, it seems obvious to use this measure to qualify the matching. But, experiments have showed that this criteria is often insufficient to discriminate bad and good matchings. For example, the mean μ of the coupling error d for the two matchings shown in Figure 4.12 differs only by 20%, which does not really reflect the large difference between the two cases where the gray object is in two completely different configurations.

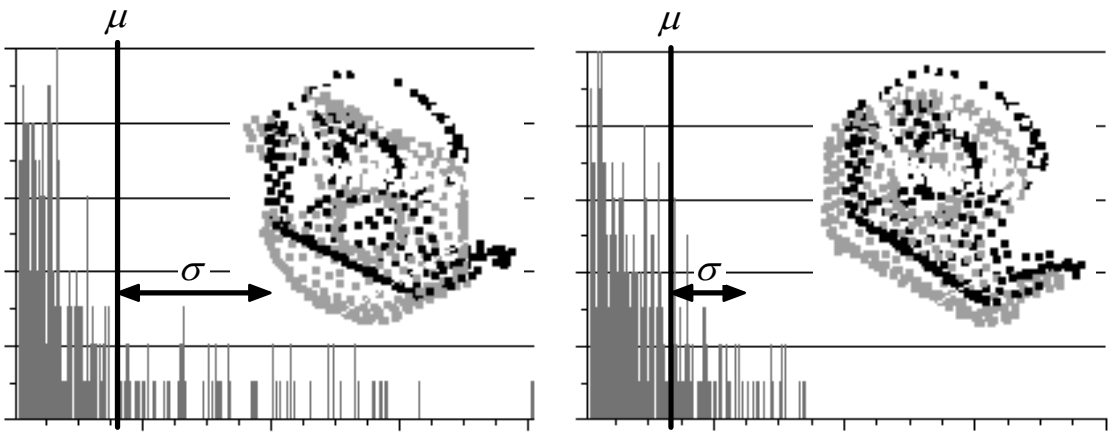


Figure 4.12 Distance histogram of closest points for two matching cases

The square distance histograms corresponding to the two matches show that the distributions of the square distances differ even if the mean values do not. Zhang proposed the inclusion of the deviation of the coupling distances to qualify a matching [ZHA]. For the cases shown in Figure 4.12, the deviation of the square distances is twice as large as for the bad cases compared to the good ones.

Therefore, a new matching error ε is defined as the sum of the mean and the deviation of the square coupling distances.

$$\begin{aligned} \mu &= \frac{1}{N}\sum_{k=1}^{N} d_k{}^2 \\ \sigma &= \sqrt{\frac{1}{N-1}\sum_{k=1}^{N}\left(d_k{}^2 - \mu\right)^2} \\ \varepsilon &= \mu + \sigma \end{aligned} \tag{4.5}$$

Cases with low error ε will indicate a promising matching and allow the pruning of the search space. The next section shows that promising initial configurations which lead to successful matching can be recognized after few iterations when observing the new quality measure.

Convergence quality

Actually, it is not necessary to do the same number of iterations for all defined initial configurations. The iterative matching can be abandoned as soon as the error ε changes only a little and the ICP algorithm has reached the minimum (see Section 3.6).

In the next experiment, the ICP algorithm matches the test, which is placed at several initial configurations, with the corresponding model. The iterations are stopped if $\varepsilon_{i-1} - \varepsilon_i < \delta$ is satisfied. When all configurations are checked the one with the lowest error ε is selected for more iterations to obtain precise matching.

Figure 4.13 shows that an increase of the error change threshold δ reduces the mean number of performed ICP iterations. If the error change threshold δ is set to high, the ICP algorithm will not reach the minimum for all initial configurations. A bad initial configuration which by chance has the smallest error and will be selected wrongly.

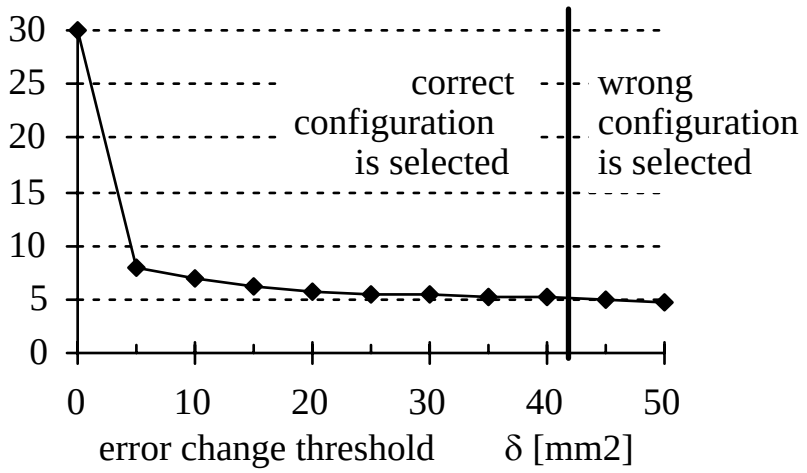


Figure 4.13 Fast initial configuration selection with error change

The mean number of needed iterations has already dropped significantly for a small error change threshold. This allows an increase in recognition speed of up to a factor of six to be gained without the risk to select a wrong initial configuration.

4.5.5 Recognition acceleration

At a first glance the necessity of multiple initial configurations seems to introduce many overheads. But since the range of successful convergence for the ICP algorithm is relatively large, the number of initial configurations can be kept low. Also, the search can be pruned by using an appropriate matching quality measure as shown in the previous section.

Other advantageous characteristics of the ICP matching algorithm, such as fast convergence after few iterations and low sensitivity to data reduction, are presented here and can be used to reduce the recognition time.

Coupling weights

As introduced in Section 3.5.3, the possibility exists to weight the couplings established by the closest point search for the matching error minimization. Here, this feature is used to accelerate the convergence of the ICP algorithm. The following weight function $w_k(d_k)$ is proposed (see Figure 4.14).

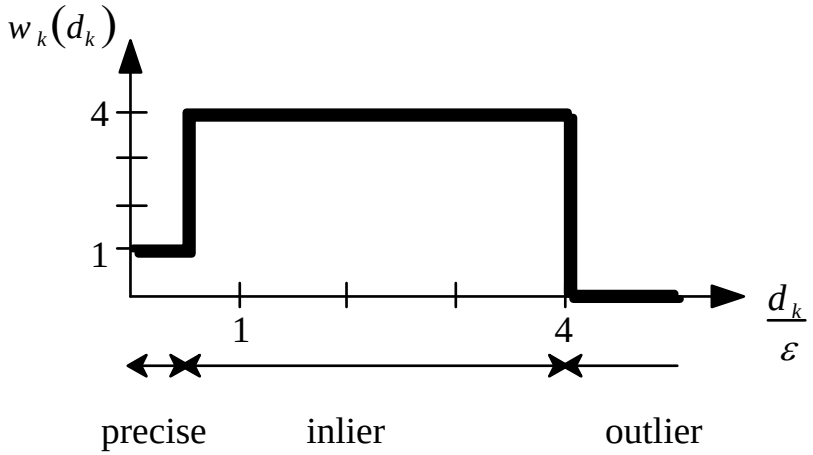


Figure 4.14 Weight function for closest point couplings

The above weight function attributes different weights w_k to every coupling according to the size of the quotient of the coupling distance d_k and the matching error ε defined in (4.5). Actually, the error ε calculated at the previous iteration is used. This function omits outliers and attributes a low weight to points which already have a near closest point. This results in a fast convergence since points which are still far apart get more influence and outliers do not disturb the matching error minimization.

As the matching error ε gets reduced after every iteration the weight function adapts itself and inliers are still detected correctly when the matching gets more precise.

The thresholds which separate the classes of the precise, inlier and outlier couplings in the weight function are determined empirically.

Other authors proposed soft transitions from one class to another and use fuzzy functions [KRE].

Figure 4.15 shows the evolution of the matching error over several iterations for a typical ICP matching converging successfully. The introduced weight function allows the reduction of the number of iterations needed to reach a minimum error by 50%.

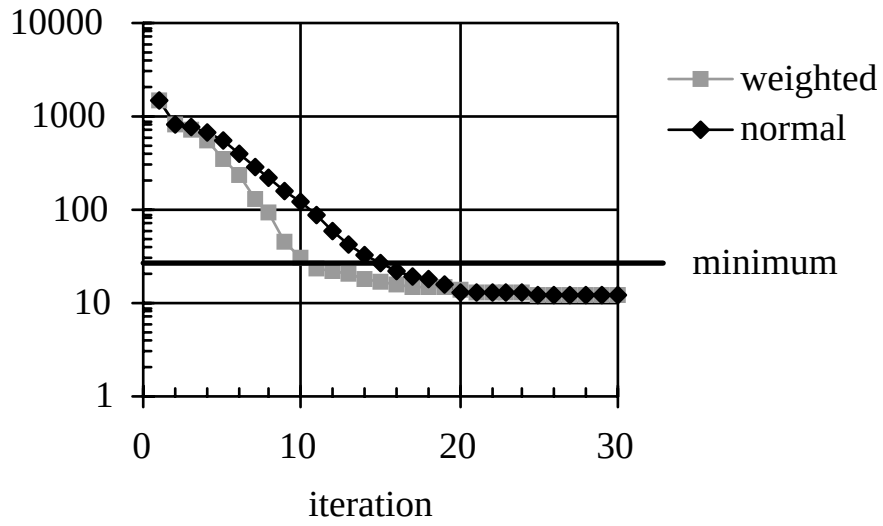


Figure 4.15 Matching acceleration with weighted couplings

Even if the convergence of the ICP algorithm using weighted couplings can not be proven (see Section 3.5.4), the experimental results show convergence. Since the ICP algorithm reduces the matching error, the outlier threshold of the weight function of Figure 4.15 decreases in general between two iterations. For this reason, most outliers never become inliers and the weighted ICP algorithm is well-behaved [ZHA].

Surface subsampling

Further reductions in calculation time is possible by reducing the number of points representing the surfaces (see Section 3.4.3). The next experiments with a 2D puzzle piece show that the ICP algorithm still converges successfully even for large subsampling factors.

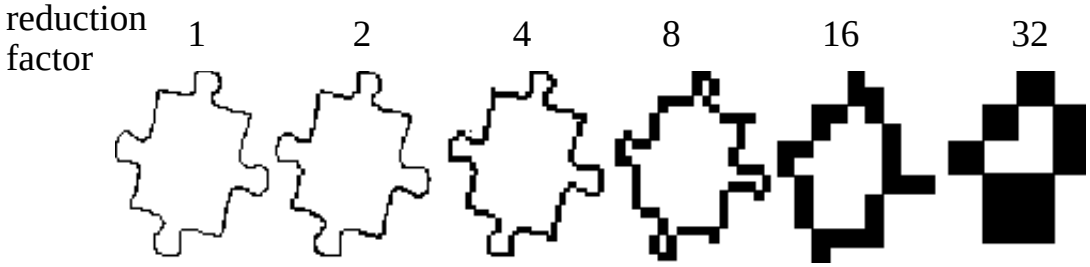


Figure 4.16 Subsampled puzzle piece

Two identical pieces of Figure 4.16 which have the same subsampling factor are matched from different initial configurations. One piece is rotated at several angle steps around the other. The final matching error for all initial configurations is plotted in Figure 4.17.

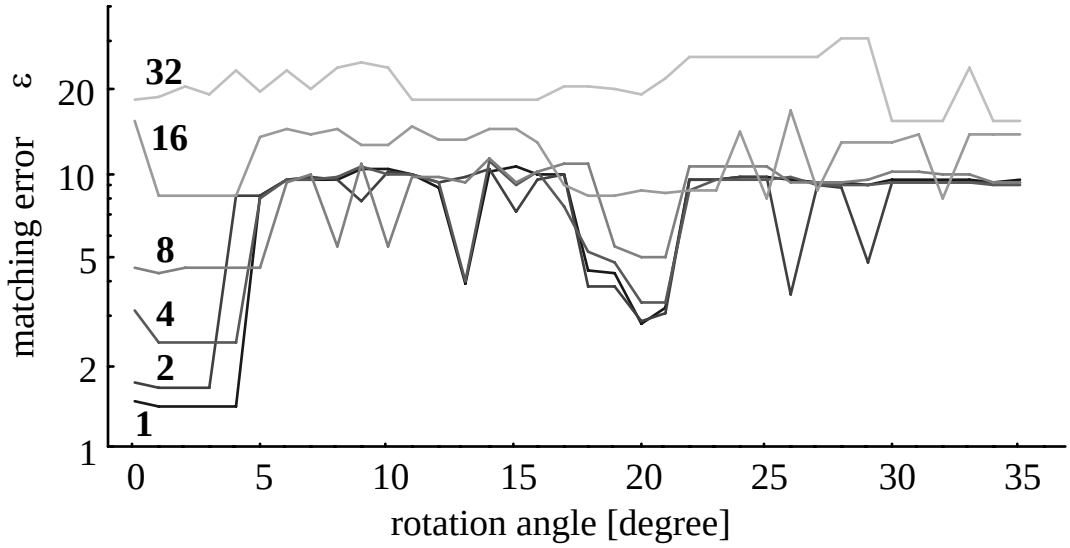


Figure 4.17 Matching errors for different reduction factors

The promising initial configurations can only be recognized reliably if they have a significantly smaller matching error than the other configurations. Figure 4.17 shows that for the reduction factors 1, 2 and 4, the first three initial configurations, for which the ICP algorithm converges successfully, have a much smaller error margin than the other configurations. A reliable selection of the promising initial configurations is no longer possible for larger reduction factors.

The computation cost of the ICP algorithm decreases linearly with the product of the test and model subsampling factors. The implemented

object recognition system reduces the test surface by a factor of two. The full surface resolution is only used for precise matching.

4.6 Recognition results

This chapter presents the results of several experiments performed with the proposed object recognition setup and ICP enhancements presented so far.

To verify the validity of the selection of the set of RIC according to Section 4.5.3, a series of recognition experiments is performed. It involves the three toy object models known as duck, fish and swan and ninety tests each containing a view of the toys in an arbitrary pose (see Figure 4.18 for some examples).

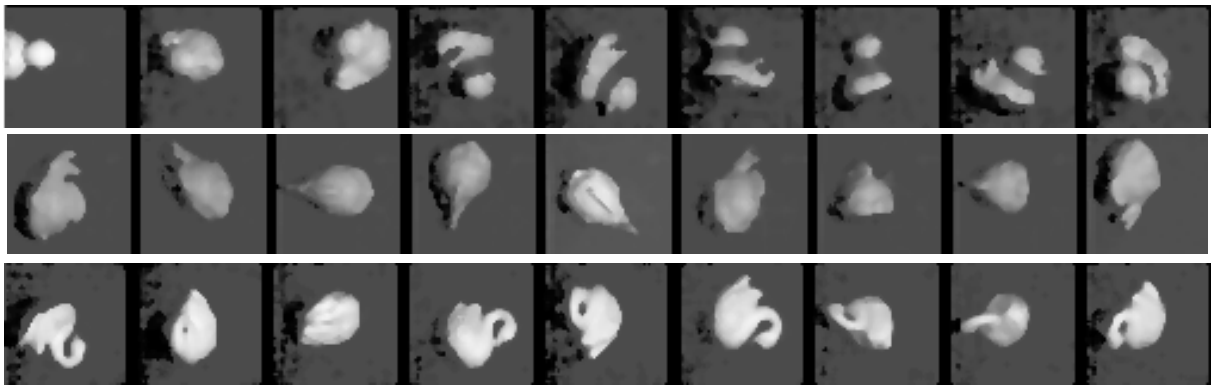


Figure 4.18 Subset of ninety test configurations

The recognition experiment evaluates the recognition performance as a function of the sets of RIC characterized by a different numbers of view points V and view axis rotations S. A specific set of RIC is run three times with all tests and the mean of the recognition results is calculated since the view point distribution is initialized randomly.

An object is recognized if the matching error ε of a test with a model falls below a given threshold. Since this threshold is low and based on geometric correspondences there are no false classifications (confusions) at all. The result is thus either correct recognition or rejection.

Figure 4.19 compares the RIC setups characterized by (V, S) values. Solid bars represent the recognition rate obtained for the ICP algorithm launched from different initial configurations while the gray curve reports the computational cost which is proportional to $V \cdot S$.

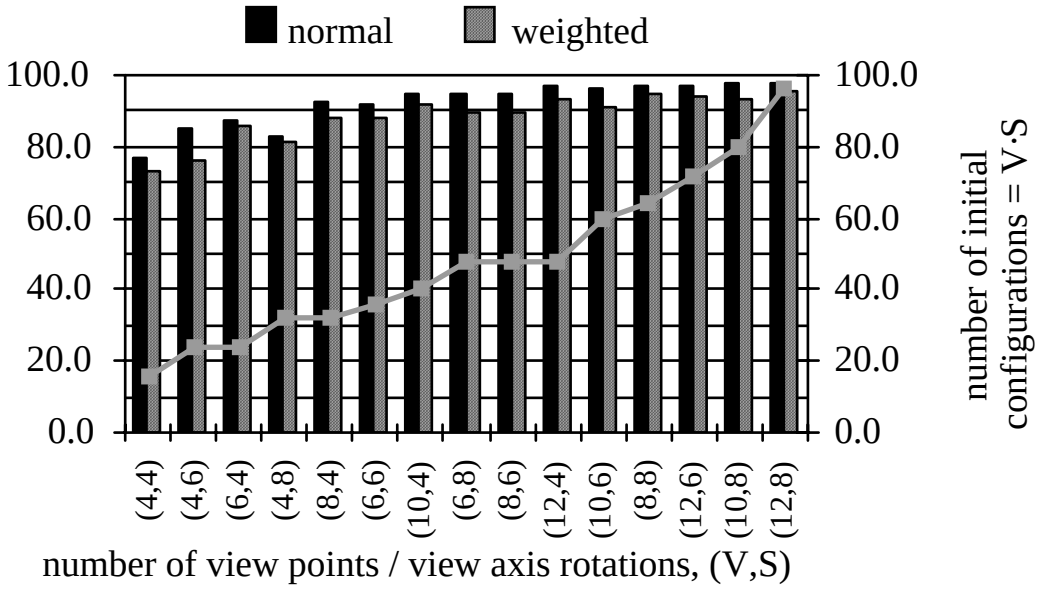


Figure 4.19 Recognition performance and cost for different setups

Maximal recognition rate of about 97% is reached for a set of RIC *(V, S)* = (12, 8). But, similar recognition rate is already obtained at less cost for *(V, S)* = (12, 4). By looking at the setups *(V, S)* = (8, 8) and *(V, S)* = (12, 4), it can be seen that a RIC with a higher number of view points obtains similar performance with much less view axis rotations. The obtained experimental results confirm the design rules of Section 4.5.3.

Almost similar recognition performance is obtained when the closest point couplings are weighted using the function of Figure 4.14. However, the number of performed iterations is reduced by about 40%.

4.7 Demonstration platform

The programming of a robot task in a virtual reality robotics environment (VRRE) is selected to demonstrate the successful working of the developed hybrid recognition system.

4.7.1 Introduction

The programming of a robot is often a tedious task which can be greatly improved in user friendliness by using virtual reality interfaces. An approach using virtual reality has been selected to overcome this problem. The description of the robot motion is done interactively in a virtual space representing the real world setup [NATa]. This allows an off-line programming and verification of the robot tasks.

A prerequisite for the success of this approach is a faithful correspondence between the virtual and real worlds. It means that changes in the real world must be fed back into the virtual environment by means of sensing devices. If a 3D vision system continuously updates the virtual world or part of it, the operator doesn't need to do it laboriously from a camera image of the real world.

To implement this vision task the above hybrid 3D vision system is proposed and implemented for the following assembly task. A robot has to be programmed to assemble the three tape dispenser parts shown in Figure 4.20. A 3D scanner and a video camera observe the working environment. The hybrid 3D vision system recognizes the objects and constructs a virtual world representation of the scene. This virtual world now allows the operator to program and verify the assembling task without having to move the real robot. Once the task is programmed it is sent to the real robot which executes it.

Figure 4.20 Scotch dispenser parts to be assembled

4.7.2 Scene acquisition

The object surfaces are acquired by a 3D scanner called BIRIS and produced by VITANA. This 3D scanner belongs to the class of laser line

triangulation scanners (see Section 2.4.1). It delivers a one-dimensional range profile. To obtain a complete range image the sensor is moved along an axis perpendicular to the profile. The successive measurements form a 2D range image which allows direct access to the 3D geometry of the scene. Figure 4.21 shows a range image of three tape dispenser parts acquired with the BIRIS range finder.

Figure 4.21 Range image of BIRIS scanner

4.7.3 Hypothesis generation

The hypothesis generation module uses the ICP algorithm together with the recognition setup of Section 4.5 to recognize the object type and pose. The hypothesises are verified afterwards by using the intensity image.

This section presents and discusses the SIC-maps of the scotch dispenser parts. These maps are used to build an optimal recognition setup (see Section 4.5.3)

SIC-maps of scotch dispenser parts

The SIC-maps are calculated for three different views of every object of Figure 4.20. For every initial configuration the ICP algorithm is launched, running enough (40) iterations to ensure convergence. An initial configuration is labeled as successful if the final matching error falls below a fixed threshold. An appropriate threshold is simply derived from the observation of the matching behavior of a number of cases. For this experiment no data reduction or coupling weighing is performed. The test shapes consists of about 150 points whereas the models contain around 400 points.

The resulting SIC-maps of the three scotch dispenser parts are presented in Figure 4.22, Figure 4.23 and Figure 4.24 for the zenith angle ranging from 0° to 80° at steps of 10°.

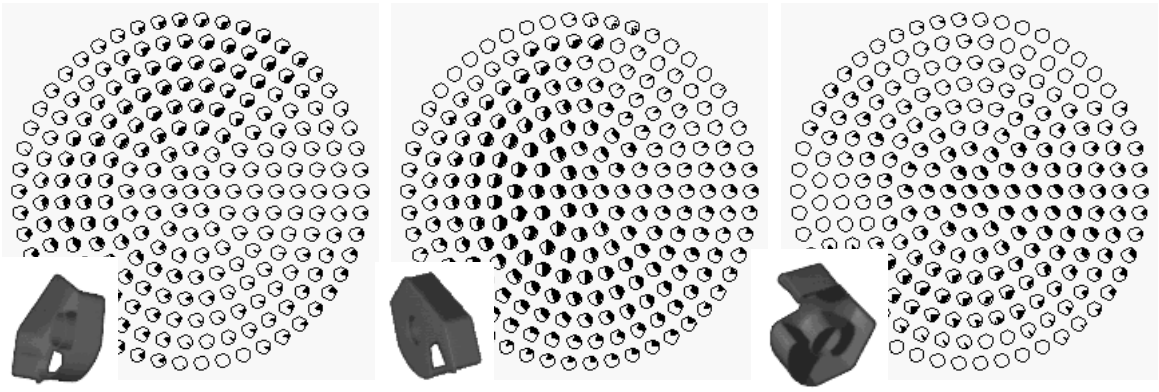


Figure 4.22 SIC-maps for three views of the scotch base

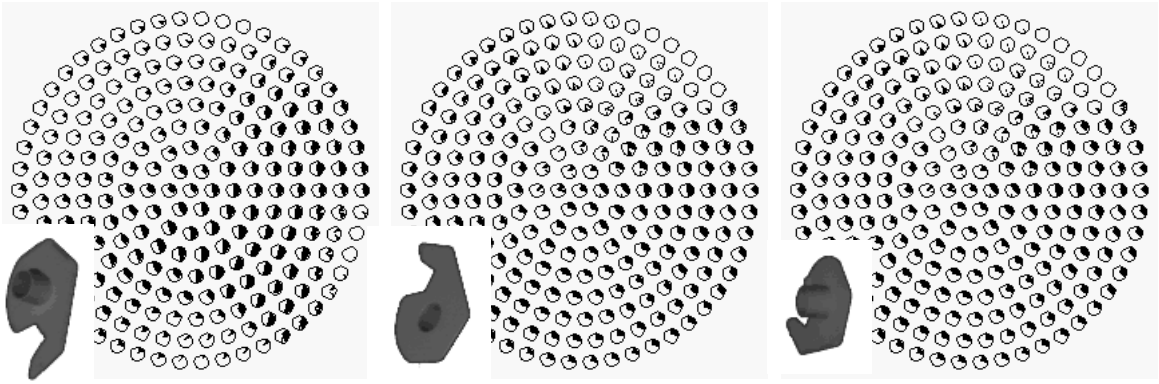


Figure 4.23 SIC-maps for three views of the scotch cover

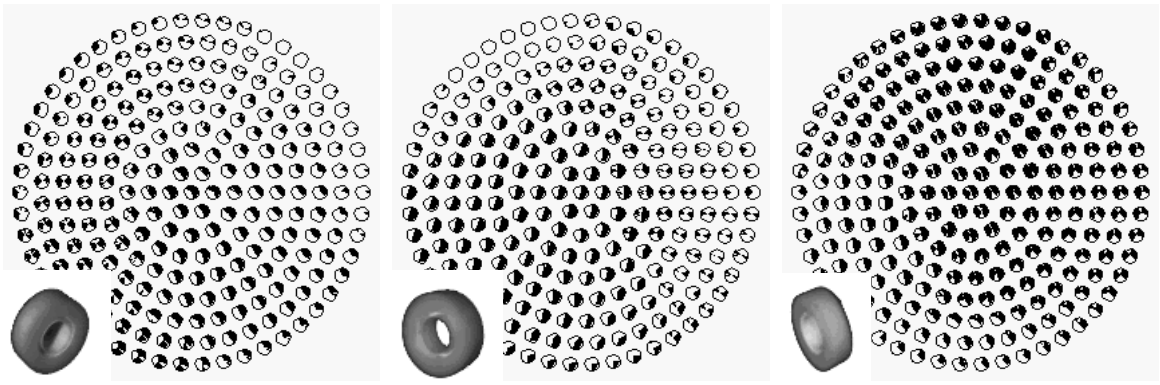


Figure 4.24 SIC-maps for three views of the scotch tape

SIC-maps discussion

Compared to the SIC-maps of the toy series in Figure 4.7 which were characterized by an approximately zenith centered SIC-range, the new SIC-ranges are characterized by a strong off-zenith dominance. Comparing for example the three scotch base SIC-maps (see Figure 4.22), the dominant part of the SIC-range is found respectively in the NW, SW and SE regions on the SIC-map. The SIC-maps for the other two scotch parts show similar behavior.

The second observation refers to the inter-object and intra-object variability. If intra-object variability is smaller than inter-object variability, it would be expected that some model specific features existed. All results obtained in this series reveal a comparable intra-object and inter-object variability. Thus, there are no specific features that could be used to improve object recognition.

Notice here that the scotch tape displays a very dark SIC-map and seems therefore to behave differently. This is in fact not the case because the darkening of the SIC-map has a different cause, namely the existence, with this scotch tape, of two optimal matching poses due to the object symmetry.

Optimal recognition setup

All measured SIC-maps exhibit a rather large SIC-range. To give a partial idea of its size, again the notation of a rectangular SIC-range as defined in (4.1) is applied. The rectangular SIC-range does not necessarily have to be

centered at the zenith of the SIC-map. Therefore, a manually estimated range which is roughly valid for all scotch parts is SIC(40, 60). In order to obtain successful recognition for a maximum number of different views, 12 view points and 6 rotations per view point are necessary as RIC in the recognition setup.

4.7.4 Robot assembly

The object recognition provides the precise poses of the database models in the real world scene. Using the generated object hypotheses, the scene can be modeled and continuously updated on a virtual world. Again the same model representations from the model database can be used, since VR modelers can work with colored triangle meshes.

Task programming

The recognized parts appear in the virtual world and allow the user to program an assembly task. Figure 4.25 shows two views of the assembly task programmed in the virtual reality environment.

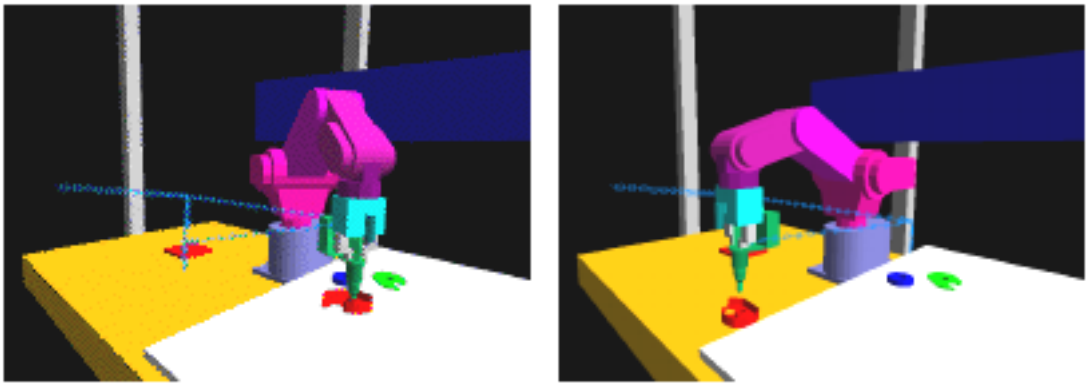


Figure 4.25 Robot task programming in the virtual world

Here, the robot has to grasp first the base part of the dispenser and to place it on an assembly template. Then, the tape roller is inserted into the base part and finally the cover closes the tape dispenser. A special gripper helps to grip the objects in their holes [NATb]. The real robot is only used for the final task execution. This saves energy and robot resources.

Task execution

Once the programmed task has passed verification by simulation in the virtual world it is downloaded via a serial link to the real robot. The robot now executes the same task as shown in Figure 4.26.

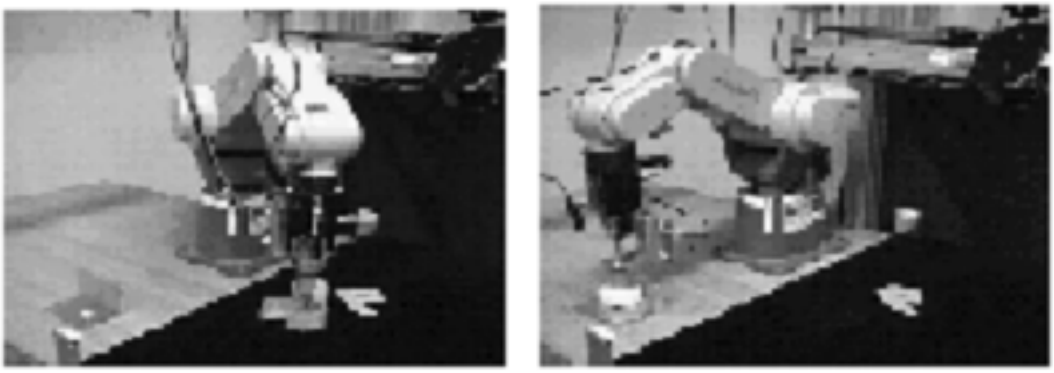


Figure 4.26 Robot executing the task

4.8 Conclusions

This chapter investigates the application of geometric point matching to object recognition. There are conclusions to be drawn on several aspects.

The proposed initial configurations for the poses of the test and model surfaces allow the successful application of the ICP algorithm without having a priori knowledge about the test pose.

Several experiments establish the quantitative nature of the dependence of the recognition performance on the selection of the number of initial configurations. SIC-maps are proposed to describe these dependencies. They are presented and analyzed for several free-form objects. The knowledge of the object SIC-maps allows a recognition system to be designed with an optimal number of initial configurations. If several models are involved in the recognition process, then it is advantageous to establish an optimal number of configurations separately for each model. Recognition experiments on a large number of scenes confirm the validity of the system design rules.

A reliable matching quality measure and an ICP termination criterion observing the error change of successive iterations allow the minimization of the number of iterations needed to search for the correct

model. The recognition is accelerated furthermore by data subsampling and weighted couplings.

The implemented object recognition system is successfully tested on free-form toys and in the frame of an assembly task of tape dispenser parts, where vision helps to update the virtual world by recognition and localization of the parts in the real environment. The vision algorithms can directly use the CAD models used for the virtual world construction.

Finally, the experiments performed with the system show the very appealing advantages of the applied methods: to treat free-form objects and to integrate them well into virtual reality environments.

Chapter 5
Object digitizing

Object digitizing aims on building a 3D representation of a real world object. It is characterized by the registration of partial views of an object surface. Once the different surface parts are aligned, they are fused into one common representation for the whole object surface.

Geometric point matching is proposed in this work to register the different surface scans. In contrast to the object recognition task the matched surfaces do only partially overlap.

The presented work has been published in [HUGe] [SCHe] [SCHf] [SCHk].

5.1 Introduction

Digitizing systems scan an object from different points of view and integrate the different acquisitions into one representation. They allow a virtual representation of a real world object that can be imported in CAD systems to be obtained.

Digitizing systems

The increasing use of virtual object representations for various applications creates a need for fast and simple object digitizing systems. Applications such as teleshopping, rapid prototyping and object

recognition need a virtual representation of the 3D geometry of real world objects.

The following object digitizing approaches exist:

CAD modeler: Objects are constructed within a modeler combining different geometric primitives such as boxes or spheres. Usually the data points are entered with the help of a visual graphic interface. The object dimensions are measured manually or with contact probes.

3D scanner: 3D scanners give direct access to the geometric information of object surfaces. They allow an accurate and dense digitizing of an object surface at low cost and high speed. Special software is needed in order to combine the different object scans.

3D scanners get used more and more since model construction with a modeler is a quite tedious task especially for objects of free-form shape.

However, since 3D objects self occlude, one acquisition captures only a subpart of the entire object surface. Therefore, a need to combine several scanner views into one unique object representation exists. The combination of the different acquisitions is straightforward if the object is moved into a well known coordinate system like a turntable: the relative transformation between the pose of two acquisitions is known. However, complex positioning systems are needed to sense the complete surface of an object and to keep track of its pose at the same time.

Proposed solution

To avoid the need for positioning systems, this work proposes a digitizing system which registers object views from unknown poses. The idea is to match the surfaces based on the sole knowledge of their geometric measurements.

The presented digitizing system captures views of a real world 3D object and then registers and integrates them into a virtual model representation. The following steps have to be performed in order to combine the acquired surfaces:

– An object to be digitized is placed in different unknown poses on the acquisition field. The visible surface is sensed by a 3D scanner. Every acquisition is represented by a triangle mesh (see Section 2.5).

– The different surface measurements now have to be registered. An interactive interface allows the operator to roughly align the two surfaces in 3D space. Then, the precise surface registration is calculated by the ICP algorithm.

– In the last step called mesh fusion, the aligned triangle meshes are combined into an unique triangle mesh representing the entire object surface.

The following sections will focus on the registration task whereas the mesh fusion has been presented in [SCHe] [SCHf] and [JOS].

5.2 Digitizing system architecture

The digitizing system consists of two main blocks: view digitizing and view integration (see Figure 5.1 for an overview). The view digitizing block generates a virtual view of the observed object surface. The view integration block iteratively integrates each new virtual view in the virtual model under construction. This allows an incremental construction of the virtual model.

The view digitizing block measures the points of the visible object surface, filters the noise present in the measured data and triangulates the surface points. The resulting output is a triangle mesh representing a virtual view of the real object. The implementation of the modules used for view digitizing has been discussed in Section 2.5.

The view integration block combines the virtual views and builds one virtual object becoming an entire model of the real object. The geometric transformation between the reference frames of a new acquisition and the virtual model is not known a priori and determined by the view registration block. The mesh fusion module combines the registered triangle meshes into a new global mesh covering the union surface of the single meshes.

The view registration block aligns the different views combining interactive rough registration and automatic registration. The interactive rough registration provides the matching estimate used for the automatic registration which is performed by the ICP algorithm. Since no external measurements of the object pose are available, the automatic registration system relies on the object surface geometry in order to register it with

the virtual model. This assumes that the virtual model and the new object view have at least some surface area in common which allows correspondences between them to be established.

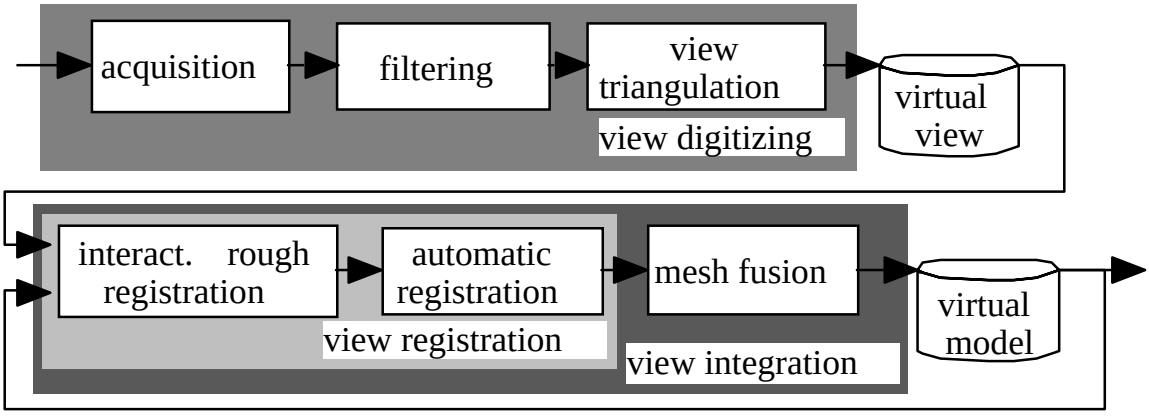


Figure 5.1 Digitizing system architecture

5.3 Comparison with other work

Similar digitizing systems have been presented by [TUR] [SOUc] [PIT] [HILa]. All of them use modified versions of the ICP algorithm in order to register the different surfaces. However, they differ significantly in how they fuse the registered object views. Some systems keep all data points in the overlapping area and retriangulate the mesh. Other authors propose the removal of one of the overlapping part (redundant data) and the fusion of the remaining meshes (see [HILb] and [SCHf] for an exhaustive comparison).

The system presented in this work belongs to the second group. It erodes the overlapping surface area of one surface and links the remaining mesh with the other surface. The proposed mesh fusion algorithm benefits from the closest point relationships established during the geometric point matching [SCHf]. There is no need to run an extra routine to erode overlapping surfaces and to detect the surface frontiers as is done in other work [PIT] [TUR].

Most digitizing systems employ a graphic interface which allows a human operator to enter a pose estimate to obtain successful convergence for the ICP algorithm. So does the system presented here. Human perception easily identifies corresponding surface parts for any object type and shape. As far as we know, there is currently no digitizing system doing the surface matching completely automatically.

Another goal is to improve the surface registration of surfaces where the simple geometric closest point criterion fails to establish reliable correspondences. Several authors proposed the extension of the ICP algorithm to integrate additional features, for example surface color or surface orientation in the closest point search [JOHa] [MAR] [GODb] [FELa] [GODa] [STOa]. Here, a flexible closest point search is proposed which can use several features such as geometry, color and surface orientation in any combination (see Section 3.4.2).

In contrast to other work [FELa], the closed-form error minimization based on quaternions (see Section 3.5) is maintained when the surface orientation is used in the closest point search. Although convergence can not be proven yet, empirical investigations show successful registration results.

5.4 User interfaces

All modules of the digitizing system can be controlled from the same graphic user interface shown in Figure 5.2. It contains menus for the 3D scanner control to import the acquired object scan views and to activate the different view integration modules of the digitizing system.

The acquired data is rendered on a fast 3D graphics work station which updates the virtual scene at screen refresh rate. Furthermore, the computer monitor can work in stereo mode and be viewed through stereo glasses. This gives improved depth cues and allows the surfaces to be aligned better.

The pose of the surfaces can be changed in all degrees of freedom of translation and rotation with a space mouse (see Figure 5.2).

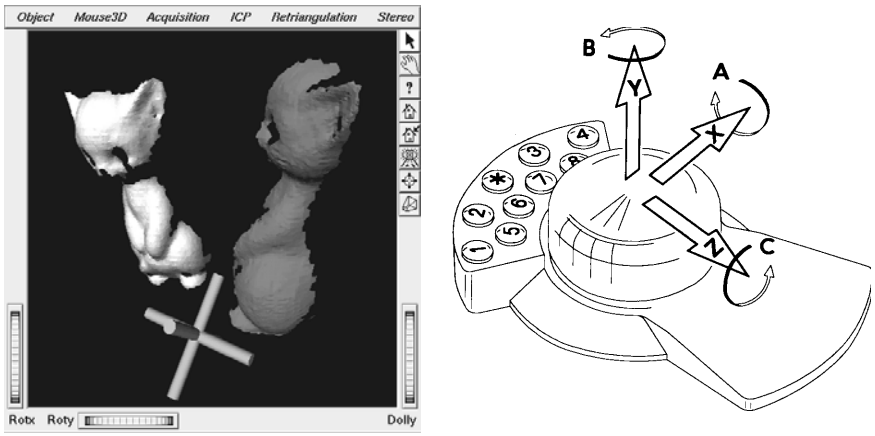


Figure 5.2 Graphic and haptic interfaces of the digitizing system

A shell window reports the matching errors and the amount of surface overlap. This allows the operator to decide if further matching is needed or if the desired matching precision is reached.

5.5 Interactive rough registration

The ICP algorithm which registers two surfaces needs an approximate match in order to converge successfully (see Section 3.7). There are various reasons why the ICP algorithm is not applied to an exhaustive search among several initial configurations.

First, the surfaces contain hundreds of thousands of data points since precise matching is needed and therefore the investigation of several initial configurations would take too much computation time.

Secondly, the initial configurations defined for the object recognition task assume that one surface is an entire subset of the other (see Section 4.5.1). This is not the case in object digitizing.

Alternative methods use extracted features or markers in order to align two surfaces. Feature extraction may fail when dealing with free-form objects (see Section 2.2.2). Markers modify the object appearance and disturb the digitizing of objects with color texture.

Human perception easily identifies the corresponding regions of two surfaces. The presented interactive graphic interface permits a human operator to enter a pose estimate for two surfaces. However, even sophisticated object rendering and pose manipulation hardware as presented in Section 5.4 is not sufficient to align the object surfaces precisely. In fact, there is no measure apart from the visual feedback

indicating the quality of the surface matching. Therefore, the interactive interface only provides a rough pose estimate to be used as an initial pose for the automatic precise registration performed with the ICP algorithm.

Figure 5.3 shows an example of two roughly aligned surfaces used as initial configuration for the automatic registration.

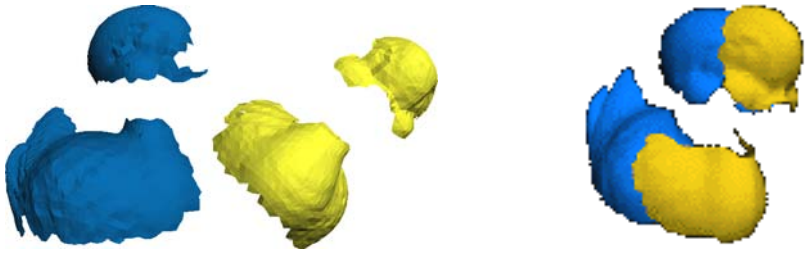


Figure 5.3 Two roughly aligned surfaces representing views of a duck toy

5.6 Automatic registration

Using the available pose estimate, the ICP algorithm finalizes the registration of two surfaces at high precision.

Figure 5.4 shows several iterations of the automatic registration started from the surface configuration shown in Figure 5.3.

Figure 5.4 Iterations of the geometric point matching

The object views to be assembled need common surface regions with enough geometric structure in order to allow the automatic matching to converge to a correct solution. For two neighbor views, 30% to 50% of common surface has been observed to be an adequate amount.

The next sections describe the implementation details of the ICP algorithm adapted for automatic registration.

5.6.1 Surface overlap detection

As the ICP algorithm is applied to two surfaces which only partially overlap, each surface contains data not present in the other and convergence is often unsuccessful. The ICP algorithm needs to be modified as proposed in [TUR] [GODb] [ZHA] and implemented as follows.

In order to couple only the surface areas which potentially overlap, closest points which are too far apart are not considered to be corresponding points and are marked as invalid. They have no influence on the matching error minimization (see Figure 5.5).

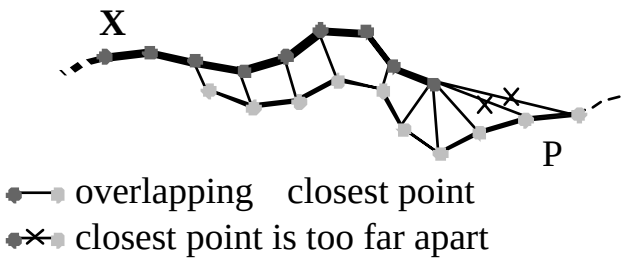


Figure 5.5 Closest points coupling for partially overlapping surfaces

This removal corresponds to a weighing of the closest point couplings as defined in Section 3.5.3. The following function $w_k(d_k)$, which uses the geometric coupling distance d_k as input, attributes a weight to every coupling:

$$w_k(d_k) = \begin{cases} 1 & d_k < \tau_g \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

Closest points with a coupling distance d_k smaller than τ_g belong to the overlap area. If τ_g is set too large, surface areas which do not correspond will be considered as overlap. If τ_g is too small, no couplings remain at all.

In order to select a reasonable value for τ_g, the overlap threshold τ_g is adapted during the ICP matching as the corresponding areas are approached. It is set to a relatively large value at the beginning of the matching when the two surfaces are far apart and is then successively

reduced for the following iterations. As the surfaces become superimposed the value of τ_g is lowered to the sampling interval of the surface points. Figure 5.6 shows some typical values of τ_g over several iterations.

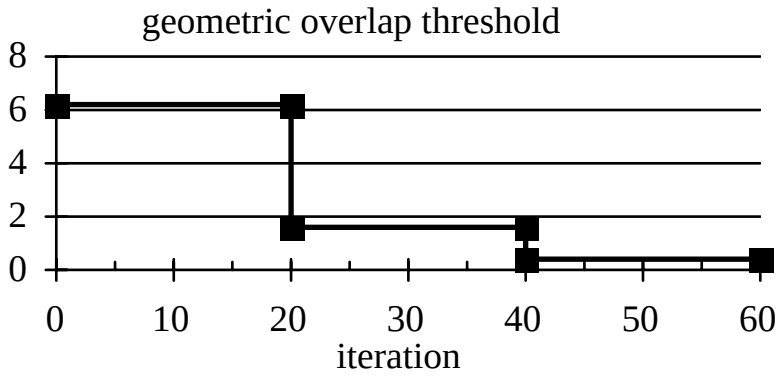


Figure 5.6 Typical values of overlap threshold over several ICP iterations

Note that, there is a link between the value of τ_g and the kD tree search. As only points which are closer than τ_g are considered for the error minimization, the fast closest point search done with a kD tree (see Section 3.4.3) can be directly initialized with $D_{\text{init}} = \tau_g$. Since τ_g is relatively small the recursive search using the kD tree is accelerated because few branches are inspected.

5.6.2 Precise closest point

In order to achieve the precise registration of two surfaces, a high point density on both surfaces would be needed. Since the input of the presented system is a triangulated surface, the ICP algorithm has access to the continuous surface geometry and not only to discrete surface points. Therefore, the vertices of one surface can be coupled with points on the triangles of the other surface as defined in Section 3.4.5. The fast closest point search using kD trees can be extended to handle triangles (see Section 3.4.5).

5.7 Multiple feature matching

The coupling between two surfaces performed by the closest point search is a key operation in the ICP algorithm. For some objects, the available surface geometry information is not sufficient to obtain a correct matching. Additional features as color or surface orientation are considered to establish the coupling.

An appropriate feature distance is defined for the color and orientation feature in Section 3.4.2. The different features are added to the coupling distance *d* (see (3.15)) and used for the search of the point which is closest to all features.

Feature range normalization

Since different features do not necessarily have the same value range, they need to be normalized to a common range. Otherwise one feature may have more influence on establishing the couplings than others. This section presents a method to select the normalization factors α_g, α_n and α_c of (3.15).

The basic idea is to normalize the different feature distances using the corresponding overlap threshold as defined in (5.1) for the geometric distance. This method has the advantage that the number of adjustable parameters used in the object digitizing system is kept at a minimum.

Therefore, an overlap threshold is introduced for every feature and adapted during the ICP iterations. Two surface points are considered to have the same orientation if the Euclidean distance between the corresponding normal unit vectors is below τ_n. The threshold τ_n is easily derived from the corresponding angle difference of two normal vectors. Two surface points are considered to have the same color if the Euclidean distance between the corresponding color vectors is below τ_c. The threshold τ_c is expressed by the maximal difference per color component measured as percentage of the component range (255 for this implementation).

Typical values, expressed in degrees and percentage, defining the overlap thresholds for orientation and color are plotted in Figure 5.7.

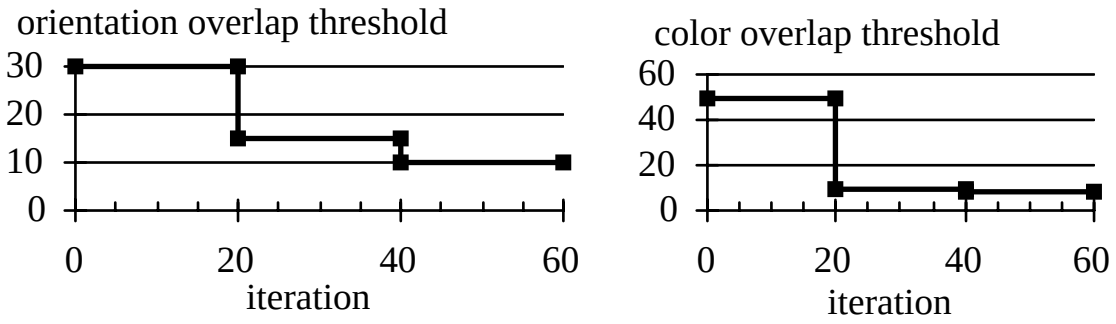


Figure 5.7 Typical values of overlap thresholds for color and orientation

If the normalization factors α_g, α_n and α_c are defined by

$$\alpha_g = \tau_g, \quad \alpha_n = \tau_n \text{ and } \quad \alpha_c = \tau_c, \tag{5.2}$$

the overlap distance ranges from 0 to 1 for every feature and the weighing function of (5.1) becomes

$$w_k(d_k) = \begin{cases} 1 & d_k < 3 \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

where d_k corresponds to the normalized total coupling distance of (3.15) integrating all three features. Actually, in order to strictly ensure that every feature is in the 0-1 range the value 3 in (5.3) should be replaced by 1. However better results have been observed with a factor corresponding to the number of features used in d_k (here 3).

kD tree with multiple features

The kD tree provides a fast closest point search for feature spaces of any dimension. It uses an Euclidean distance measure to find the point with the smallest distance for all features. Therefore, an Euclidean distance measure has been defined for the color (3.12) and the surface orientation feature (3.13). If all features are used, k equals nine since every feature is composed of three coordinates.

Convergence behavior

The aim of the following experiment is to measure the gain in convergence speed when using surface orientation or surface color for the

closest point search. Furthermore, the overlap detection is integrated in the ICP algorithm. Note that the matching error minimization still uses the method based on quaternions and finds the optimal rigid transformation which only reduces the geometric error.

The ICP algorithm matches the two colored surfaces of a toy rabbit shown in Figure 5.8. The evolution of the matching error *e*, equal to the geometric coupling error, is plotted in Figure 5.9.

Figure 5.8 Test and model surfaces of a colored toy rabbit

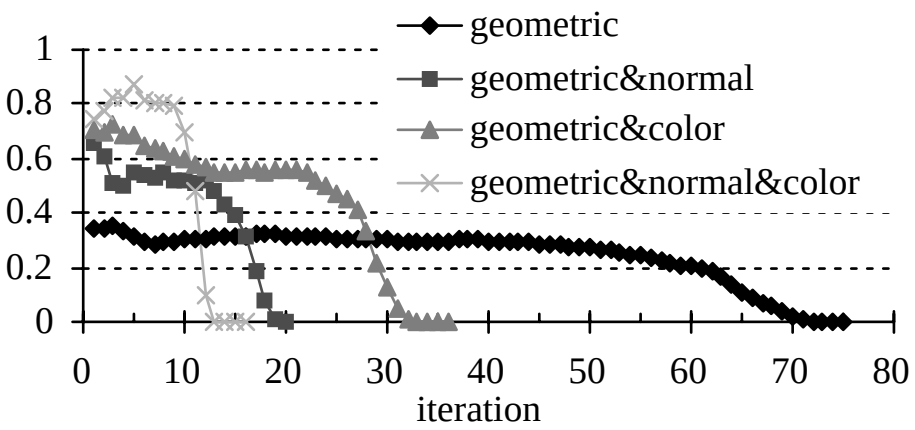


Figure 5.9 Convergence of ICP using multiple features

The ICP algorithm did not converge successfully if the geometric feature was not included in the coupling distance. Figure 5.9 shows that the more surface features are used the less iterations are needed for the matching to converge since the ICP establishes better couplings. If all three features

are used the number of iterations needed to reach the minimum is reduced by a factor six.

Although the convergence of the ICP algorithm can not been proved if the couplings are weighted or the surface orientation is used for the closest point search (see Section 3.5.4), the experiments in Figure 5.9 show successful convergence. However, the ICP algorithm does not converge monotonically to a minimum any more.

A further analysis of the results of Figure 5.9 shows that the absolute execution time of the ICP algorithm does not always decrease when using several features as shown in Figure 5.10.

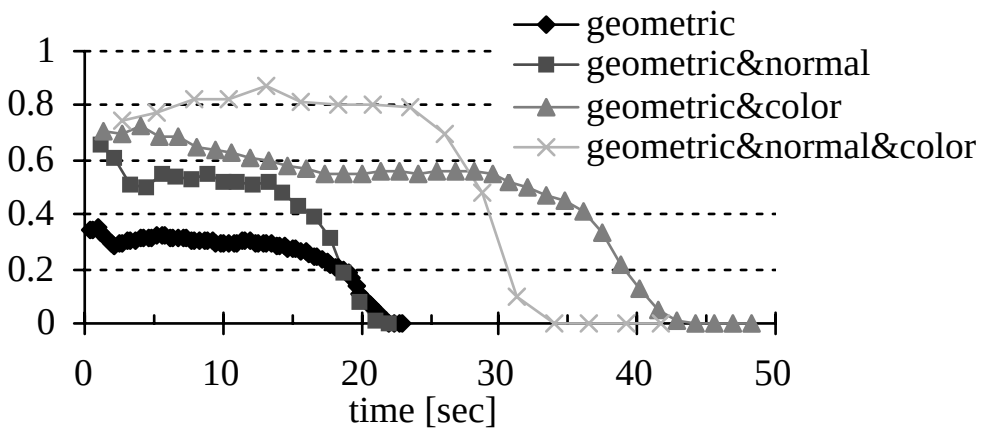


Figure 5.10 Absolute execution time of ICP using multiple features

An explanation is that the kD tree is only efficient for features which vary a lot and allows the early exclusion of most of the search candidates.

This is not the case for the color feature which has been quantified into few levels by the used framegrabber and also the rabbit in Figure 5.8 contains few really different colors. The total ICP execution time is about double if color is used even if the number of performed iterations is halved. The normal feature does not significantly improve the execution time either although it needs about a third of the iterations.

Finally, even if the number of iterations is reduced the execution time is hardly improved.

Optimal overlap threshold size

The following experiment investigates the variation of the execution time when changing the size of the overlap thresholds.

Here, the closest point distance includes the geometric and the normal feature. The ICP algorithm matches the two surfaces shown in Figure 5.8. Note that in this experiment, the size of the overlap thresholds is not changed during the ICP iterations.

Figure 5.11 plots the different matching errors for different values of the geometric overlap threshold τ_g. The normal overlap threshold τ_n is fixed to 40 degrees. A matching error below 10 mm^2 corresponds to successful matching.

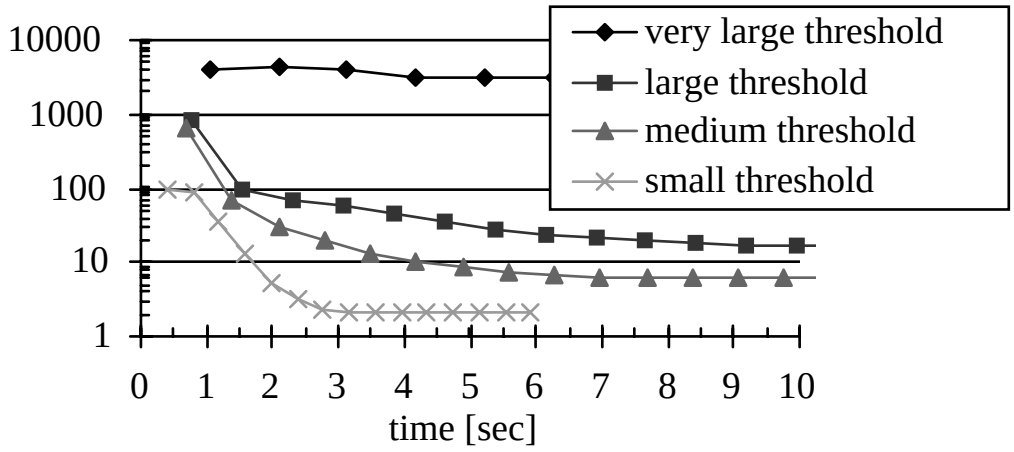


Figure 5.11 Matching convergence for different geometric thresholds

No successful matching is obtained for a large τ_g. As the feature distances are normalized by their corresponding overlap threshold, the normalized geometric distance is very small compared to the normalized normal vector distance. This results in the closest points couplings being established for points with similar normal orientations even if they are far apart from the point of view of geometry.

The convergence becomes faster and converges successfully when using a smaller τ_g. Best results are obtained for a geometric threshold which corresponds to the initial distance of the two shapes shown in Figure 5.8. If the threshold τ_g is reduced further, no couplings are established any more since the closest point search can not find any candidate in the search range.

These results lead to the following practical rules for the determination of the optimal size for the feature overlap thresholds:

– Choose the smallest geometric overlap threshold τ_g possible and lower it as the two shapes approach each other. This ensures fast convergence.

– The thresholds τ_n and τ_c for the other features should be larger than the variations due to the 3D scanner and framegrabber noise. An upper optimal limit has to be established by running several experiments, where lower thresholds result in the better use of the feature discrimination capability in the closest point search. The ICP matching converges faster, since fewer kD tree branches are inspected.

5.8 Advantages of multiple feature ICP

In the presented digitizing system an operator roughly registers two object surfaces. The ICP algorithm needs this estimate for successful convergence. The larger the SIC-range the rougher the pose estimate can be.

The following sections inspect the range of successful convergence of the ICP algorithm integrating several features such as color, texture and orientation.

5.8.1 Comparison of convergence qualities

Again the dedicated setup presented in Section 3.7.1 is used to measure the SIC-range. The results are plotted in a SIC-map showing the domain of successful initial configurations. Such SIC-maps are established for several surfaces and the closest point distances including the normal and color features.

Figure 5.12 presents the SIC setups for three experiments with the test shapes in the zenith configuration.

– Figure 5.12a shows the matching of a colored view *P* with a complete colored model *X* of a toy rabbit. Such configurations are encountered in object recognition or object inspection.

– Figure 5.12b shows the alignment of two partially (50 %) overlapping object views *P* and *X*. This disposition is observed in object digitizing where a complete model is assembled out of several views.

– Figure 5.12c shows two identical colored ball surfaces *P* and *X*.

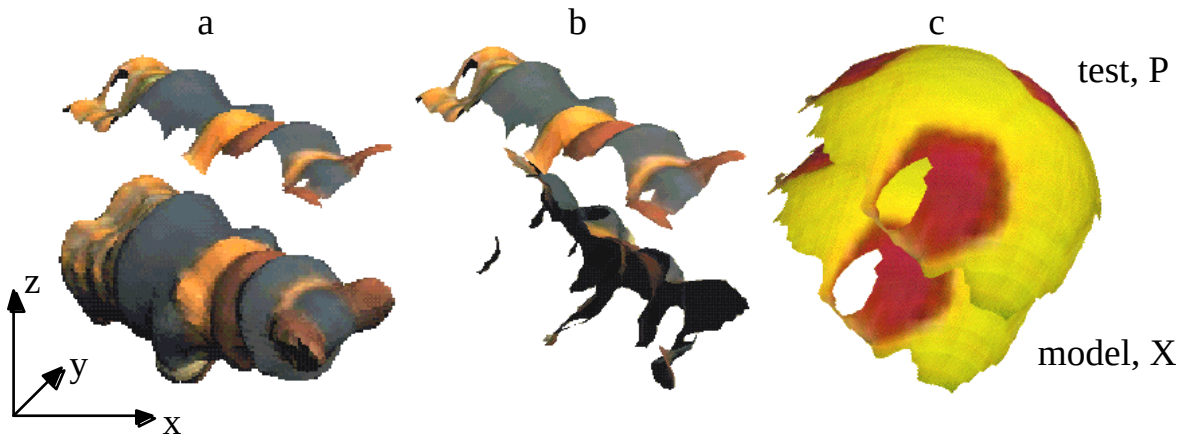


Figure 5.12 SIC-map setups to test different coupling distances

The resulting SIC-maps for the setups shown in Figure 5.12 are presented in the following three figures. The zenith angle is sampled at steps of 20 degrees.

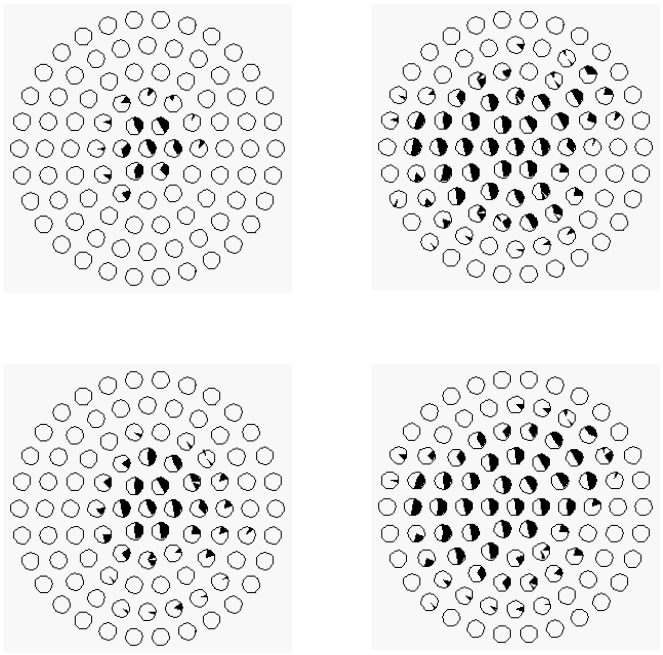


Figure 5.13 SIC-maps for experiment a in Figure 5.12

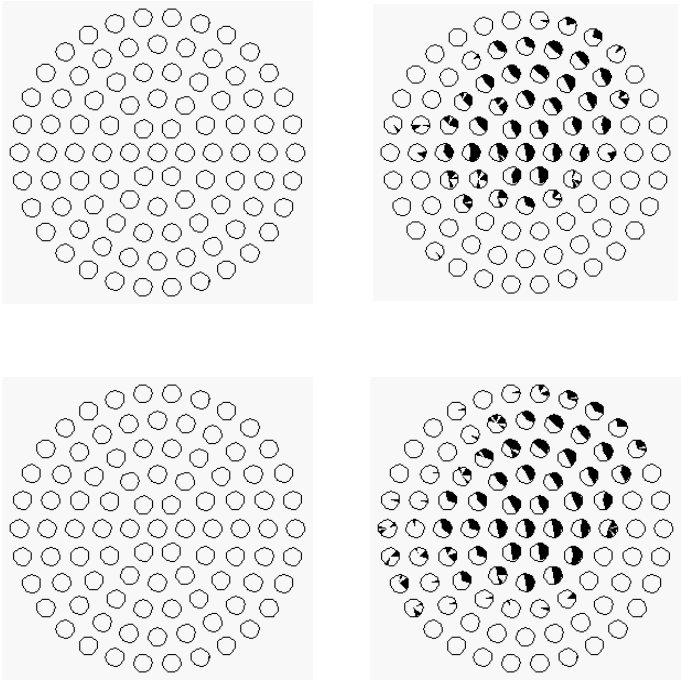


Figure 5.14 SIC-maps for experiment b in Figure 5.12

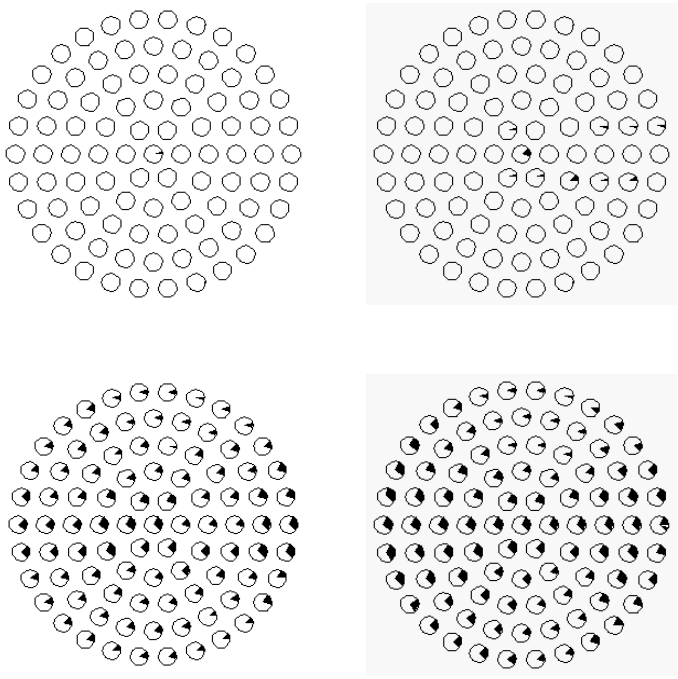


Figure 5.15 SIC-maps for experiment c in Figure 5.12

The resulting SIC-maps show that the range of successful convergence is considerably improved in all experiments when including color and normal features in the closest point search.

The ICP algorithm does not converge at all to the correct pose for the setups shown in Figure 5.12b and Figure 5.12c if only the geometric distance is used. Only the surface normal information permits a correct surface coupling and registration of the two object views of Figure 5.12b.

Since the surface in Figure 5.12c is invariant under rotation only the color information allows the color spots to be correctly aligned.

Table 5.1 gives an overview of the results obtained for the different features. The best matching results are obtained if all three features are combined.

	exp. a	exp. b	exp. c
geometric	poor	none	poor
geometric and normal	good	good	poor
geometric and color	fair	none	good
geometric, normal and color	best	best	good

Table 5.1 Convergence quality for different experiments of Figure 5.12

5.8.2 Automatic pose estimate

If an object surface has different color regions, a closest point coupling based exclusively on the color feature can lead to a first pose estimate.

Figure 5.16 shows an example of two partially overlapping views of a colored toy rabbit. The two views are first in a random pose. Then, a first ICP matching is performed using only the color feature to establish correspondences. The matching error minimization uses these couplings to match the two surfaces. The resulting match is not very precise but can be used to launch the ICP algorithm again using geometric and normal features this time. The final matching is successful.

Several experiments have showed that this approach works only in some exceptional cases and can therefore not replace the human operator entering a rough pose estimate.

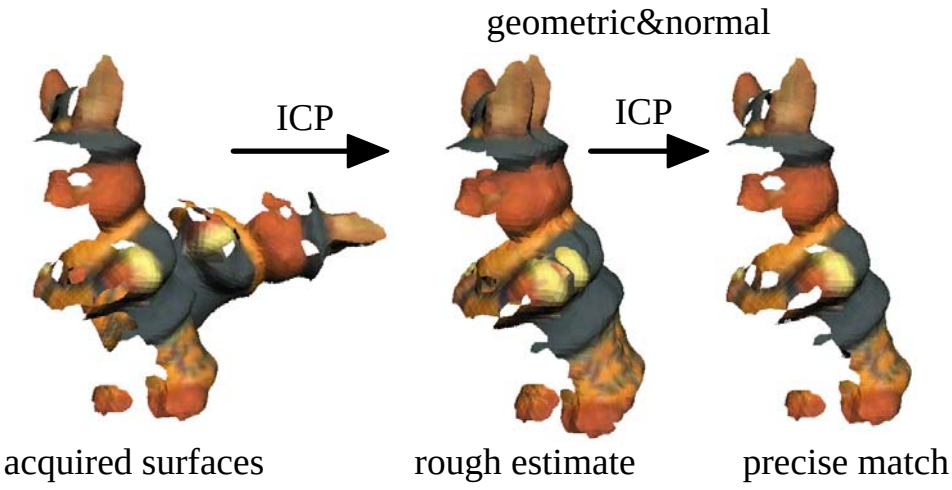


Figure 5.16 Pose estimate using only color feature in the coupling distance

5.8.3 Color texture matching

If an object surface is not sampled very densely, color images can be mapped onto the surface geometry in order to restore the color texture information. The resulting texture triangle meshes are presented in Section 2.5.

Section 3.4.5 presents a method to use texture images to establish closest point correspondences. Similar to the color feature the color texture can provide an escape where the surface geometry does not allow unique correspondences to be established. This is the case for example for the surfaces in Figure 5.17 representing two views of the same cup and having exactly the same shape.

Figure 5.17 shows how the color texture allows the correct registration of the color stripes painted on the cup. The correct matching is found in one iteration starting from a registration performed with the geometric ICP algorithm.

The texture windows have a width of nine pixels and about hundred vertices are correlated. The established couplings are used to find the optimal rigid transformation with the quaternion method.

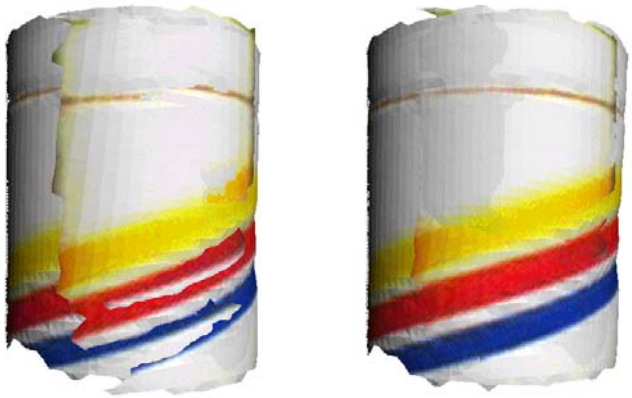


Figure 5.17 Geometric and texture matching of two views of a cup

The texture matching succeeds only for small corrections. Several experiments showed that is only useful to improve the registration of already quite precisely matched surfaces.

5.9 Gallery of digitized objects

The described 3D object digitizing system has been implemented and used to create virtual models from several real world objects. This section presents the results of three digitized free-form objects.

Table 5.2 shows the different object statistics: the number of points and triangles the object is composed of, the number of object views which have been registered to build the complete object and the information used to register the objects.

	number of points	number of triangles	number of views	ICP features
duck	5'667	11'327	10	geometry, orientation
brick	17'470	34'933	8	geometry, orientation
rabbit	3'601	7'174	11	geometry, orientation, color

Table 5.2 Digitized object statistics

The duck object shown in Figure 5.18 is of free-form shape. The surface has many concave parts and small details.

This example illustrates the potential of the system to perform reverse engineering on complex shapes. As the complete surface has been reconstructed the object can now be regenerated with processes like stereo lithography. This is useful to replicate artist work for the jewelry industry for example.

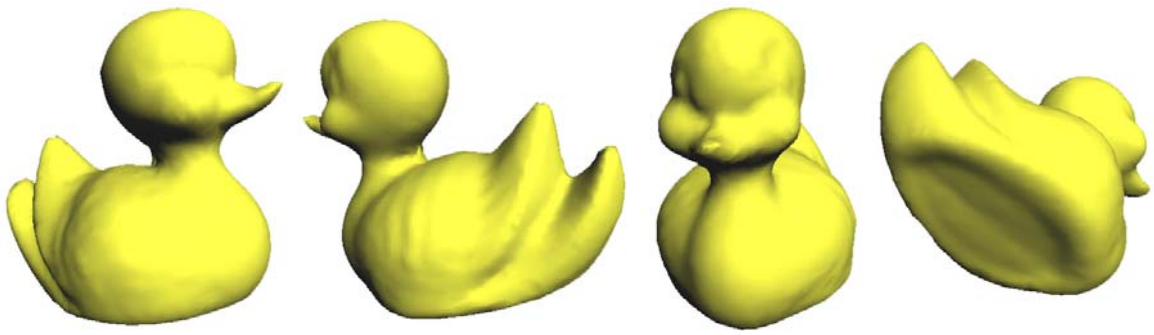


Figure 5.18 Toy duck

The antique brick has been lent by the archeological museum of Neuchatel. Again the complete surface has been precisely digitized in order to keep the engraved details of the brick.

Digitized models of museum objects could be arranged in a virtual museum and made accessible over the Internet. This allows precious objects from remote places to be inspected without danger of damage.

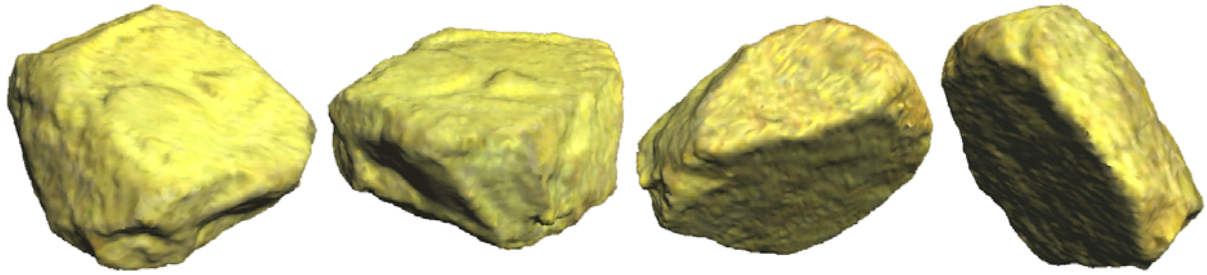


Figure 5.19 Antique brick

The rabbit objects shows the successful integration of different views of a colored object. The use of multiple features in the ICP matching permits the correct reconstruction of the color stripes on the pants of the rabbit.

Such digitized objects are of great interest for the multimedia industry. Realistic models can be obtained with little effort from a real world counterpart.

Figure 5.20 Colored toy rabbit

5.10 Conclusions

The presented digitizing system permits the fast construction of models of free-form 3D objects. It registers and integrates different object views acquired by a 3D scanner and does not need any a priori information about the object poses. The various object views are registered by a combination of interactive rough registration followed by automatic precise registration.

The ICP algorithm which registers precisely different object views runs at the core of the system. The proposed weighing function allows the application of the ICP algorithm to the registration of surfaces which only partially overlap. Closest point couplings which are not in the overlapping area are excluded from the matching error minimization.

A further improvement of the ICP algorithm is the use of surface color and orientation in the closest point search. Both features improve the surface matching where the surface geometry is not discriminative enough to obtain successful convergence.

Although convergence can not be proven for the ICP algorithm using weighted couplings or the surface orientation feature, several experiments and SIC-maps show successful matching results. Therefore, the error minimization calculated by the closed-form quaternion method does not need to be adapted.

Sparsely sampled triangle meshes can be registered with the presented texture matching when the color texture images are available.

Finally, several digitized objects are presented and demonstrate the successful operation of the object digitizing system. It takes only some hours to digitize an entire object which can be considered as quick compared to other commercially available systems.

Chapter 6
Object inspection

In object inspection, the sensed data of a scene has to be matched with an object model. The scene may be composed of several objects. In the case of an inspection task the complete scene or a part of its geometry has to be verified. A different application where the scene is reconstructed is also presented. In both applications, the operator chooses the correct model from a database, enters a rough pose estimate and launches the automatic registration.

The applications presented in this chapter use the same extensions of the geometric point matching as the object digitizing application presented in the previous chapter. Again, the two surfaces to be matched have only some partial areas in common. Several features are combined in the closest point search and the same user interface allows an operator to roughly register two surfaces.

The presented work has been published in [SCHj] [SULa].

6.1 Introduction

The object inspection task is similar to the object recognition task in the sense that sensed data of a real world object is matched with a model. But, the main interest in object inspection is to visualize the measured data together with a virtual model and the results of the inspection process. Augmented reality covers the domain of such applications and is introduced in the following section.

6.1.1 Augmented reality

The major problem in the field of augmented reality (AR) consists of the registration of virtual and real objects. The object registration problem is elementary to all AR applications since the information which augments the real world information has to be in correspondence with it. For example, an aim has to follow the real target as it moves.

Definition of AR

In general, AR is defined by the following characteristics: a combination of real and virtual data which is registered and updated if necessary in real time [AZU]. Real data is acquired with a sensor as, for example, a video camera whereas virtual data stands for the additional information provided by a database or other knowledge.

In the case of video images the virtual data may be a wireframe object representation which enhances the real object boundaries or instruction text which assists the operator during a task. The virtual data used in AR allows the operator's perception and understanding of a scene in applications as medical visualization, guidance, assembly, quality insurance and inspection to be augmented.

Data augmentation

Several techniques to combine the virtual and the real data in one representation exist. Although, AR can also be applied to audio or other information, the following discussion considers only visual data. The available devices which combine real and virtual images or views can be classified into two broad categories: optical or video technology.

The optical approach typically uses see-through displays that place an optical combiner in front of the user's eyes. This provides a relatively simple and effective way to add visual information from the virtual world to the human view [HOF] [PEU]. However, since the virtual world is generated in general with a simple camera model and projected by a monitor system the combination with the real world considering its different lighting conditions is problematic. The virtual object in the resulting representation may be of low contrast and is not necessarily in focus with its corresponding real world object [AZU].

The video approach provides an escape to this problem: Digital hardware is used to combine the real world image scanned by a video camera and the virtual image rendered on a computer. Video or even stereo displays present the resulting image to the user [MAM] [MIL].

Another advantage of the video approach is the possibility to verify the registration since both the real and virtual representation are available to the computer. In optical systems, the virtual information is projected onto the real world by estimating the user's view point with head position tracking devices. If there is no sensing of the real world, the virtual world generation runs in open loop mode and there is no means by which the registration result can be verified. Video technologies however allow the comparison of the real and virtual image by image processing methods: A closed loop control of the object registration provides more accuracy.

Once the real world image is available, not only the registration verification but also the calculation of the registration itself can be performed with image processing techniques. Different types of image features to locate objects and registration methods ranging from manual to fully automatic have been proposed.

AR implementation in this work

The following implementation of AR follows the video approach for the above reasons. Real-time system answer is of minor concern in the aimed applications since the scene is static. However, an accurate matching is necessary to get a correct representation of the real world or a precise quality measure.

A 3D scanner measures the scene geometry and the geometric point matching algorithm performs the accurate model registration.

6.1.2 Applications

Microsystems inspection

In the first application, a micropart inspection system has been developed with contributions of the Institute of Microengineering Production of the Swiss Federal Institute of Technology in Lausanne.

Microsystems are modules with dimensions in the micrometer domain which incorporate various functions of electronic, mechanical, optical or chemical nature. The measuring of these small structures ensures the quality of the process and is the key for successful development of manufacturing technologies.

So far, the inspection of microstructures has been done by looking through a microscope. New approaches use the intuitiveness and 3D visualization of virtual reality environments coupled with a 3D vision system to perform measurements and verification of real microstructures at a high 3D accuracy [SULb]. The direct feedback between a real microscope and the virtual reality world is established by 3D vision and enables a realistic visualization of the measured micro structure data.

The measuring block uses a computerized 2D optical microscope combined with a 3D vision system to inspect the scene and to acquire range images of the specimen [SULb]. This vision system extracts the position and the 3D shape of the microparts.

The range images are triangulated as explained in Section 2.5 and then displayed on an interactive user interface similar to the one in Section 5.4. The operator selects the correct model and roughly aligns it with the sensed data using a space mouse. The geometric point matching snaps the model onto its real world counterpart. The real data is augmented by superimposing the virtual object representation and by coloring the measured points according to their fitting error [SULa].

Virtual world construction

In a second application, the construction of virtual worlds having a sensed scene view and the object models as input is demonstrated. The construction of virtual worlds has several application fields. Virtual worlds are used for instance in architecture to represent building geometry or in virtual reality robotics to program assembly tasks. Once the virtual representation of a real world is constructed the operator's view is not limited any more to the camera observing the real world. The virtual world provides a copy of the real world where the user has free inspection and interaction possibilities. There are other advantages as for example in a virtual robotics environment where the real robot is only used for the final task execution which saves energy and robot resources (see Section 4.7).

Virtual worlds have been constructed automatically in the object recognition application. The object detection problem has been solved

with a range change segmentation algorithm (see Section 4.4.1). This is not always possible and therefore a visually aided vision system is proposed in this chapter. AR provides a comfortable way to construct virtual worlds in an interactive manner and to see the progression of the construction.

The vision system disposes of an object database containing objects represented by their geometry which may be of any shape. These models are generated by a digitizing system as presented in Chapter 5 or provided by a CAD model database. The operator constructs the virtual world object by object with a point and click interface. He detects the different objects present in the scene and roughly registers the model. The virtual world is registered precisely to the real world using geometric point matching.

6.2 Comparison with other work

6.2.1 Object inspection

Usually the input of an inspection system is a cloud of points representing the sensed data. In order to inspect these data the exact coordinate transformation between the 3D scanner and the object has to be known. Traditionally, this needs a lengthy registration and complicated fixture of the sample to be inspected. Therefore, several authors have proposed automatic registration methods for sensed data of free-form objects and CAD models [MEN] [BESb]. Since these methods work directly on the sensed data points, no segmentation is needed and any object shape can be processed. Furthermore, the matching results can be visualized easily by attributing a color corresponding to the matching error to every data point [MOR].

This work develops an AR interface for the inspection task and extends the ICP algorithm to use surface orientation and color features. Triangulated CAD models can be used without conversion in the vision algorithm. Different methods are proposed to accelerate the registration since the sensed shape usually contains a huge amount of data points.

6.2.2 Virtual world construction

During registration, the object which has to be registered with some virtual data has first to be detected in the real world. Depending on the application several techniques to perform this task exist. Visual marks put on the object are easily located applying traditional segmentation techniques to the video image [HOF]. More flexible systems which allow several objects to be present in the scene search for geometric features in the video image. Features such as object edges are detected and grouped into one object representation.

As soon as we deal with objects of more complex shape the definition of pertinent and reliably detectable features becomes more difficult (see Chapter 2). Object detection is even more difficult if the objects are placed on a complex background formed by a pile of objects or textured material.

To avoid this problem, interactive systems for virtual world construction have been presented [MAM] [MIL]. Here, the operator locates and identifies the object and its type, this results in high flexibility since any object type can be treated. This approach is applied in this work since the use of human perception has the following advantages: a high score in estimating the kind and the rough pose of an object. A high quality interface is crucial for successful and ergonomic operator task completion. Stereo vision systems as used in this work provide the necessary depth perception and allow the operator to manipulate and locate objects successfully in 3D space.

Different levels of real world knowledge [MIL] ranging from simple to complex and from fully modeled to completely unknown lead to different reconstruction methods. With some methods simple unknown polyhedral worlds are constructed interactively by drawing lines along the object edges and by relying on grouping the complete object is formed. Other methods focus on the construction of block worlds where cubes are aligned manually [MAM]. Methods to construct virtual worlds built of more complex objects rely on a model database. Here, the object types present in the real scene are known from a model database, as for

example in assembly or medical tasks where the model information is accessible from a CAD database or computer tomograph scans.

Accurate matching of the virtual and real world is crucial for a successful virtual world construction. The pose alignment of 3D objects by matching image features fails in the presence of edgeless, smooth objects. In order to cope with any object form, again the ICP algorithm is used to register the model with the sensed data starting from a rough pose estimate entered by the operator.

6.3 Implementation

Since this chapter uses the same modifications of the ICP algorithm as presented in Chapter 5, they just get summarized here. Also, the same interactive user interface shown in Section 5.4 is used to present the results to an operator.

A method to handle large amounts of data in the registration process and the implementation of AR is presented later on.

ICP modifications

The ICP algorithm automatically aligns the selected model with the sensed data. Again, the use of several features in the closest point search as introduced in Section 3.4.2 is of great help. Especially in the virtual world construction task, different objects may touch one another and the geometric distance is not pertinent enough to distinct them. Surface color and orientation information provides an escape to this problem as demonstrated by the following results.

The original ICP algorithm holds for the matching of two surfaces where one surface is a subset of the other. Here, the scene is not segmented any more and is usually composed of several objects. The sensed and the virtual data contain surface areas not present in the other one. Therefore, again the weighing of Section 5.6.1 is used to exclude from the matching error minimization the points of parts which do not overlap.

Data reduction

Usually the acquired scene or object data represents a huge amount of data, typically in the order of magnitude of 100’000 points. The following methods are introduced to reduce the amount of the data point representing the surface P for which the closest point search is performed.

Since the operator roughly registers the model with the scene, the data points used in the geometric point matching are limited to the points falling into the model circumsphere.

Furthermore, the matching can be started with a low resolution representation of the sensed data only including every fourth point for example. Once a first matching is done, a scene at higher resolution can be loaded and the matching is continued.

Error indication with AR

Since the automatic matching couples every data point with the closest point on the model, the resulting coupling distances give a measure of the objects' similarity. Statistical indicators as mean and deviation can be calculated and displayed to the operator. Especially in the inspection task, it is of great interest to easily identify the object areas which are outside the error tolerance. The presented system colors the data points according to their coupling distance and provides a quick visual overview of the object quality.

6.4 Results

6.4.1 Microsystems inspection

Acquisition

Inspection of microsystems using an AR interface is interesting since it deals with object dimensions which are not accessible to the human eye. As stated in the introduction, an optical microscope is used to measure the small microparts. Several scans performed at different microscope

table height positions allow the calculation of a range image using depth from focus algorithms (see Section 2.4.1) [SULb].

The following Figure 6.1 shows a range image acquired with the mentioned 3D scanner. The range image contains about 300'000 pixels from a micromotor of 132 microns in height and 280 microns in diameter. The micromotor contains the stator and the rotor part.

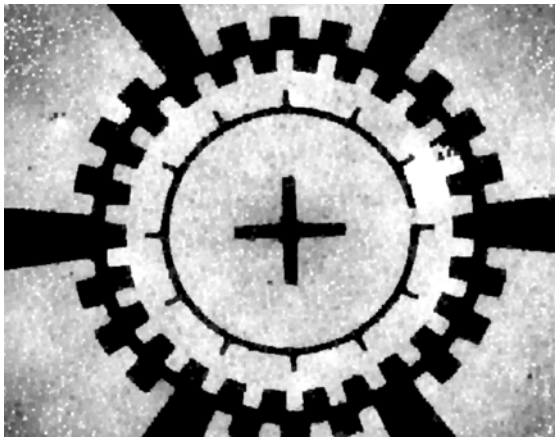


Figure 6.1 Range image of a micromotor

The acquired range image is converted in a triangulated view as described in Section 2.5 and is rendered on a graphics monitor. The operator selects the corresponding model from the database and uses the space mouse to register it roughly with the data measurements. Figure 6.2 shows the models of the rotor and stator building the micromotor of Figure 6.1.

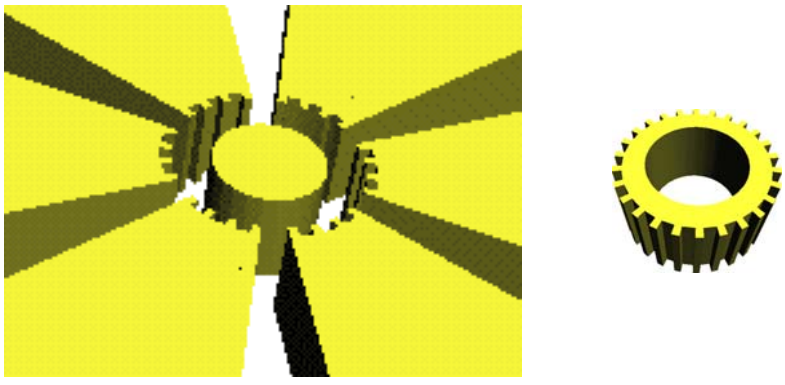


Figure 6.2 Micropart models

Inspection

The automatic registration is executed using the entered pose estimate. Figure 6.3 shows a sequence of screen shots of several iterations of the ICP algorithm matching the stator part.

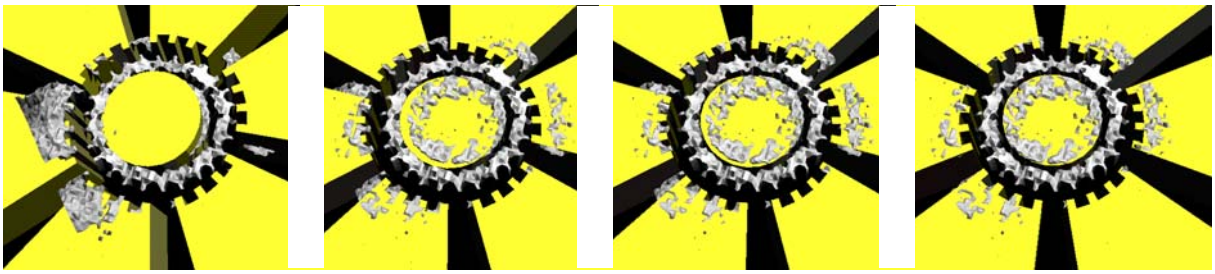


Figure 6.3 Four iterations of the automatic matching of the stator

The matched surface is coded by coloring the data points according to the matching error. The color of the points which are inside the matching tolerance ranges from red to blue, where blue represents a perfect match. Points which are outside the tolerance are rendered in grayscale according to their height. Figure 6.4 shows the matched stator model superposed to the object data and the data colored according to the matching error.

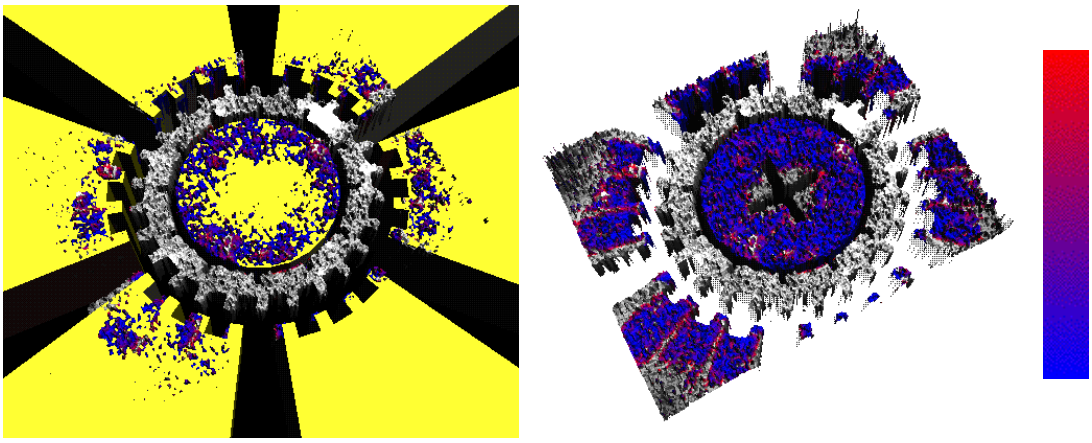


Figure 6.4 Augmented reality showing the matching errors for a micromotor

The augmented reality allows the quick identification of the stator parts which are out of tolerance (the center and the four corners in the above example). As expected the data points corresponding to the rotor are not

coupled at all. The colored regions represent matchings with a maximal error of 2.4 µm. The scanner works at a resolution of 0.5 µm. Compared to this the resulting mean fitting error of 1.32 µm is quite high and the data considerably noisy.

Problems with CAD models

In the above example the stator has been modeled by a standard modeler. Such models may cause problems when using the kD tree to find the nearest point on a triangle as defined in Section 3.4.5. This accelerated search algorithm assumes that the nearest triangle is attached to the nearest vertex.

Since standard modelers usually compose a complex shape by adding or subtracting simple triangulated shapes such as cubes or spheres, the resulting surface is not necessarily correctly triangulated. Vertices may lie on triangle borders and the nearest triangle is therefore not necessarily connected to the nearest vertex.

The triangulation of such models needs to be solved before they can be used with the kD tree search.

6.4.2 Virtual world construction

In the second application, an AR interface helps to reconstruct a scene having only one view and the object models of it. The different steps are similar as for the object inspection task.

An operator uses the 3D interface to roughly place the chosen model. Then, the ICP algorithm performs the precise alignment. Figure 6.5 illustrates these three steps. Augmented reality helps to verify quickly if the geometric point matching has converged successfully and if some more models need to be added to the scene.

Figure 6.5 Scene view with missing data, roughly placed model and automatic matched model

The ICP algorithm converges successfully only for a limited range of initial configurations (see Section 3.7). However, experiments in Section 5.8 show that this range appears to be comfortably large and does not demand a precise alignment from the operator.

Once all models are placed correctly in the scene, the sensed data can be removed and the constructed virtual world can now be inspected from any view point as shown in Figure 6.6.

Figure 6.6 Different views of the constructed virtual world

The above example shows that a virtual representation of a real world composed of modeled objects can be constructed successfully with the presented system. Only one view is needed and therefore one static 3D scanner observing the scene is sufficient.

For teleoperation tasks, it is important to be able to change the view point, which is possible if the virtual world is completely constructed. Manipulation and path planning tasks are much more easily defined if the operator is not limited to one view point.

6.5 Conclusions

The presented system provides a method with which to successfully implement augmented reality for object inspection and virtual world construction.

The scene is sensed by a 3D scanner which allows the triangulation of the data points and the rendering of the acquired data in any pose.

A semi-automatic registration approach is proposed in order to register the real data and the virtual models. The operator locates and identifies objects and the computer calculates the registration. Human perception gives flexibility while the computer provides precision.

The ICP algorithm allows an efficient and fast data registration of free-form shapes. The matching error of the closest point search provides a measure for the surface similarities.

The augmented reality interface presents the results by visualizing the errors directly on the sensed data. The data points are colored according to their individual error distance. This permits for example the easy identification of the bad object regions.

Chapter 7
Conclusions

This work exploits the advantages of low-level primitives for the processing of free-form objects in 3D vision. Geometric point matching has been investigated as an escape from the classical image segmentation and feature extraction problem and to register precisely free-form surfaces. Different applications show the usefulness and demonstrate the successful implementation of the proposed approach.

7.1 Conclusions

The processing of 3D objects is greatly simplified by the use of 3D scanners. The optical working principle of the used 3D scanners allows the quick acquisition of the surface geometry of objects of any shape. Since 3D scanners directly sense the object geometry the implemented vision algorithms do not have to deal with geometry recovering, nor with scaling. The fusion of color and depth is straightforward as the same camera senses both pieces of information in the employed 3D scanner.

Low-level surface representations such as a cloud of points or a triangle mesh are directly obtained from the 3D scanner output data. This representations are well-suited for the processing of free-form objects and have been used successfully to model several free-form objects.

Geometric point matching is proposed to register precisely free-form surfaces. For this task an iterative closest point (ICP) algorithm is implemented and extended with the following contributions.

The integration of additional features such as color, texture and surface orientation improves the matching where the surface geometry alone fails.

Several methods are proposed to reduce and to organize the primitives representing an object surface. The kD tree provides a method to establish quickly correspondence between points or triangles. Furthermore, the kD tree is well-suited to include any number of features in the correspondence establishment. Homogenous data reduction is obtained by observing the range image structure.

Further extensions adapt ICP algorithm for the use in different applications. Weighing functions provide a method to match surfaces which only partially overlap. Discriminating matching quality measures are used to classify the matching of two surfaces. Change error allows the reduction to a minimum of the needed iterations of the ICP algorithm.

The convergence behavior of the extended ICP algorithm is extensively investigated with the proposed setup leading to the SIC-maps.

Finally, the presented applications and the obtained results show the successful integration and extension of the ICP algorithm for the encountered task.

7.2 Contributions

In addition to the above general conclusions, this section highlights the main contributions of this work.

SIC-maps

This work proposes a new way to measure the qualify of a matching. It defines a setup which reduces significantly the space of all possible poses of two surfaces to be matched. The configurations for which the ICP algorithm finds a successful match are indicated in a SIC-map.

This SIC-maps allow the investigation of the convergence behavior of the ICP algorithm and have been established for several free-form objects. The information of the SIC-maps has been successfully used to design the object recognition application.

Enhanced correspondence establishment

The integration of the surface color and surface orientation feature in the closest point search results in significant improvement of the matching behavior of the ICP algorithm. All features use an Euclidean distance measure and are therefore easily added to a total coupling distance and integrated in the fast kD tree search. An intuitive method is proposed to normalize the different feature value ranges in combination with the overlap detection weighing function.

A novel texture matching algorithm allows the registration of sparsely sampled surfaces using their texture image.

Implemented applications

A major contribution of this work covers the implemented and operational applications. They demonstrate the successful integration of the several extensions proposed for the ICP algorithm. The selected applications provide solutions to problems of major interest in multimedia, industrial inspection, teleoperation or reverse engineering.

Exhaustive tests have been performed to assess the recognition performance of the object recognition application. The successful assembly of a scotch dispenser part validates the performance of the vision system.

Several objects have been digitized to show the capabilities of the object digitizing system to obtain a virtual model which corresponds to the real world counterpart in geometry and color.

The object inspection system and its augmented reality interface allow the easy inspection of details on microparts invisible to the human eye. The visual error representation permits the fast location of shape differences.

7.3 Limitations and future work

Object recognition

Although several methods have been proposed to accelerate the ICP algorithm the response time of the object recognition is still quite high

(about 10 seconds on a fast workstation). Geometric point matching should be chosen for its capabilities to accurately calculate an object pose.

When the recognition includes a large number of models, indexing techniques must be introduced. Appearance based methods such as those presented in Section 2.2.1 could be useful to select a few potential models used for the relatively costly geometric point matching and they even provide a rough pose estimate.

Object digitizing and inspection

The object digitizing application constructs models by adding one view after the other sequentially to the growing model. Therefore, the registration errors get accumulated. Different authors have proposed methods to escape this problem [BEN] [BERb] [STOb] [TUR]. A spring model which minimizes the total registration error of all surfaces is presented in [STOb]. Turk and Levoy use a cylindrical scan of the object as a master onto which every additional surface is registered [TUR].

More experiments are necessary to investigate the precision of the registration performed by the ICP algorithm. In general, the correct rigid transformation which matches two surfaces is not known and it is difficult to evaluate the ICP matching results. In a first experiment not presented yet, two surfaces have been aligned with photogrammetric methods and compared to the matching obtained with the ICP algorithm. The difference between the two matchings was very small and shows that the ICP algorithm is capable of reaching more or less the same performance than established methods.

This work focused on the digitizing of free-form objects. Usually the objects used in industry are simpler but have sharper edges. The scanning and reconstruction of such surfaces leads to new problems not yet investigated.

The recursive closest point search based on kD trees is efficient for long feature vectors since the memory requirement of the kD tree is independent on the number of features used. However, other methods as distance maps which give direct access to the closest point may show better performance for small feature vectors, for example the vector of the Cartesian coordinates.

A remaining challenge is the automatic development of poses for the establishment of a rough surface registration. Some first experiments with the color feature have been performed in this work. Further investigations are necessary in order to completely automate the digitizing and inspection systems. Curvatures may provide the necessary information to construct a set of potential pose estimates.

Acknowledgements

Several people have contributed in different manners to the successful completion of this work. Many ideas were born during discussions with my colleagues. It was important for me to have people around me who showed interest in my work, asked questions and proposed solutions.

I am especially grateful to my supervisor Heinz Hügli who supported me with advice and encouragement. He gave me the personal freedom and infrastructure in which to develop my ideas and made it possible for me to enlarge my knowledge at several scientific conferences.

I would like to thank Horst Bunke, Jacques Jacot and Hans-Heinrich Nägeli for their interest in my work, the fruitful discussions and for accepting to co-examine this thesis.

The good ambience at our research group made me feel comfortable at work. Olivier Hüsser and Timothée Jost were always open to discuss a problem or to share a joke. Olivier encouraged me to do sports which helped me to compensate the mental effort of work. Timothée contributed a lot to the object digitizing and inspection applications. He developed the main parts of the user interface, the mesh fusion algorithm and digitized most of the objects presented.

Some of the former members of our group did some of the first work in the domain of my research. Laurent Claud implemented the first version of the ICP algorithm. Philippe Gingins installed the 3D scanner at our laboratory and introduced me into the domain of 3D vision. François Tièche and Claudio Facchinetti helped me to acquire some knowledge in 2D vision.

Several trainees visited our laboratory and were involved in my work. Jean-Christian Houg developed and programmed several routines for the object recognition application. Furthermore, he implemented a first version of the object digitizing tool. Dimitrios Semitekos made the first implementation and experiences with the kD tree.

Of all the collaborators at IMT who shared some time with me at work or in my spare time I would like to thank especially Heinz Burri who maintained our UNIX workstations and Vincent Moser who provided me with the styles for the layout of this work.

Some parts of this work have been done in collaboration with external research groups. I would like to thank Emerico Natonek and Armin Sulzmann from the Swiss Federal Institute of Technology in Lausanne for their contributions to the robot teleoperation and the micropart inspection applications. Marjan Trobina from the Swiss Federal Institute of Technology in Zürich who provided the software for our 3D scanner.

Part of this research has been funded by the Swiss national Priority Program in Informatics, Knowledge-based systems, under project number 5003-344336 and by the Swiss National Science Foundation under project number 2100-43530.

Finally, special thanks go to Alison Chisholm who checked my English writing. Also, I would like to express my gratitude to the people from the Stadtmission Neuchâtel for their spiritual support, to my parents and my sister for their education and love, to Roland and Margrit who made life comfortable and to Irena for having pushed me to finish this work for a very special reason.

References

[AMA] H.-P. Amann, “3D Object Recognition Based on Surface Representations,” PHD thesis, University of Neuchatel, 1990.

[ARU] K. S. Arun, T. S. Huang and S. D. Blostein, “Least-Squares Fitting of Two 3-D Point Sets,” Proc. of IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 9, no. 5, pp. 698-700, 1987.

[AZU] R. Azuma, “A survey of augmented reality,” Proc. of SIGGRAPH, ACM, Los Angeles, 1995.

[BAR] G. Barequet and M. Sharir, “Partial Surface and Volume Matching in Three Dimensions,” Proc. of IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 19, no. 9, pp. 929-948, 1997.

[BEN] R. Benjemaa and F. Schmitt, “Fast Global Registration of 3D Sampled Surfaces Using a Multi-Z-Buffer Technique,” Proceedings of the International Conference on Recent Advances in 3D Imaging and Modeling, Ottawa, 1997.

[BERa] R. Bergevin, D. Laurendeau and D. Poussart, “Estimating the 3D Rigid Transformation Between Two Range Views of a Complex Object,” IEEE Proceedings of Int. Conf. on Pattern Recognition, pp. 478-482, 1992.

[BERb] R. Bergevin, M. Soucy, H. Gagnon and D. Laurendeau, “Towards a General Multi-View Registration Technique,” Proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 18, no. 5, pp. 540-547, 1996.

[BESa] P.J. Besl, “Triangles as a Primary Representation,” Lecture Notes in Computer Science, Springer, vol. 994, pp. 191-206, 1995.

[BESb] P.J. Besl and N.D. McKay, “A Method for Registration of 3-D Shapes,” Proc. of IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pp. 239-256, 1992.

[BESc] P.J. Besl, “The Free-Form Surface Matching Problem,” Machine Vision for Three-Dimensional Scenes, Academic Press, pp. 25-71, 1990.

[BESd] P.J. Besl, “Geometric Modeling and Computer Vision,” Proc. of the IEEE, vol. 76, no. 8, pp. 936-958, 1988.

[BLA] G. Blais and M. Levine, “Registering multiview range data to create 3D computer objects,” Proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 17, no. 8, pp. 820-824, 1995.

[CHA] G. Champleboux, S. Lavallée, R. Szeliski, L. Brunie, “From accurate imaging sensor calibration to accurate model-based 3-D object localization,” Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 83-89, 1992.

[CHE] Y. Chen and G. Medioni, “Object modelling by registration of multiple range images,” Int. J. of Image and Vision Computing, vol. 10, no. 3, pp. 145-155, 1992.

[CHS] J.-L. Chen and G.C. Stockman, “Determining pose of 3D objects with curved surfaces,” Proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 18, no. 1, pp. 52-57, 1996.

[CHU] C.S. Chua, R. Jarvis, “3D Free-Form Surface Registration and Object Recognition,” International Journal of Computer Vision, Kluwer Academic Publishers, vol. 17, pp. 77-99, 1996.

[DAN] P.-E. Danielsson, “Euclidian Distance Mapping,” Computer Graphics and Image Processing, vol. 14, pp. 227-248, 1980.

[DOR] C. Dorai and A.K. Jain, “Shape spectra based view grouping for free-form objects,” Proc. of the IEEE Int. Conf. on Image Processing, vol. 3, pp. 340-343, 1995.

[FAU] O.R. Faugeras and M. Hebert, “The Representation, Recognition, and Locating of 3-D Objects,” The International Journal of Robotics Research, vol. 5, no. 3, pp. 27-52, 1986.

[FELa] J. Feldmar, N. Ayache, F. Betting, “3D-2D projective registration of free-form curves and surfaces,” research report, no. 2434, Sophia Antipolis, INRIA, 1994.

[FELb] J. Feldmar and N. Ayache, “Rigid and Affine Registration of Smooth Surfaces using Differential Properties,” Lecture Notes in Computer Science, vol. 801, no. 2, pp. 397-406, 1994.

[FRE] W. Frei, M. Singh, T. Shibata, “Digital image change detection,” SPIE Optical Engineering, vol. 19, no. 3, pp. 331-338, 1980.

[FRI] J.H. Friedman, J. Bentley and R. Finkel, “An algorithm for finding best matches in logarithmic expected time,” ACM Trans. on Mathematical Software, vol. 3, no. 3, pp. 209-226, 1977.

[GAR] M.A. Garcia, “Fast Approximation of Range Images by Triangular Meshes Generated through Adaptive Randomized Sampling,” Proc. of IEEE Int. Conf. on Robotics and Automation, Nagoya, pp. 2043-2048, 1995.

[GIN] P. Gingins and H. Hugli, “Model-based 3D object recognition by a hybrid hypothesis generation and verification approach,” Intelligent Robots and Computer Vision XIII: Algorithms and Computer Vision, SPIE, Boston, vol. 2353, pp. 67-77, 1994.

[GODa] G. Godin and P. Boulanger, “Range image registration through viewpoint invariant computation of curvature,” From Pixels to Sequences, Proc. conf. ISPRS, Zurich, vol. 30, pp. 170-175, 1995.

[GODb] G. Godin, M. Rioux, and R. Baribeau, “3-D registration using range and intensity information,” Proceedings of SPIE Videometrics III, Boston, vol. 2350, pp. 279-290, 1994.

[HER] M. Hebert, J. Ponce, T.E. Boult, A. Gross, “Report on the 1995 Workshop on 3-D Object Representations in Computer Vision,” Lecture Notes in Computer Science, Springer, vol. 994, pp. 1-18, 1995.

[HILa] A. Hilton, A.J. Stoddart, J. Illingworth and T. Windeatt, “Building 3D Graphical Models of Complex Objects,” Eurographics Conf., vol. 1, pp. 193-204, 1996.

[HILb] A. Hilton, A.J. Stoddart, J. Illingworth and T. Windeatt, “Reliable Surface Reconstruction from Multiple Range Images,” European Conf. on Computer Vision, Springer, pp. 117-126, 1996.

[HOF] W. A. Hoff, K. Nguyen, T. Lyon, “Computer vision-based registration techniques for augmented reality,” Proc. of SPIE Intelligents Robots and Computer Vision, Boston, vol. 2904, pp. 538-548, 1996.

[HOR] B.K.P. Horn, “Closed-form solution of absolute orientation using unit quaternions,” Journal of the Optical Society of America, vol. 4, no. 4, pp. 629-642, 1987.

[HOU] J.-C. Houg, “Reconstruction de formes tridimensionnelles,” report IMT/HU, University of Neuchatel, no. 384, 1995.

[HUGa] H. Hügli, C. Schütz, C. Baur, E. Natonek, “Knowledge-based 3-D vision system,” Proceedings of Information Conference Module 2, Knowledge based systems, Swiss National Science Fundation, pp. 46-53, 1994.

[HUGb] H. Hügli, C. Schütz, D. Semitekos, “Geometric matching for free-form 3D object recognition,” ACCV, Singapore, vol. 3, pp. 819-823, 1995.

[HUGc] H. Hügli, C. Schütz, “How well performs free-form 3d object recognition from range images?,” Proceedings of SPIE Intelligents Robots and Computer Vision, Boston, vol. 2904, pp. 66-74, 1996.

[HUGd] H. Hugli, Ch. Schutz, “Computer vision of free-form 3D objects by geometric matching,” Proc. 3rd Japan-French Congress on Mechatronics, Besancon, vol. 1, pp. 156-160, 1996.

[HUGe] H. Hügli, C. Schütz, “Geometric Matching of 3D Objects: Assessing the Range of Successful Initial Configurations,” Int. Conf. on Recent Advances in 3-D Digital Imaging and Modeling, pp. 101-106, 1997.

[JIA] X. Jiang, H. Bunke, "Dreidimensionales Computersehen: Gewinnung und Analyse von Tiefenbildern," Springer Verlag, 1997.

[JOHa] A. Johnson and S.B. Kang, “Registration and Integration of Textured 3-D Data,” International Conference on Recent Advances in 3-D Digital Imaging and Modeling, Ontario, pp. 234-241, 1997.

[JOHb] A. Johnson and M. Hebert, “Surface registration by matching oriented points,” International Conference on Recent Advances in 3-D Digital Imaging and Modeling, Ontario, pp. 121-128, 1997.

[JOS] T. Jost, C. Schütz, H. Hügli, “Modeling 3D Textured Objects by Fusion of Multiple Views,” European Signal Processing Conference, to be published, 1998.

[KRE] B. Krebs, P. Sieverding and B. Korn, “Correct 3D Matching via a Fuzzy ICP Algorithm for Abitrary Shaped Objects,” DAGM96, Proc. of Mustererkennung, Heidelberg, pp. 521-528, 1996.

[LI] S.Z. Li, “Object recognition from range data prior to segmentation,” Image and Vision Computing, vol. 10, no. 8, pp. 566-576, 1992.

[MAM] D. Maman, P. Fuchs, F. Nashashibi and C. Laurgeau, “Utilisation des techniques de la réalité virutelle pour la modélisation interactive en téléoperation,” Proc. 5th Int. Conf. on Interface to real and virtual worlds, Montpellier, pp. 51-64, 1996.

[MAR] Fernando C. M. Martins and José M. F. Moura, “Video representation with 3D entities,” IEEE Journal on Selected Areas in Communications, 1997.

[MEN] C.-H. Menq, H.-T. Yau and G.-Y. Lai, “Automated Precision Measurement of Surface Profile in CAD-Directed Inspection,” IEEE Trans. on Robotics and Automation, vol. 8, no. 2, pp. 268-278, 1992.

[MIL] P. Milgram, A. Rastogi, J.J. Grodski, “Telerobotic Control Using Augmented Reality,” Robot and Human Communication (RO-MAN), Japan, 1995.

[MOR] V. Moron, P. Boulanger, T. Masuda, T. Redarce, “Automatic inspection of industrial parts using 3-D optical range sensor,” Proc. of SPIE Videometrics, Philadelphia, vol. 2598, pp. 315-325, 1995.

[NATa] E. Natonek, L. Flückiger, Th. Zimmerman, C. Baur, "Virtual Reality: an intuitive approach to Robotics," Proc. SPIE Int. Symposium on Photonics, Telemanipulator and Telepresence Technologies, vol 2351-32, Boston, 1994.

[NATb] E. Natonek, “3D-MBA: Système de reconnaissance d'objets par vision basé modèles 3-D appliqué aux mondes virtuels robotisés,” Thesis No. 1477, Dept. of Microtechnics, Swiss Federal Institute of Technology, Lausanne, Switzerland, 1996.

[PEU] B. Peuchot, A. Tanguy and M. Eude, “Virtual reality as an operative tool during scoliosis surgery,” Proc. 5th Int. Conf. on Interface to real and virtual worlds, Montpellier, pp. 149-157, 1996.

[PIT] R. Pito, “Mesh integration based on co-measurements,” IEEE International Conference on Image Processing, Lausanne, vol. 2, pp. 397-400, 1996.

[POT] M. Potmesil, “Generating Models of Solid Objects by Matching 3D Surface Segments,” Proc. Int. Joint Conf. on Artificial Intelligence, Karlsruhe, pp. 1089-1093, 1983.

[RUSa] J.C. Russ. The Image Processing Handbook, CRC Press, pp. 272-274, 1995.

[RUSb] D. Rusin, Frequently-asked questions about spheres, http://www.math.niu.edu/~rusin/known-math/index/spheres.html.

[RUT] M. Rutishauser, M. Stricker, M. Trobina, “Merging Range Images of Arbitrarily Shaped Objects,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, pp. 573-580, 1994.

[SCHa] C. Schütz, “Range finder systems - a tentative overview,” report IMT/HU, University of Neuchatel, no. 410, 1996.

[SCHb] C. Schütz, “Free-form object surface triangulation from a cloud of points,” report IMT/HU, University of Neuchatel, no. 419, 1997.

[SCHc] C. Schütz and H. Hügli, “Change detection in range imaging for 3D scene segmentation,” Proc. of SPIE Europto, Besancon, vol. 2786, pp. 98-106, 1996.

[SCHd] C. Schütz, H. Hügli, “Towards the recognition of 3D free-form objects,” Intelligent Robots and Computer Vision XIV, Algorithms, Techniques, Active Vision and Materials Handling, SPIE, Philadelphia, vol. 2588, pp. 476-484, 1995.

[SCHe] C. Schütz, T. Jost, H. Hügli, “Free-Form 3D Object Reconstruction from Range Images,” Proceedings of VSMM'97, Geneva, pp. 69-70, 1997.

[SCHf] C. Schütz, T. Jost and H. Hügli, “Semi-Automatic 3D Object Digitizing System Using Range Images,” Proc. of ACCV, Lecture Notes in Computer Science, Springer, vol. 1351, pp. 490-497, 1998.

[SCHg] C. Schütz, N. Natonek, Ch. Baur, H. Hügli, “3D Vision in a Virtual Reality Robotics Environment,” Proc. of SPIE Telemanipulator and Telepresence Technologies, Boston, vol. 2901, pp. 138-146, 1996.

[SCHh] C. Schütz and H. Hügli, “Recognition of 3-D objects with a closest point matching algorithm,” From Pixels to Sequences, Proc. conf. ISPRS, Zurich, vol. 30, pp. 128-133, 1995.

[SCHi] C. Schütz, H. Hugli, “Free-form 3D object recognition,” Optical 3-D Measurement Techniques, Wichmann, Heidelberg, vol. 3, pp. 516-525, 1995.

[SCHj] C. Schütz, H. Hügli, “Augmented reality using range images,” Proceedings of SPIE Stereoscopic Displays and Virtual Reality Systems, San Jose, vol. 3012, pp. 472-478, 1997.

[SCHk] C. Schütz, T. Jost, H. Hügli, “Multi-Feature Matching Algorithm for Free-Form 3D Surface Registration,” International Conference on Pattern Recognition, Brisbane, to be published, 1998.

[SIM] D.A. Simon, M. Hebert and T. Kanade, “Techniques for Fast and Accurate Intra-Surgical Registration,” Journal of Image Guided Surgery, vol. 1, no. 1, pp. 17-29, 1995.

[SOUa] M. Soucy, G. Godin, R. Baribeau, F. Blais and M. Rioux, “Sensors and Algorithms for the Construction of Digital 3-D Colour Models of Real Objects,” Proceedings of IEEE International Conference on Image Processing, vol. 2, pp. 409-412, 1996.

[SOUb] M. Soucy, G. Godin, M. Rioux, “A Texture-Mapping Approach for the Compression of Colored 3D Triangulations,” The Visual Computer, vol. 12, no. 10, pp. 503-514, 1996.

[SOUc] M. Soucy and D. Laurendeau, “A general surface approach to the integration of a set of range views,” Proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 17, no. 4, pp. 344-358, 1995.

[STE] F. Stein and G. Medioni, “Structural Indexing: Efficient Three Dimensional Object Recognition,” Three-Dimensional Object Recognition System, Elsevier Science Publishers, pp. 353-373, 1993.

[STOa] A. J. Stoddart, S. Lemke, A. Hilton, T. Renn, “Estimating pose uncertainty for surface registration,” British Machine Vision Conference, Edinburgh, pp. 23-32, 1996.

[STOb] A. J. Stoddart, A. Hilton, “Registration of multiple point sets,” Int. Conf. on Pattern Recognition (ICPR), Vienna, pp. 40-44, 1996.

[SULa] A. Sulzmann, C. Schütz, H. Hügli and J. Jacot, “Augmented Reality as an Interactive Tool for Microscopic Imaging, Measurement and Model Based Verification of Simulated Parts,” First Int. Conf. on Modeling and Simulation of Microsystems, Semiconductors, Sensors and Actuators (MSM), Santa Clara, 1998.

[SULb] A. Sulzmann, “Hochpräzise 3D Bildverarbeitung zur visuellen Relativpositionierung von Robotersystemen in der Mikromontage,” Thesis, Dept. of Microtechnics, Swiss Federal Institute of Technology, Lausanne, 1997.

[TUR] G. Turk and M. Levoy, “Zippered polygon meshes from range images,” Proceedings ACM Siggraph, Orlando, pp. 311-318, 1994.

[ZHA] Z. Zhang, “Iterative point matching for registration of free-form curves and surfaces,” Int. Journal of Computer Vision, vol. 13, no. 2, pp. 119-152, 1994.