
Article:
Pears, Nick orcid.org/0000-0001-9513-5634, Heseltine, Tom and Romero, Marcelo (2010) From 3D Point Clouds to Pose-Normalised Depth Maps. International Journal of Computer Vision. pp. 152-176. ISSN 0920-5691

https://doi.org/10.1007/s11263-009-0297-y

---

ijcv manuscript No.
(will be inserted by the editor)

From 3D point clouds to pose-normalised depth maps

Nick Pears, Tom Heseltine and Marcelo Romero

Received: date / Accepted: date

Abstract We consider the problem of generating either pairwise-aligned or pose-normalised depth maps from noisy 3D point clouds in a relatively unrestricted poses. Our system is deployed in a 3D face alignment application and consists of the following four stages (i) data filtering (ii) nose tip identification and sub-vertex localisation (iii) computation of the (relative) face orientation; (iv) generation of either a pose aligned or a pose normalised depth map. We generate an implicit radial basis function (RBF) model of the facial surface and this is employed within all four stages of the process. For example, in stage (ii), construction of novel invariant features is based on sampling this RBF over a set of concentric spheres to give a spherically-sampled RBF (SSR) shape histogram. In stage (iii), a second novel descriptor, called an isoradius contour curvature signal, is defined, which allows rotational alignment to be determined using a simple process of 1D correlation. We test our system on both the University of York (UoY) 3D face dataset and the Face Recognition Grand Challenge (FRGC) 3D data. For the more challenging UoY data, our SSR descriptors significantly outperform three variants of spin images, successfully identifying nose vertices at a rate of 99.6%. Nose localisation performance on the higher quality FRGC data, which has only small pose variations, is 99.9%. Our best system successfully normalises the pose of 3D faces at rates of 99.1% (UoY data) and 99.6% (FRGC data).

Nick Pears, Marcelo Romero
Department of Computer Science
University of York, UK
E-mail: nep,mromero@cs.york.ac.uk

Tom Heseltine
Aurora Computer Services Ltd
Hannington, UK
E-mail: t.heseltine@auroracs.co.uk

Keywords 3D feature extraction · invariance · 3D landmark localisation · 3D pose normalisation

1 Introduction

This paper focuses on the problems associated with generating a pair of aligned depth maps for the purpose of matching 3D shapes. The input to our system consists of noisy 3D point clouds of arbitrary resolution and in relatively unrestricted poses. We also consider the closely-related problem of generating a pose-normalised depth map, where the depth map is put into some canonical pose, such as the frontal pose (front view mug shot pose) often used in both 2D and 3D face recognition applications. Such depth maps are useful when applying a variety of classification techniques to 3D retrieval tasks, which includes methods based on linear sub-spaces, such as principal components analysis (PCA) and linear discriminant analysis (LDA), and other methods such as support vector machines (SVM), boosting methods, and so on. Our method may be applied to any 3D retrieval task, where there is at least one distinctive 3D feature on the visible surface, but here we discuss our methods in the context of 3D face recognition, with the nose tip selected as the distinctive point, as this is the application in which we have deployed and evaluated our system.

Recently, there has been a lot of research interest in both 3D face processing [7], [38], [42], [62], [36], [9], [57], [31] and 2D/3D face processing [60], [13], [8], [43]. Many researchers have cited the perceived benefits of using 3D data for face recognition instead of, or in addition to 2D data; namely an improved robustness to pose and lighting variations and potentially more reliable mechanisms for dealing with expression changes. Such bene-

fits were perhaps overstated five years ago, in the initial phase of 3D face recognition activity, when *invariance* to pose and lighting conditions was sometimes claimed. However, even current active sensors that project their own known light source onto the scene cannot yet generate scans that are completely immune to the ambient lighting conditions, such as the level of sunlight streaming through a window. Furthermore, when head pose changes, a 3D sensor can not produce data that can be modelled as a simple rigid Euclidean transformation of the data generated from the original pose. The main reason is self occlusion when, for different head poses, different parts of the face are visible. However, there are other reasons, such as the angle of incidence of the projected light on the facial surface changing and different parts of the the face moving into more or less favourable ambient viewing conditions as the head pose changes. Despite such problems, which are partly due to shortcomings in 3D sensor technology, 3D does offer the possibility of facial recognition in more unconstrained viewing conditions than is currently available in 2D approaches. Such '3D at a distance' recognition technology is suitable for applications where highly prescribed subject cooperation is impossible or undesirable.

Much of the 3D face work presented in the literature uses low noise 3D data in a frontal pose and normalisation techniques sometimes even require that both eyes are visible, which is at odds with a main selling point of 3D approaches, namely robustness to pose variations. In contrast, our method requires us to be able to identify a single distinctive point within the 3D scan, which is less restrictive than needing to view several features simultaneously and, in addition, it manages significant areas of missing data, such as occurs from self-occlusion, in robust and natural way. This refers to the nose occluding part of the cheek or the upper lip, when the facial pose is allowed to vary up to 45 degrees relative to frontal, but does not imply reconstruction of missing data in extreme poses, such as a pure profile, which are not used in our experimentation.

Appearance based methods have proved competitive in terms of achieving state-of-the-art performance in 2D face recognition. It is possible to adapt these methods, such as fisherface [4], to work with 3D data [29]. The results have been promising, because of the excellent background segmentation and explicit, discriminating 3D data. A requirement for such methods to work well is that all the data has a common alignment, which is usually a frontal view. We have developed a process for robust frontal 3D face alignment, when that 3D face data is potentially noisy and has missing parts due to spectacles, beards and self-occlusion. The four steps of this process are: (i) Filter the data automatically; (ii) Identify the nose tip vertex and interpolate the nose tip location to sub-vertex resolution; (iii) Compute the (relative) face orientation; (iv) Generate a pose aligned or pose normalised depth map.

There are two main themes that run through this process: (i) *The use of a radial basis function (RBF) model of the facial surface.* This is employed in all four stages above. The RBF describes the signed 'distance to surface' (DTS) of any point in 3D space. In terms of nose tip localisation, for example, the RBF provides a natural mechanism to generate pose-invariant 3D shape descriptors, that have high immunity to missing parts, without having to explicitly reconstruct those missing parts. In terms of the final stage, which generates an arbitrary resolution depth map, interpolating where the RBF is zero allows us to find facial surface points to any desired resolution. (ii) *The use of spherically defined methods and features for pose invariance.* This occurs in three layers: firstly the RBF itself is spherical in nature, in that each component has a fixed value over a sphere in 3D space. Secondly this RBF is sampled over a set of concentric spheres, to give novel pose invariant features called 'spherically-sampled RBF' (SSR) shape histograms. These have been very successful in identifying the facial nose tip. Thirdly, concentric spheres centred on the nose tip generate 3D space-curves, called 'isoradius contours' by intersecting with the implicit facial surface (where the RBF is zero). This provides an effective method for either the alignment of a pair of faces, or the normalisation of facial pose to a canonical pose.

In the following section, we overview previous work in 3D object retrieval and review related work in the key areas that this paper addresses. The next two sections describe our two new 3D invariant feature types, SSR descriptors (section 3) and isoradius contours (section 4), and how they are extracted using a globally supported RBF. The next section describes the implementation of our four stage depth map generation process. Before our final conclusions section, two sections detail our evaluations. Here, section 6 evaluates SSR histograms and their derivatives, when compared to spin images [35], in the context of facial nose tip identification. Section 7 evaluates isoradius contours, when compared to 'iterative closest points' (ICP) [6], in the context of facial pose alignment.

The Face Recognition Grand Challenge (FRGC) 3D dataset [48] has provided an excellent benchmark to evaluate various 3D face recognition strategies and compare 3D face recognition performance with 2D performance. Despite this, we have elected to augment FRGC based evaluations by also using the University of York 3D (UoY) face dataset (1736 facial scans, 280 subjects)

for evaluation, because the FRGC dataset does not contain test conditions for significant pose variations. Furthermore, the UoY dataset contains subjects with head gear, such as spectacles, in addition to six facial expression variations, and is lower resolution and poorer quality data than the FRGC data. The UoY dataset includes 50% of data in frontal pose and neutral expression, 38% of data in frontal pose and non-neutral expression and 12% of data in non-frontal pose and neutral expression.

The work presented here represents the integration and significant extension of our earlier work [46], [47].

Representation	PI/ PA	HO/ FB	Inter/ Intra
EGI [32]	PA	HO	Inter
Splash [55]	PI	FB	Inter
Shape Hist. [1]	PI	HO	Inter
Sph. harm. SEF [50]	PA	HO	Inter
Sph. harm. EDT [25]	PI	HO	Inter
Light field [16]	PA	HO	Inter
Fishersurfaces [29]	PA	HO	Intra
CRSP [45]	PA	HO	Inter
Keypoints [44]	PI	FB	Intra
This paper	PA	HO	Intra

Table 1 A comparison of a selection of 3D object retrieval methods. First column, pose-invariant (PI) or pose-aligned (PA). Second column, holistic (HO) or feature-based (FB). Third column, inter-class or intra-class retrieval tasks.

2 Related work

In this section, we first give an overview of shape representation in the context of different forms of 3D object retrieval tasks (section 2.1). We then review previous work on 3D local surface descriptors for landmark localisation (section 2.2). Finally, in section 2.3, we review the theory and application of RBF modelling in 3D surface representation and interpolation.

2.1 Shape representation in 3D object retrieval tasks

The 3D object retrieval literature can be considered in the context of a broad three-dimensional categorisation, namely: (i) shape representations that are either pose-invariant or pose-aligned, this relates way in which the retrieval system deals with arbitrary translations and rotations of the object when representing shape; (ii) shape representations that are either holistic or feature-based, this relates to the global/local nature of the shape representation; (iii) retrieval applications that are either inter-class or intra-class [56], this relates to whether the system retrieves fundamentally different object classes (car, table, vase) or different instances of the same class, as in 3D face recognition applications. Of course, this is not the only categorisation and not all 3D retrieval systems fall neatly into these categories, but this is a useful initial framework to discuss the literature. An example of how a small, but broad cross-section of recent work falls into these categories is given in table 1, and we use these three categories to develop our literature discussion in the following three subsections.

2.1.1 Pose-invariant and pose-aligned descriptors

Typically, pose-invariant, holistic descriptors are positioned at the centre of mass of the object and are based on spherical representations encompassing the whole object shape. An early example is Ankerst et al's 3D shape histograms [1], which decompose the shape into a set of concentric shells centred on the object's centre of mass. The object surface area intersected by each shell is stored in a histogram indexed by shell radius, thus giving a 1D array of values to represent global shape.

Often 3D shape has been described as a function on a sphere [32] [50] [25] and this provides the opportunity to compactly describe shape in the spectral domain, using spherical harmonics. These are a set of orthogonal functions that originate from the angular part of the solution to Laplace's equation, expressed in polar coordinates. The low order amplitude coefficients of a spherical harmonic shape decomposition capture gross shape, while higher order coefficients represent the higher spatial frequencies, such as fine surface detail. Typically, phase information of the spherical harmonic function is discarded (for pose-invariance) and thus the amplitude information provides a pose-invariant shape description.

There are several ways of describing shape as a function on one or more spheres, examples include: the Extended Gaussian Image (EGI) [32], which describes shape by accumulating surface area-weighted normal directions into a histogram on the sphere; Spherical Extent Functions (SEF) [50], where shape is described by casting a ray from the object's centre and computing the furthest intersection point on the object surface; and voxel grid binary functions of the object surface, restricted to a set of concentric spheres [25]. In their original form, some of these approaches [32][50] have required an initial PCA based alignment stage (i.e. they are pose-aligned rather than innately pose-invariant). However, Kazhdan et al [37] has shown that employing pose-invariant spherical harmonic representations of these functions gives either a similar or better retrieval

performance than the original PCA-aligned descriptors, depending on the class of object being retrieved.

The main advantage of pose-invariant, holistic representations is that they allow fast matching, both because pose alignment is not necessary, and also because the descriptors tend to be quick to extract and provide compact representations for fast shape matching. Conversely, the main disadvantage of these representations is that, when discarding pose-dependent data, some pose-independent information is lost which can lead to a reduction in the descriptors power to discriminate between different object classes. Indeed, when such descriptors are designed, the aim is to achieve invariance with a minimal compromise in discriminating power.

In contrast to pose-invariant techniques, whole 3D objects may be aligned before matching them and this can be done in two ways: (i) by exhaustive search for an optimal alignment between each pair of objects (probe and gallery), which is typical in inter-class retrieval problems or (ii) by aligning to some common canonical view of the stored models, which is the case of pose-normalisation, and is typical in intra-class retrieval problems.

An example of exhaustive search is the light field descriptor approach [16]. Here silhouette images are generated from projections down to 2D images over the full view sphere. These 2D images are characterised by Zernike moments and Fourier coefficients and matched over all possible alignments. Although this approach is computationally expensive, it generates highly descriptive shape representations that have performed well in inter-class retrieval tasks [54].

The simplest and most efficient way to align to a canonical view is to use the three principal axes of the object surface data, computed using some variant of principal component analysis (PCA). Ankerst et al [1] used this approach when augmenting their shell-based shape decomposition with sectors. However, in its raw form, this can be unreliable when comparing objects of the same class [25], for example, in arbitrary pose 3D face recognition when some of the shoulder area is included in the scan. Further problems that many PCA based approaches need to solve are: a 180 degree ambiguity in the direction of the principal axes, principal axes may switch for shapes that have eigenvalues similar in value, and a vulnerability to outliers in the raw shape data. Recently, Papadakis et al [45] have addressed the pose normalisation problem in inter-class retrieval by applying PCA on both surface points and surface normals (separately). For each query/dataset comparison, both alignments are compared and the distance metric with the smallest value is selected as the match score. The representation that they develop is called a concrete radialized spherical projection (CRSP, detailed in table 1) and this has given excellent retrieval performance on the Princeton Shape Benchmark.

An alternative to PCA based alignment is to align directly to an object template already in canonical pose. Given a set of point-to-point correspondences on a pair of 3D objects that we wish to align, several research groups have shown that we can compute the relative rotation between the two sets of data using least-squares techniques [23], [2], [28]. Once we have the 3D rotation, the relative 3D translation can be computed using the means of the two data sets. The question then becomes: how do we determine point-to-point correspondences? In the *iterative closest points* (ICP) approach of Besl and McKay [6], point-to-point correspondences are determined by using the minimum Euclidean distance (closest points) across the two 3D data sets and these correspondences are iteratively refined, as aligning rotations and translations are computed for each set of new correspondences, until the alignment algorithm converges. If ICP converges successfully, this generally occurs in a relatively small number of iterations, but the algorithm has the disadvantage of converging to local minima if the initial misalignment is too great. To avoid this, an initial estimate of the transformation between the two surfaces is generally achieved with a coarse correspondence scheme, such as that used by Lu [42], where heuristics applied to local, curvature based shape indices are used before application of ICP. Chetverikov et al [17] have developed a 'trimmed' version of ICP in order to improve robustness. Alignment can also be achieved by localising three or more landmarks on the 3D surface and transforming these into the canonical frame [12]. Often this is used as a coarse initial alignment method and ICP is used as a refinement.

The main advantage of pose-aligned (view-based) descriptors is that they can be highly discriminating, as no information is 'washed out' in order to achieve pose-invariance. The disadvantages include the high computational cost of exhaustive search for alignment, or the non-trivial problem of localising landmarks for pose normalisation to a canonical view.

2.1.2 Holistic and feature-based representations

A holistic representation is global in the sense that it captures the whole shape, which has the advantage of using all of the available raw shape data for discrimination within the matching process. Classical holistic approaches in 2D face recognition include the Eigenface approach [59] and the Fisherface approach [4], both of which have been adapted to 3D face recognition [30]

[29]. The disadvantage of such representations is that they are vulnerable to occlusions and shape deformations, such as may be encountered in deformable or articulated objects. Conversely, feature based approaches extract local features, typically at distinctive points on the 3D surface, such as curvature extrema. The global distribution of such local features can be used in structural (graph) matching procedures to match between a probe and gallery graph [44], or the features may be used in hashing procedures [55]. The advantages of such feature-based approaches is that they have immunity to missing parts, such as occurs from self occlusion in 2.5D shape data.

2.1.3 Inter-class and intra-class applications

The category of approach adopted has been dependent on the form of the 3D object retrieval task. In general, pose-invariant, holistic descriptors have been applied to inter-class retrieval problems. For example, spherical harmonic approaches [37][25] have been applied to the Princeton Shape Benchmark inter-class retrieval problem [54]. This accords with the need for compact, efficient, whole-shape descriptions for searching large 3D datasets. (A notable exception to this is Chen et al's light-field descriptor (LFD) method [16], which is a large, view-based representation. With this rich information representation, the LFD system retrieval accuracy was reported to be highly competitive with other methods [54].) In contrast, for intra-class retrieval applications, such as the 3D face recognition applications [36] [31], most researchers have used pose-aligned or pose-normalised descriptors. This accords with the notion that the discriminating power of aligned/normalised descriptors is required to give the necessary fine-grained classification performance [56].

2.2 Local surface descriptors for landmark localisation

The system presented in this paper uses novel 3D surface descriptors for landmark localisation prior to pose alignment or pose normalisation. Thus we now look at previous work related to *local* 3D surface descriptors used for 3D alignment in both recognition and retrieval applications, with particular emphasis on the work applied to 3D facial surfaces.

Historically, many researchers have sought to extract pose invariant 3D surface descriptors. For example, Besl and Jain [5] used Gaussian curvature and mean curvature to categorise surface shape into eight distinct categories. Dorai and Jain [22] developed this to define two new measures, called the 'shape index' and 'curvedness'. Colbry et al [20] use shape index for what they term anchor point localisation. Chang et al [14] use mean curvature and Gaussian curvature to localise the nose tip, nose bridge and eye cavities in 3D face data.

Gordon's work [26] on developing curvature maps for 3D face data was an early example of a local, invariant 3D facial surface characterisation. This curvature was generated with a view to generating discriminating features for recognition rather than localising facial landmarks. However, extrema of curvature have since been used to generate regions of interest over which more discriminating and computationally expensive local descriptors can be extracted to determine a reliable landmark localisation [12].

Three particularly notable local 3D surface descriptors were presented in the 1990s; splash representations [55], point signatures [19] and spin images [35]. Stein and Medioni [55] proposed the 'splash representation' to encode local 3D surface shape. Here, a local contour is extracted, that is some fixed *geodesic* distance from a vertex and surface normals are generated at fixed angular displacements within the tangent plane of that vertex. The angle of the surface normals along the geodesic contour, with respect to the vertex normal, are computed and used as a mechanism for identifying a vertex. The representation is used in a hash table 3D object indexing/retrieval approach, which the authors call 'structural indexing'.

Chua and Jarvis [19] present an alternative, which they call the 'point signature' representation. Here, a sphere is centered on a vertex to provide an intersecting curve, C, with the object surface, that is some *Euclidean* distance from the vertex. The normal of a least-squares plane fit of the points in C and the vertex itself define a reference plane and the heights of the points on the curve, C, relative to this reference plane gives a signed distance profile. Comparison of signatures is made by scanning the signed distance values out from the maximum distance value. If there are several local maxima, the comparison is executed at each local maximum. Point signatures have been used for 3D facial feature detection and 3D face recognition [18], [60].

At around the same time as point signatures, Johnson and Hebert presented the 'spin image' representation [35], which cylindrically encodes shape relative to a local tangent plane. To construct a spin image, both radius and height of neighbouring vertices relative to the local tangent plane are measured and the results are binned into a histogram. Of these methods reported in the 1990s, spin images have been taken up most widely by the research community (see, for example, [3]), perhaps because they are intuitive and simple to compute. More recent work has focussed on matching

multi-resolution pyramids of spin images [21] in order to speed up the matching process. Other researchers have used spin images to localise 3D facial features [12].

Some approaches to 3D facial landmark localisation have adopted rules based on local surface descriptors and their distribution. For example, Xu et al [62] select nose candidate vertices as those points that have maximal height in their local frame. Many of these are eliminated, based on the mean and variance of neighbouring points projected in the direction of the vertex's normal. Final selection of the nose position is based on the most dense collection of nose tip candidates. Segundo et al [53] developed a heuristic technique for nose tip localisation, using empirically derived rules applied to projections of depth and curvature.

An alternative approach to matching local surface descriptors in order to localise 3D surface landmarks, is to use a 3D model, marked up with the relevant landmarks, and then globally align the manually annotated model to the data. The landmarks can then be mapped directly from the model into the data, for example, as closest vertices. This approach was applied to 3D faces by Whitmarsh et al [61]. The key step is the registration process, which uses ICP for a rigid transformation (translation and rotation) and a scaling step, to independently match the height width and depth of the model to that of the data. This approach appears promising, due to its efficiency in localising multiple landmarks simultaneously. However, the method relies on ICP convergence, which is difficult to guarantee in uncropped, arbitrary pose data.

2.3 RBF surface modelling

We use a radial basis function (RBF) model of the 3D facial surface in all four processing stages presented in this paper and so we now present an overview of this 3D surface modelling approach. Scattered data interpolation using radial basis functions has been studied from at least the 1980s [24], with notable contributions by Savchenko et al [51] and Carr et al [11]. Essentially, a 3D object surface is represented implicitly (where the RBF has the value zero), which provides a compact representation with inherent interpolation abilities, since the RBF is defined everywhere in $\Re^3$.

Applications have been widespread and include: automatic mesh repair in range-scanned graphical models [11], cranioplastic skull model repair [10], surface reconstruction in ultrasound data [49], 3D shape transformation [58] and animated face modelling [15], where an RBF is used to transform corresponding 3D feature points between a template face and a face scan. However, the use of RBFs specifically for 3D facial feature descriptors is currently sparse and the only related RBF-based 3D face feature extraction that we are aware of is that of Hou and Bai [33], who use RBFs to detect ridge lines on 3D facial surfaces. This lack of literature is possibly because of the perception of RBF fitting and evaluation being computationally expensive. Indeed, conventional methods for RBF implicit surface fitting to N points requires $O(N^3)$ operations and $O(N^2)$ storage, whereas our implementation employs the fast multi-pole method (FMM) developed by Greengard and Rokhlin [27] and used by Carr et al [11] for interpolating 3D object surfaces. In this method, approximations are allowed in both the fitting and evaluation of the RBF. For example, for RBF evaluation at a particular point, the centres are clustered into 'near field' and 'far field'. The contribution of only those centres 'near' to the evaluation point are directly evaluated and those 'far' from the evaluation point are approximated, allowing a globally supported RBF to evaluated quickly to some prescribed accuracy. This method requires $O(NlogN)$ operations and $O(N)$ storage for the fitting process. For evaluation of the RBF at M points, the algorithm requires $O(NlogN)$ setup operations followed by $O(M)$ operations.

In our work, we closely follow the approach and notation of Carr et al [11]. To briefly recap from their work, a *radial* function has a value at some point in n-dimensional space $\mathbf{x}$, which only depends on its 2-norm relative to another point, called a 'centre'. Hence, in our case, the radial function value is constant over a sphere. A radial basis function uses a weighted sum of basis functions to implicitly model a surface, where the basis function may be Gaussian, cubic spline or some other function, which is radial in form, as shown in equation 1,

$$s(\mathbf{x}) = p(\mathbf{x}) + \sum_{i=1}^{N_c} \lambda_i \Phi(\mathbf{x} - \mathbf{x_i}) \tag{1}$$

For our 3D facial surface RBF model, p is a linear polynomial, λ_i are the RBF coefficients, Φ is a biharmonic spline basis function such that $\Phi(r) = r$, and $\mathbf{x}_i$ are the N_c RBF centres. In fitting a 3D surface, s is chosen such that $s(\mathbf{x}) = 0$ forms a surface that smoothly interpolates the data points $\mathbf{x}_i$. Thus the RBF model parameters implicitly define the surface as the set of points where the RBF is zero. This is called the zero isosurface of the RBF. Note that one can not simply solve the equation $s(\mathbf{x}_i) = 0$ for our N data points, as this yields a trivial solution of $s(\mathbf{x}) = 0$ everywhere. Constraints where $s(\mathbf{x})$ is non-zero need to be used. Since we may readily generate 'off-surface points' using

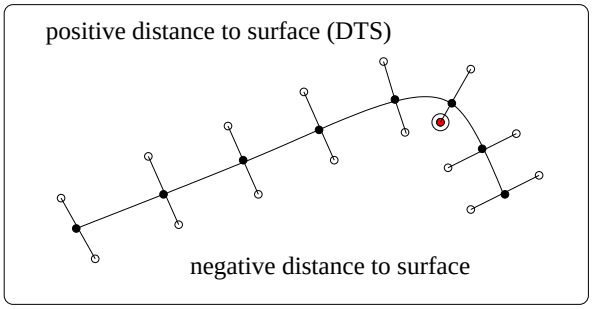


Fig. 1 Adaptive generation of 'off surface' points along the surface normal directions of a nose profile. The point marked in solid red and circled been adapted and brought nearer to the facial surface.

surface normal data, s can be chosen to approximate a signed *distance to surface* (DTS) function.

Figure 1 illustrates the cross-section of a nose, where surface normals are used to generate off-surface points with known (signed) DTS values. In this process, care is essential at regions of high local curvature. In such cases, the distance to the surface has to be reduced on the concave side of the surface in order to avoid generating inconsistent DTS data. Our implementation employs the simple approach of Carr et al [11], which is to validate an off-surface sample point by checking that its nearest surface point is the point, **p**, from which it was projected. If this is not the case, then the projection distance is progressively reduced until the nearest point is **p**.

We use the biharmonic spline as the RBF basis function, as this is known to be the smoothest interpolant in the sense that it minimises a certain energy functional associated with the fit, producing an implicit surface with minimal curvature. Thus it is well suited to representing 3D object surfaces [11]. We perform a globally supported RBF fit and when we have performed the fit once, it can be evaluated anywhere in $\Re^3$ where we need to determine a signed distance to the object surface, through all four stages of the depth map generation process described in this paper. By convention, points below the facial surface (inside the head) are negative, those above the facial surface are positive and those on the facial surface are zero.

3 Spherically-sampled RBF (SSR) descriptors

In spin images [35], a surface point uses its associated surface normal to form a basis with which to encode neighbouring points. Neighbouring point positions are encoded in cylindrical coordinates, as the radius in the tangent plane and height above the tangent plane. All points are binned onto a fixed grid. Corresponding 3D points across a pair of similar objects can be matched by a process of correlation of spin images or any other matching metric. Issues in spin image generation include (i) noise affecting the computation of the local surface tangent plane and (ii) problems of appropriate bin size selection. Due to these issues, we were motivated to make use of an RBF model to generate invariant 3D surface descriptors, which we call *spherically-sampled RBF* (SSR) surface descriptors.

3.1 SSR shape histograms ('balloon images')

Here we propose a new kind of local surface representation, which can be derived readily from the RBF model and we call this an *SSR shape histogram*. To generate such an SSR shape histogram, we first distribute a set a n sample points evenly across a unit sphere, centered on the origin. To do this, we employ the octahedron sub division method, which, for K iterations, generates $n = \alpha\beta^K$ points. The constants are $[\alpha,\ \beta]^T = [8,\ 4]^T$ and we use $K = 3$, which gives $n = 512$. The sphere is then scaled by q radii, r_i, to give a set of concentric spheres and their common centre is translated such that it is coincident with a facial surface point. (Note that this can be a raw vertex, but can also be anywhere between vertices, on the RBF zero isosurface).

If a sphere of radius r_i is placed at some object surface point, then the maximum distance of any point on that sphere from the object surface is r_i, implying that typical maximum and minimum evaluated RBF values for a flat object surface region are $+r_i$ and $-r_i$ respectively. Thus a reasonable normalisation of RBF values is to divide by r_i to give a typical range of [-1, 1] for normalised RBF distance-to-surface values. Such a normalisation allows RBF values distributed over a wide range of radii to be accumulated into the same local shape histogram.

The RBF, s, is evaluated at the $N = nq$ sample points on the concentric spheres, and these values are normalised by dividing by the appropriate sphere radius, r_i. If this normalised value, $s_n = \frac{s}{r_i}$, is binned over p bins, then we can construct a (pxq) histogram of normalised RBF values, which may, for visualisation purposes, be rendered as a 'balloon image'. (Note that the balloon analogy comes from incrementally inflating a sphere through the 3D domain of the RBF.) Examples of balloon images for the protruding nose and flat forehead are given in figure 2. Here we use 8 radii ranging from 10mm to 45mm inclusive and we accumulate the normalised RBF values into 23 bins from -1.1 to 1.1 in steps of 0.1. We use a slightly larger range than [-1 1] to ensure that all RBF values are accumulated.

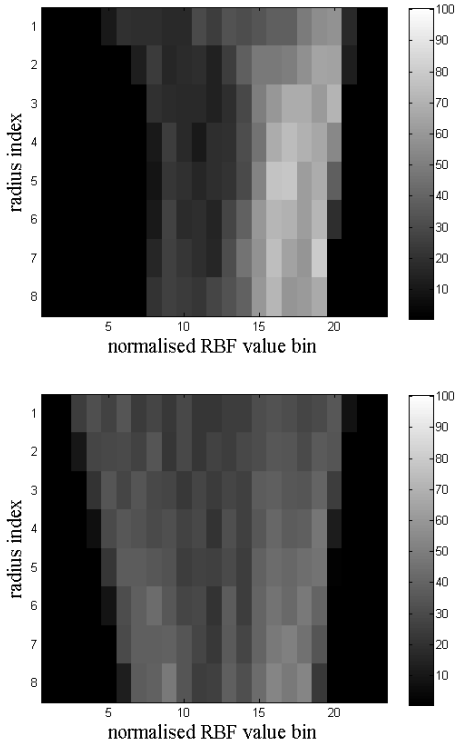


Fig. 2 Spherically sampled RBF (SSR) histograms generated over 8 radii and 23 normalised SSR bins: nose tip (upper image), forehead vertex (lower image)

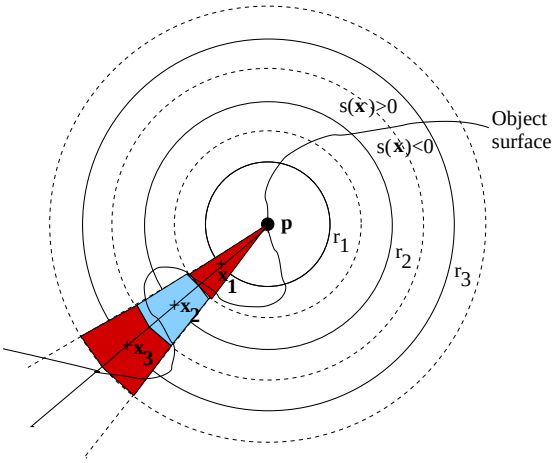


Fig. 3 Computation of an SSR value, a measure of the volumetric intersection of the object (head) and a sphere, centred on the object surface. This is an indicator of surface convexity at a selected scale. The two red shaded sectors have positive RBF evaluations and the blue shaded sector has a negative evaluation.

3.2 SSR values

Clearly, the convexity of the local surface shape around some point is related to the brightness distribution of the balloon image. This motivates us to consider how SSR histograms may be processed to give a pose invariant convexity value for high resolution, repeatable landmark localisation. For example, if we wish to localise the nose tip, we may first define the nose tip as the point on the facial surface where a sphere of appropriate radius (centered on that point) and the face have minimum volumetric intersection. We then need to consider how to calculate the volumetric information from the SSR histogram and our approach is illustrated in figure 3. In this figure, the point $\mathbf{p}$ is on the object (face) surface, the upper left part of the figure is above the object surface ($s(\mathbf{x}) > 0$) and the lower right part of the figure is below the object surface ($s(\mathbf{x}) < 0$). We have illustrated three concentric spheres (solid lines) of radius (r_1, r_2, r_3), separated by Δr over which the RBF is sampled and we consider three co-radial samples for each of these radii at $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ respectively, noting that $s(\mathbf{x}_1) > 0$, $s(\mathbf{x}_2) < 0$ and $s(\mathbf{x}_3) > 0$. The dashed circles in the figure indicates the position of (non-sampling) concentric spheres that bound volumetric segments, and these have radii, ρ_i midway between the sampling spheres, namely at $\rho_i = \frac{(r_i + r_{i+1})}{2}$. In order to determine an estimate of the total volumetric intersection within the outer (dashed) sphere of radius $\rho_3 = r_3 + \frac{\Delta r}{2}$, we need to sum all of the volumetric contributions centred on radial sampling directions with $s(\mathbf{x}_1) < 0$, over all sampling radii and all sampling spheres.

In figure 3, the central blue shaded volumetric segment contributes to the object/sphere intersection, but the two outer red shaded volumes do not. Note that the segments centred on the larger radii have bigger volumes, and thus a weighting vector needs to be applied to the summation. Thus the volumetric intersection, V_p, at point $\mathbf{p}$ is given by:

$$V_p = \frac{k}{n} \mathbf{v}^T \mathbf{n}^- \tag{2}$$

where $k = \frac{4\pi}{3}$ is a constant related to the volume of a sphere, n is the total number of sample points on a sphere, $\mathbf{v}^T$ is a vector containing the q volumetric weights (one for each radius), and $\mathbf{n}^-$ is a vector where each element is the count of the total number of sample points on a given sphere in which $s(\mathbf{x}) < 0$.

An equivalent, but more elegant approach, is to define a metric that is a relative measure of the volume of the sphere that is above the object surface compared with the volume of the sphere below the object surface. With this in mind, we define a SSR based convexity value for the point, $\mathbf{p}$, as

$$C_p = \frac{k}{n} \mathbf{v}^T [\mathbf{n}^+ - \mathbf{n}^-] \tag{3}$$

where $\mathbf{n}^+$ is a vector in which each element is the count of the total number of sample points on a given sphere where $s(\mathbf{x}) > 0$. With this metric, a highly convex shape will have a value approaching 1.0, a highly concave shape will have a value approaching -1.0 and a flat area will have a value close to zero. This can be clearly seen from equation 3, where the elements in $\mathbf{n}^+$ and $\mathbf{n}^-$ will be similar, giving a near zero vector on the right of the equation. In its simplest form, a very approximate SSR value can be computed using a single sphere, which makes both the constant k and the volumetric weighting vector $\mathbf{v}$ in equation 3 redundant. We use this form in this paper, which amounts to averaging the signs of n RBF evaluations over a sphere.

$$C_p = \frac{1}{n}\sum_{i=1}^{n} sign(s_i) \tag{4}$$

In order to illustrate the potential of this technique, a single sampling sphere of radius 20mm and 128 sample points is moved over a facial surface. Figure 4a, illustrates the RBF distance-to-surface values of this facial surface by a colour mapping and the RBF sampling sphere (yellow) is shown positioned close to the nose bridge. The resulting SSR value map is shown from different views in figures 4(b),(c),(d). A surface is rendered over this plot to aid visualisation, where the lighter areas have a convexity value near to +1 and the darker areas are close to -1 (i.e. concave). The figure indicates that, in this case, the nose is the peak convexity value in the map. Note also that the inner eye corners have high concavity, suggesting that they are also good landmarks to localise with this descriptor.

3.3 SSR descriptors: A comparison with the literature

To our knowledge, the closest work to SSR histograms in the literature is Johnson and Hebert's spin images [35]. Although our method requires a global set of normals to computed the RBF, unlike the spin image, a local normal is not required to encode points in a local frame. We hypothesise a number of advantages that SSR histograms may have over spin images: (i) Missing parts or any residual data spikes may corrupt the local normal estimate, which can have a big influence on the spin image; (ii) This is likely to be exacerbated in areas of high curvature, such as the nose tip, particularly, when the raw vertex data is of limited resolution; (iii) Missing parts can corrupt the content of spin-images, unless an effective interpolation process is implemented. For SSR histograms, the interpolation is implicit in the method, as the RBF is defined everywhere in 3D space; (iv) Issue of correct bin-size selection is an issue in spin-images, but is not a problem for SSR histograms, because we choose a set of radii explicitly; (v) Local density of points is an issue for spin images, but again this is not a problem for SSR histograms, because we choose the number of sampling points on the concentric sampling spheres explicitly. In section 6, we evaluate SSR histograms and compare them to three variants of spin-image, of the same size and resolution.

Given that we employ spherical methods, we now compare our approach with the general application of spherical harmonics to shape representation. Generally speaking, spherical harmonic methods have been applied to global shape representations, rather than local surface representations and they have been used either to achieve pose-invariance, or to generate a compact shape descriptor for efficient matching or both. The reasons why we did not apply the Spherical Fourier Transform to our RBF 'distance-to-surface' function, defined on local concentric spheres are: (i) local shape descriptors need to be computed at potentially many surface points on the same 3D object, which can be computationally expensive; (ii) the SSR histogram is already inherently pose invariant for a sufficiently large number of samples on the sampling spheres and (iii) we achieve compactness by projecting the SSR shape histogram into a reduced dimension space, using standard PCA. Nevertheless, we believe that there are several interesting avenues of research to be explored, by applying spherical harmonic methods to RBF shape models evaluated over concentric spheres. For example, the RBF could be evaluated over a global set of concentric spheres and spherical harmonic methods could be applied to encode holistic shape in an inter-class retrieval application. This is particularly attractive when the raw 3D object data has missing parts, as is the case when shape data is derived from 3D sensor systems.

Since any arbitrary pose 3D point cloud can be interpolated to give depth values over a regular Cartesian grid, we can represent 3D shape (or rather 2.5D shape) as depth maps, also referred to as range images. This means that we can apply any feature detectors available that may have initially been developed for standard 2D intensity images. A seminal example of this is the scale invariant feature transform (SIFT) algorithm, developed by Lowe [41], which has proved to be one of the most successful feature detectors used by the Computer Vision community. It has been widely used on standard 2D intensity images in a range of applications including object recognition [40], matching objects in video sequences [34] and robot navigation [52]. In order implement a small scale test of the SIFT algorithm on 3D facial depth maps, we have used the publicly available version 4 of SIFT from David Lowe's web pages

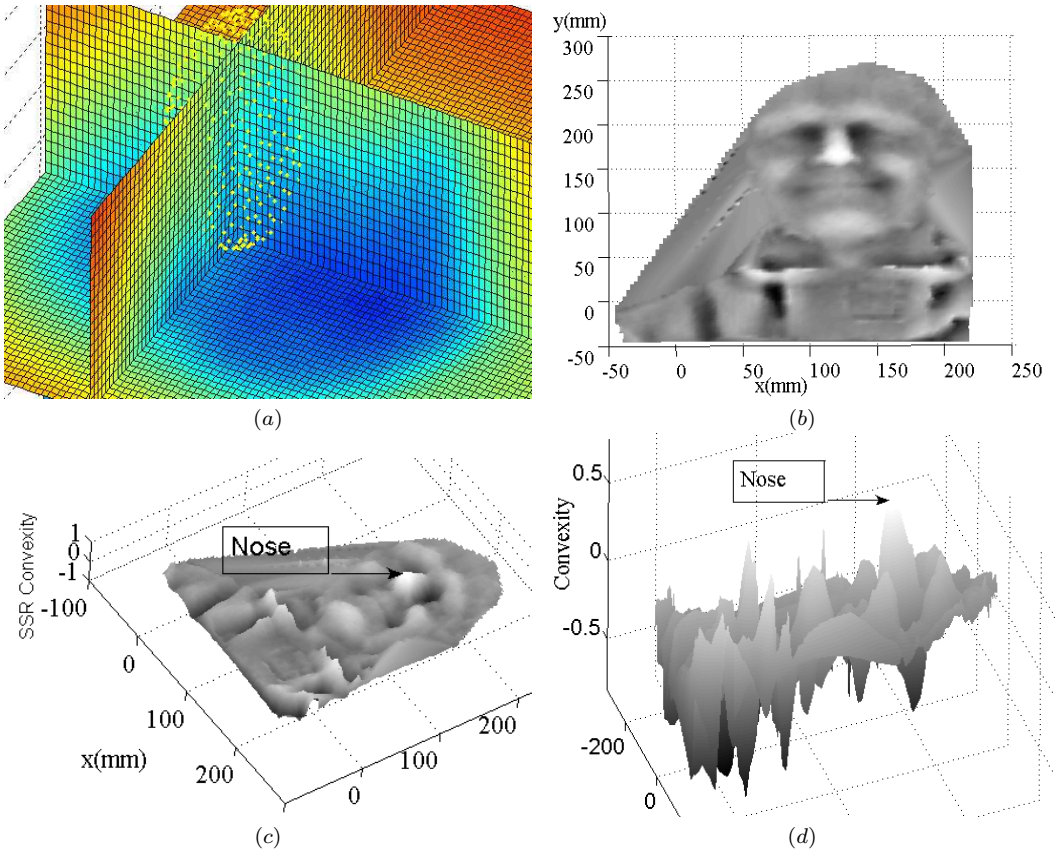


Fig. 4 (a) Top left shows spherical sampling of the RBF. The blue areas are negative RBF values (below the facial surface), yellow/red areas are positive RBF values (above the facial surface) and the turquoise areas contain the zero RBF isosurface (facial surface). Plots (b,c,d) in grey show the SSR values (convexity) of the same face from three different views.

at the University of British Columbia. Figure 3.3 shows the results of SIFT when applied to 60x90 depth maps from the UoY dataset. Frontal poses are shown in the left column and poses looking down are shown in the right column. All SIFT feature with scale values greater than 2 are shown and nose and eye features have been manually colored in red. Since the nose tip lies on the plane of bilateral symmetry, this often causes SIFT to generate a pair of dominant orientations for the same nose tip keypoint. This is because, in the SIFT algorithm, dominant directions for local gradients are detected as peaks in the SIFT orientation histogram. In the algorithm, the highest peak is detected and any other peak that is within 80% of this highest peak is also retained, creating a pair of coincident keypoints with different orientations. Also, as head pose changes (see figure 3.3), the dominant orientation of the keypoint changes, which is dependent on head pose; worse still, the keypoint descriptor itself must, in general, change because the changes in the depth map around a facial landmark over out-of-plane rotations can not be modelled as similarity transforms, which is the class of transforms over which the SIFT algorithm is designed to be invariant. If we compare SSR descriptors to the SIFT approach, the extrema in the SSR value function are our interest points (for example maxima at nose tip, minima at inner eye corners, see fig 4) and are analagous to SIFT keypoints and SSR histograms are our descriptors, analagous to SIFT's orientation histogram descriptor. Both our interest point generator and descriptor are based on spherical representations in 3D as opposed to being based on a depth signal defined on an orthogonal, regular grid. This property provides significantly greater immunity to out-of-plane pose variations than is afforded by SIFT operating on single viewpoint depth maps.

4 Isoradius contours

Once the nose tip has been localised using SSR descriptors, as will be described in detail in section 5.2, we use our second new representation, called the 'isoradius contour', to align a pair of faces. This can be used in two ways. Firstly, as a direct alignment method be-

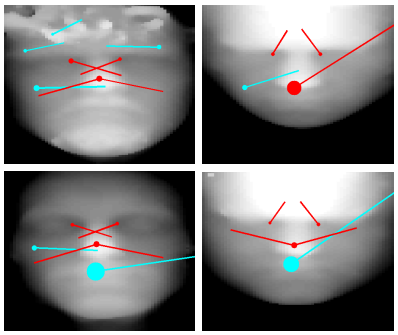


Fig. 5 SIFT features (scale greater than 2) in 60x90 *unaligned* facial depth maps (generated by sampling UoY dataset RBF models). Frontal pose (left column) and looking down (right column). Nose and eye corner features are manually colored in red.

tween any pair of faces, both of which are in non-specific poses. In this case, once the optimal alignment is determined, depth maps for both faces are generated ready for feature extraction and matching. Alternatively, if a particular face (such as an average face) is known to be in canonical pose (frontal), it can act as a reference face to align all other faces in a dataset to the same canonical pose. This is useful when we wish to build statistical models of depth map variation, which requires the depth maps to be pose-normalised.

An isoradius contour is a space-curve defined by the locus on a 3D surface that is a known fixed distance relative to some predefined reference point. Thus an isoradius contour (IRAD) can be thought of as the intersection of a sphere, centered on that reference point, with the object surface. (We note that this is the same space-curve definition that is used in the point signature method [19], although highly sampled contours using RBF models are not used in this point signature work. In addition, we encode shape information around the contour differently, and we use the space-curve for pose alignment, rather than identification of a 3D point.)

In the case of faces, an obvious choice for the reference point (sphere centre) is the tip of the nose. Clearly the shape of the intersection of the sphere with the face is independent of the 3 DOF head orientation, due to the infinite rotational symmetry of the sphere. This pose invariance is a major benefit of the representation. To encode the shape of the contour, we compute its local curvature tangential to the sphere and we call this an IRAD curvature signal. If IRAD curvature signals are scanned out in a consistent manner, that is in an anticlockwise direction around the nose tip normal, then these signals are pose invariant, modulo a rotational phase shift. This suggests that we can align a pair of faces by a process of 1D curvature signal correlation, applied across a pair of IRAD curvature signals (one on each face) derived using the same sphere radius. Thus, we can generate an IRAD curvature correlation signal by sliding the smaller curvature signal exhaustively over the larger curvature signal. This correlation signal constrains the possible rotational alignments to a set of n, where n is the number of points on the larger of the two contours, typically around 150 using 1mm contour steps over a 30mm sphere radius. We hypothesize that the best rotational alignment occurs within this set of n alignments, where the IRAD curvature correlation signal is a maximum.

4.1 Extracting Isoradius Contours

In order to extract an isoradius contour, we need to intersect a sphere of specific, known radius, with the facial surface, when that sphere is centred on the localised nose tip. In order to generate an IRAD of radius R, we make extensive use of the the RBF model that we have generated within an IRAD 'point chaining' procedure, which consists of the following steps:

1. *Find a starting point,* $\mathbf{p_1}$*, on the facial surface.* Here 'facial surface' is defined by the zero isosurface of RBF model. In order to do this, we generate a circle, radius R, centered on the nose tip. This circle resides in a plane defined by the two eigenvectors of the point cloud around the nose tip that have the two smallest eigenvalues. This guarantees that, for a sufficiently small radius, the circle will intersect the facial surface and we simply have to interpolate any zero-crossing of the RBF (distance to surface) function evaluated on the circle, to find a starting point for the contour.
2. *Localise an appropriate second point,* $\mathbf{p_2}$ *on the facial surface.* We now generate a small circle of radius r, centered on the starting point $\mathbf{p_1}$ (described above), which sits on the surface of the IRAD sphere (shown in red in figure 6). Note that r is the step length over which we chain the IRAD contour and we use $r = 1mm$. Again the RBF model can be used to find where this circle intersects the facial surface, by computing the RBF values over sampling points on the circle and interpolating the locations where the RBF value is zero. We obtain a pair of zero-crossings and, in contrast to step 1, here we need to choose the correct zero crossing (facial surface point), such that the isoradius contour starts to circle the nose tip in a consistent, anticlockwise (right handed) sense. This is done by checking the direction of the cross product between two vectors, the first of which is from the nose tip to $\mathbf{p_1}$ on the contour and the second of which is from $\mathbf{p_1}$ to $\mathbf{p_2}$.

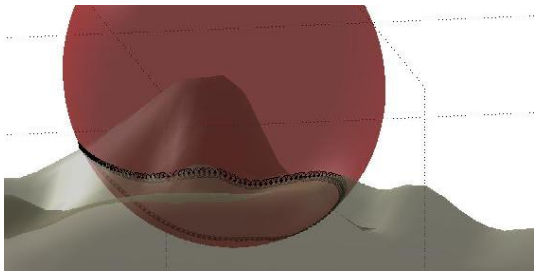


Fig. 6 The IRAD chaining process generates a high density of points at the intersection of a sphere and the facial surface.

3. *Chain IRAD points around the nose tip.* Once we have found $\mathbf{p_2}$, a small circle centered on $\mathbf{p_2}$, radius r, and on the IRAD sphere surface can be generated. Again the RBF evaluations on this circle will have a pair of zero-crossings. This time, however, the cross product direction check is not required, because one zero crossing is very close to $\mathbf{p_1}$ and so can be ruled out. In this way, we chain around the intersection of the IRAD sphere and the facial surface by selecting the $\mathbf{p_{i+1}}$ RBF zero-crossing as the one most distant from $\mathbf{p_{i-1}}$.
4. *Terminate chaining process.* When the chain comes within a threshold distance ($\frac{r}{2}$) of the start position, then the chaining process is halted.

The IRAD chain, consisting of intersecting circles on the surface of the IRAD sphere, at the junction of the IRAD sphere and facial surface is illustrated with real data in figure 6. The ouput of this process is a set of points in 3D space that are a distance R from the nose tip and a distance r from their two neighbouring points (with the exception of the first and last point). A set of contours over a range of radii, for the purpose of illustration, are shown in figure 8. The question now is how to encode this contour and this is dealt with in the following subsection.

4.2 Encoding the contour

To encode the IRAD contour, we measure the IRAD space-curve curvature that is due to the face shape, rather than the curvature that is simply due to the fact that the IRAD is distributed across the surface of a sphere. Put simply, over a step r, the space-curve can turn to the left on the IRAD sphere surface or turn to the right, both by varying degrees, or continue straight on.

The process is illustrated at the centre of figure 7. Given that curvature, $\kappa = \frac{\Delta\theta}{\Delta s}$, and if we maintain a constant step length, Δs, along the isoradius contour, then the angular changes, $\Delta\theta$, encode the contour shape.

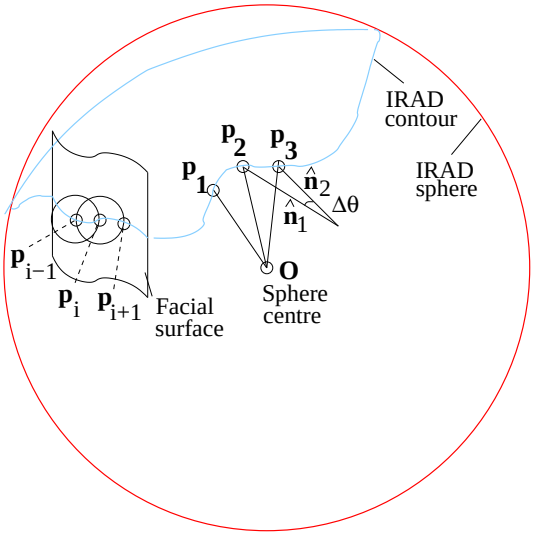


Fig. 7 Extraction of an IRAD and encoding of its tangential curvature

How do we actually compute $\Delta\theta$ along the contour? Consider three consecutive points $(\mathbf{p_1}, \mathbf{p_2}, \mathbf{p_3})$ on the contour, separated by a fixed, but small Δs, as shown in figure 7. A normal to the contour, $\hat{\mathbf{n}}_\mathbf{1}$, is approximated as the cross product of the two vectors $\mathbf{Op_1}$ and $\mathbf{Op_2}$, where $\mathbf{O}$ is the centre of the IRAD sphere. This vector can be recomputed for points $\mathbf{p_2}$ and $\mathbf{p_3}$ using the cross product of $\mathbf{Op_2}$ and $\mathbf{Op_3}$ to give the vector $\hat{\mathbf{n}}_\mathbf{2}$. The change in angle of these normal vectors, $\Delta\theta$, is the angle that we use to encode shape in a pose invariant way. Given that, for sufficiently small r, we approximately move along the IRAD space-curve in even steps, this change of angle approximates a curvature, which is in a plane tangential to the IRAD sphere at the given point on the space-curve. Examples of 30mm IRAD curvature signals for different head poses is shown in figure 9. Note that these are approximately the same shape and differ by small phase shifts. The phase shifts are less than one might expect due to the adaptive way of generating the starting point of the contour. The figure also shows how the use of a 10th order low-pass Butterworth filter can reduce noise in these curvature signals.

4.3 The effect of facial expression on IRADs

We have observed that isoradius contours can slide across non-rigid parts of the facial surface and deform under varying facial expression, particularly in the lower hemisphere of the face, which includes the jaw area. In order to illustrate this, we extract a set of four isoradius contours (r=30mm, 38mm, 46mm, 54mm) on the facial surface of the same subject, under two conditions:

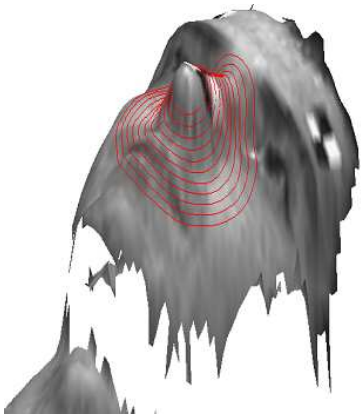


Fig. 8 Isoradius contours extracted over eleven different radii for illustration purposes. For 3D face alignment, we use a single 30mm isoradius contour, which traverses the central nose bridge area and upper lip area.

mouth open and mouth closed. The extracted contours are shown in figure 10, where the color red is used to mark 'mouth closed' isoradius contours and blue is used to mark 'mouth open' isoradius contours.

We have noted that the isoradius contours vary very little across the nose bridge and upper part of the face, whereas they do vary in the lower half of the face, the degree being dependent whether the contour falls on an area of significant surface deformation.

We are able to significantly reduce the influence of facial expression on our facial alignment process in the case when we match to a reference face in a known canonical pose. Here, we match the full isoradius contour of a face to be aligned (in this case, the 'mouth open' face), to a smaller isoradius contour that only contains the rigid nose bridge area of the reference face (in this case, the 'mouth closed' face). This nose bridge region provides a very strong feature for the isoradius curvature correlation to lock onto. When seeking the maximum correlation, we exhaustively shift the smaller reference contour curvature signal relative to the larger, full contour signal of the face to be aligned.

Figure 10 c, shows the isoradius contours after this alignment process (the full contours of the reference are shown in red for comparative purposes). Clearly, the upper parts of the contours are closely matched over the nose bridge area, whereas the contours in the lower part of the face are quite different. The largest two 'open mouth' contours marked in blue fall down into the mouth region, giving a radically different shape to the contours in the lower part of the face. Since only the upper part of the face is used in alignment, the process is successful and the result is shown in figure 10 d. Examination of this figure shows that the alignment is clearly better in the upper part of the face than the lower part. Finally, we note that the smallest IRAD

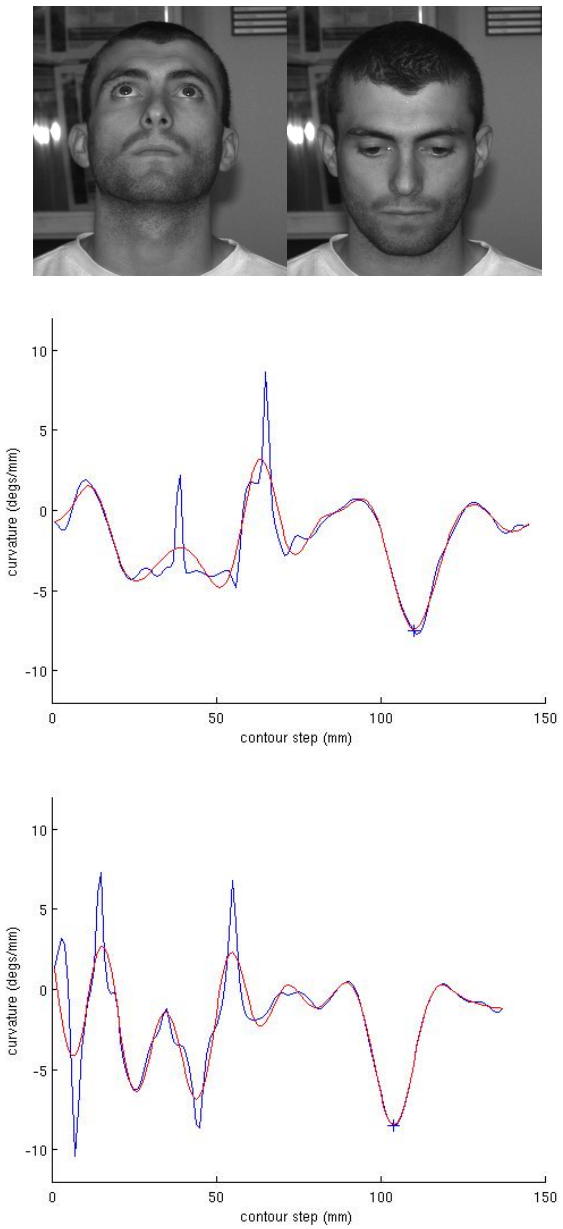


Fig. 9 IRAD curvature signals for the different head poses shown at the top of the figure. Raw curvature data is shown in blue and low-pass filtered data is shown in red. The upper graph shows the signal associated with 'looking up' pose and the lower graph shows signal associated with 'looking down' pose. The blue cross shows the manually marked position of the nose bridge in each case.

shown (radius 30mm) may be more desirable in terms of avoiding 'open mouth' face regions for typical nose sizes, if we were to perform alignments using a pair of full contours both of which fully encircle the nose.

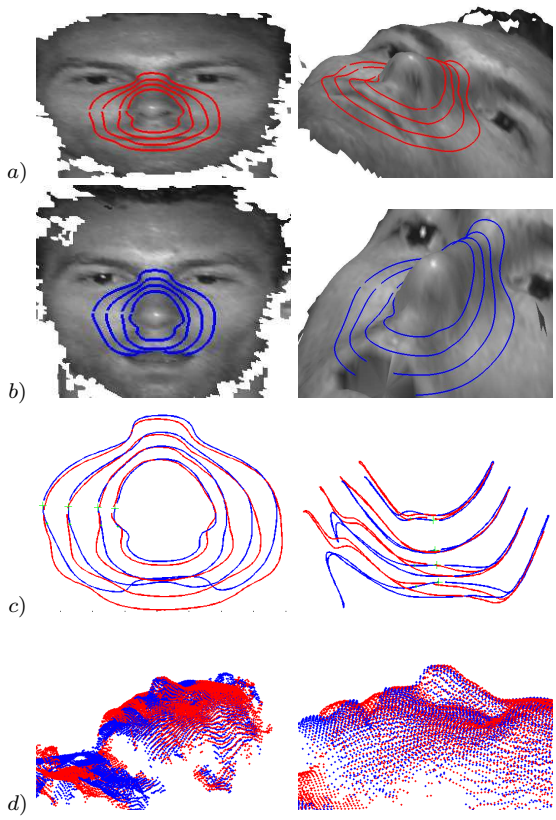


Fig. 10 The influence of mouth closed(red)/open(blue) on isoradius contours (radii=30,38,46,54mm). a) Mouth closed. b) Mouth open. Note that isoradius contours fall under the texture map in the mouth area. c) Isoradius contours after alignment: front view and profile view (associated with d, right). d) Aligned point clouds

4.4 IRADs: A comparison with the literature

The closest related works to our concept of isoradius curvature signals are Stein and Medioni's splash representations [55] and Chua and Jarvis' point signatures [19]. Firstly, the splash representation generates geodesic contours around the surface, which are more difficult contours to compute than isoradius contours. Secondly, we do not attempt to extract a set of piecewise linear structural features from the data around the contour. Breaking a softly curved organic structure such as a human face into a piecewise linear segments can be unstable. In contrast, we extract signals that can be matched by a straightforward process of one-dimensional signal correlation. Note that, unlike 'point signatures' [19], we have not used a local plane normal estimate to encode our signal, as this plane (defined as the least squares fit of the contour) will be affected both by facial expression changes and missing parts. Any deviations in this plane have a global impact on the descriptor, as is the case with spin images. In contrast, our method maintains a consistent signal for all rigid sections of the surface, regardless of any structural changes in other regions. For example, the curvature signal associated with the part of the contour passing through the rigid nose bridge is not affected by the same contour passing through the malleable mouth area. The tradeoff made is that the difference operators that we use to compute curvature tend to amplify surface noise, which is detrimental to performance if the facial surface defined by the RBF model is not smooth. However, we mitigate this effect with the use of a 10th order low-pass Butterworth filter applied to the curvature signals before they are correlated.

5 Algorithm for depth map generation

We now describe each of the four stages of generating pose-normalised depth maps from noisy 3D point-clouds using our RBF model. These steps are (1) filter the data automatically (section 5.1), (2) localise the nose tip (section 5.2), (3) compute the face orientation (section 5.3) and (4) generate a pose-normalised depth map (section 5.4). Section 5.5 gives typical computation times for our system.

5.1 Automatic noise filtering

All non-synthetic 3D point cloud data, collected from 3D imaging systems, is noisy in the sense that it contains both spurious data, such as spikes and pits (inward pointing spikes), which are not associated with the surface of interest, and missing parts where no surface data is available. Spikes and pits generally occur due to incorrect correspondences in a stereo matching process or due to clutter in the scene. Missing parts can occur when the surface reflectance is undesirable, such as the specular surfaces on spectacles and oily skin patches, or the poor reflectance of eyebrows, facial hair and head hair. They also occur due to self-occlusion, for example, when the nose occludes the cheek in a partial side-view of the face. Many researchers have dealt with noise using very simple filtering masks on ordered data. We have designed a more sophisticated approach that does not require data ordered on a grid and establishes a self-consistent set of surface normals.

We use an aggressive filtering policy, in the sense that we would rather remove some valid points from the face surface data than leave in spurious points, such as small data spikes. This is because we can always interpolate, using our RBF model, over regions in which there is missing data, whereas residual noise after the filtering process corrupts the RBF model on which both

surface interpolation and our new invariant 3D feature descriptors are based. Our method of filtering the data is premised on (i) the nose being the most locally convex point that we are interested in and (ii) the inner eye corners being the most locally concave point that we are interested in within our depth map outputs. The method consists of the following steps.

1. *Remove long arcs and isolated meshes.* The UoY dataset contains mesh data, in addition to 3D point-cloud data and texture mapping data. We use this to remove long arcs of above 12mm and then we identify how many submeshes we have. Each of these is checked for vertex count and those below 10% of the total vertex count are removed.
2. *Compute normals and DLP values.* The surface normal around a spherical neighbourhood (radius = 10mm) is computed by finding the eigenvectors of this localised point cloud, $\mathbf{x}_i$, computed using singular value decomposition (SVD). The eigenvector with the smallest eigenvalue describes the surface normal, $\mathbf{n}$. We check the z-component of the normal to ensure that it is pointing away from the centre of the head towards the camera. The distance to local plane (DLP) $d_i = \mathbf{n}.(\mathbf{x}_i - \bar{\mathbf{x}})$ is also computed as a computationally cheap means of measuring local convexity/concavity.
3. *Remove noisy and isolated vertices.* The DLP value is compared to the mean DLP value for a set of nose vertices from 100 training images. If the vertex DLP value is greater than four standard deviations above the mean value for a nose, then the vertex is flagged as a spike. Similarly, if the DLP value is less than four standard deviations below the mean value for an inner eye corner, then the point is flagged as a pit (negative spike). If there are insufficient neighbours (less than 3) to compute a DLP value, then the point is flagged as 'isolated'. All such vertices (spikes, pits and isolated points) are removed from the data.
4. *Repeat steps 2 and 3 until there are no corrupted normals.* If there are any spikes, pits or isolated points in the neighbourhood of some vertex, then the normal of that vertex is considered corrupted. Thus both normal and DLP value for that vertex are recomputed after the corrupting points have been removed. Clearly this could generate new spikes and pits when the normal vectors adjust their orientation, and so iteration of steps 2 and 3 is required until all normals are considered to be free from noisy data. Note that there is no data-replacement policy at this stage, which could cause some vertices to be repeatedly culled an then re-introduced.
5. *Generate RBF model from valid point-set.* Given a filtered set of data points, with a set of normals that are self-consistent, it is now appropriate to generate an RBF model of the face.
6. *Compute distance to surface values for noisy vertices and reinstate some vertices.* We have a list of points that have been filtered from the original dataset. It is straightforward to compute the RBF 'distance to surface' values for this list of points with a single function call. Those vertices with a distance to surface value of close to zero can be reintroduced into the valid vertex list. This re-instatement can occur when, for example, an isolated vertex lies on the facial surface.

The left column of figure 11 shows typical raw data in the UoY 3D face dataset. This 3D data is shown from two views: a frontal view and a view from under the chin to show depth variations in the data. The corresponding 2D image for which the 3D scan was taken is shown on the bottom left of the figure. The output of our filtering process for this data is shown in the right column of figure 11. The spurious data has been cleaned away successfully, but there are large gaps in the data around the brow area, for example, where we can see specular reflection in the texture image. Also in figure 11, we show a new facial mesh that has been derived from the zero-isosurface of the RBF, fitted to the filtered raw data. Note that this zero-isosurface mesh, generated from a standard 'marching cubes' algorithm [39], is used here simply to illustrate the interpolation power of RBF model fitting. Note that, in the algorithm described in this paper, we never need to generate a global zero-isosurface, other than for the final regular grid depth map interpolation (stage 4). However, a small, localised, high density zero-isosurface is generated around the identified raw nose tip vertex (in stage 3), in order to localise the nose tip to sub-vertex resolution. This is particularly useful if the nose tip area itself has missing data, either in the raw scan or due to vertex removal in the noise filtering process.

5.2 Nose tip identification and localisation

Generating and matching SSR histograms over all vertices is computationally expensive, thus we *identify* the raw nose tip vertex via a cascaded filtering process, as illustrated in figure 12 from left to right. We then apply a *localisation* refinement by maximising the SSR value, in the local vicinity of the identified raw vertex, using a local high density RBF-derived zero isosurface (see top to bottom path on the right of figure 12). The concept here is to use progressively more expensive operations to eliminate vertices. The constraints (thresholds) employed at each filtering stage are designed to be weak,

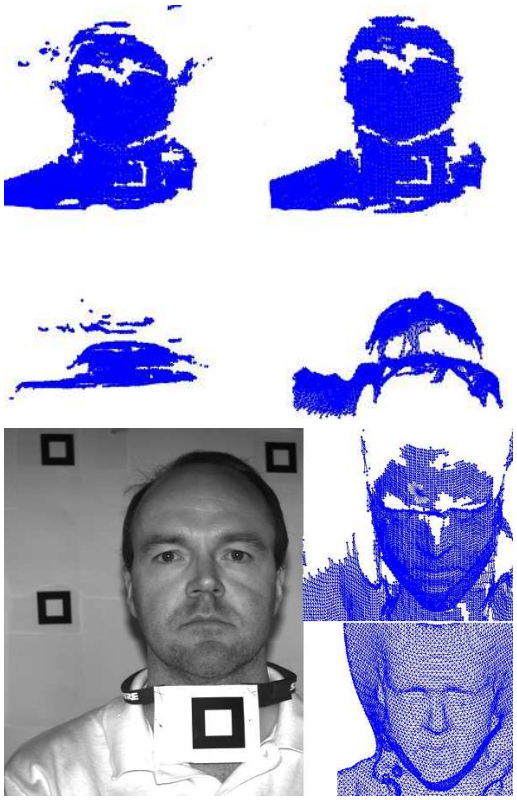


Fig. 11 The filtering process. Left column shows raw UoY data (top and middle left are 3D, bottom left is 2D). Right column shows filtered 3D data and an RBF interpolated face (bottom-right), generated from a 'marching cubes' style algorithm. This is for illustration purposes: we do not need to compute this interpolated surface in order to generate our SSR descriptors, which are highly immune to missing parts in the raw data.

by examining trained nose feature value distributions, so that the nose tip itself is never eliminated. Conceptually, this amounts to considering every vertex as a candidate nose position, where all but one vertex are 'false positives'. Then, at each stage, we apply a filter to reduce the number of false positives, until we have a small number of candidates at the final stage, at which point our most expensive and discriminating test is used to find the correct vertex.

The feature that we use in filter 1 is a *distance to local plane* (DLP), which has already been used to remove data spikes. The filter uses a weak threshold, which is four standard deviations around the average DLP value for nose tips in the training set.

In filter 2, we compute SSR values using a single sphere of radius 20mm with 128 sample points and, again, we set a weak threshold based on the Mahalnobis distance to the mean SSR value in the training data. At this stage, we have multiple local maxima in SSR value (see figure 4d) and so we find these and eliminate all vertices that are not local maxima. Finally, we use SSR shape histograms to select the correct nose vertex by finding the minimum Mahalanobis distance to the average nose-tip in a reduced dimensional space defined by the training dataset. This nose position is refined to sub-vertex resolution by selecting the maximum SSR value over a small, local, high density zero isosurface of the RBF.

Figure 13 shows the nose candidates for each stage in the filtering process. 3D vertices are mapped into the registered texture image for clearer visualisation.

5.3 Pose computation

In section 4 we defined an isoradius contour (IRAD) and showed how to extract an IRAD curvature signal. Since head pose changes shift this signal in a rotational sense, we use a process of 1D correlation to align IRAD signals, by searching for the maximum correlation value over all possible rotational phases shifts. Of course, in the correlation process, we need to deal with IRAD signals of different sizes. For now, lets suppose that the two signals are the same size. We express these signals as discrete data sets: $\mathbf{x} = [x_1 ... x_n]^T$ and $\mathbf{y} = [y_1 ... y_n]^T$. The normalised cross correlation C is given as:

$$C = \frac{\mathbf{x^T y}}{\sqrt{\mathbf{x^T x} + \mathbf{y^T y}}}, \quad where \quad \mathbf{x^T x} + \mathbf{y^T y} > t^2 \qquad (5)$$

for some threshold t. For n-1 rotational shifts of the $\mathbf{x}$ vector, we obtain n values of C, which yields a normalised cross correlation signal over n values.

The maximum value of the correlation signal suggests the correct alignment of the IRAD contour pair and we can generate a list of 3D correspondences along the matched pair of IRAD contours, as:

$$\mathbf{x}_q(i) \rightarrow \mathbf{x}_d(j) \quad , i = 1...n, j = i + k, \ modulo(n) \qquad (6)$$

where $\mathbf{x}_q = (x, y, z)_q^T$ is a 3D point on the query surface, $\mathbf{x}_d = (x, y, z)_d^T$ is a 3D point on the dataset surface, n is the number of points on the IRAD signal pair, and k is the rotational shift (in contour steps) required to achieve the peak in correlation.

We compute these rotations using least squares [2][28]. First compute the cross covariance matrix, $\mathbf{K}$ given by:

$$\mathbf{K} = \Sigma_{i=1}^{n} (\mathbf{x_q}(i) - \overline{\mathbf{x}}_\mathbf{q})(\mathbf{x_d}(j) - \overline{\mathbf{x}}_\mathbf{d})^T \qquad (7)$$

we then compute the singular value decomposition of $\mathbf{K}$ as

$$\mathbf{K} = \mathbf{USV'} \qquad (8)$$

where $\mathbf{S}$ is the diagonal matrix of singular values and $\mathbf{V}$ and $\mathbf{U}$ are orthogonal matrices. The rotation matrix, $\mathbf{R}$, is then given by

$$\mathbf{R} = \mathbf{VU'} \qquad (9)$$

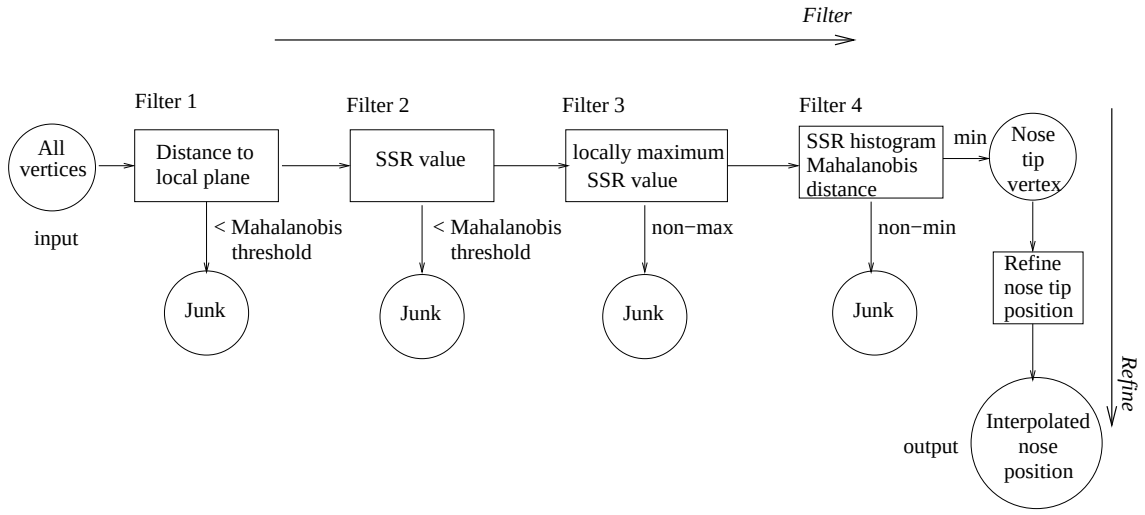
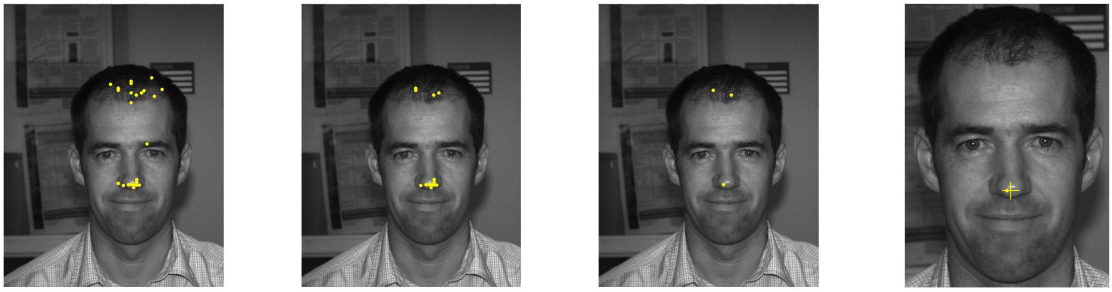


Fig. 12 The cascade filter for nose tip identification (left to right). Also shown is the sub-vertex refinement process (top right to bottom right).



(a) Filter 1 output (b) Filter 2 output (c) Filter 3 output (d) Raw (dot), refined (cross)

Fig. 13 Vertex outputs of the *cascade filter and refine* process for nose tip identification and localisation. 3D vertices have been mapped into the associated registered 2D image for the purpose of visualisation.

In this procedure, the two signals are generally not exactly of the same length and the shorter signal is shifted and correlated across the full length of the longer signal.

5.3.1 Pose checking and refinement

When we are doing one-to-one alignments of 3D face pairs with neutral expressions, we use a pair of complete isoradius contours that fully encircle the nose and we find that the rotation matrix computed in 9 gives good results, which are given in sections 7.1 and 7.2. However, when we use the method to normalise to a canonical pose over large datasets containing facial expressions (see section 7.3). we only use the nose bridge area of an averaged isoradius contour (using 100 3D scans) to reduce the influence of large changes in the lower facial area, such as occurs during movements of the mouth. In this case, we find that it is necessary to do checking and refinement of the rotation matrix. Both of these processes can be implemented by using an average upper face template in conjunction with the RBF model. The average upper face template is a set of 3D points, with a width that spans the outer eye corners an a height that spans from the upper lip area to the eyebrows. The idea is to position this template over the face using the nose tip location and rotation matrix, R, from equation 9, and evaluate the RBF at each point on the template. In general, the set of evaluations will contain both positive and negative values, and we can compute an RMS value representing how well the template fits to the face at that particular rotation (low values mean a good fit). Now, the curvature correlation signal, containing n values (typically 150) of C (equation 5) typically contains 4-6 significant peaks, each of which has an associated rotation matrix. If we compute each of these rotation matrices (instead of just the one with the maximum correlation value), we can select the minimum RMS value as being the best alignment. Fi-

nally, we can refine the rotation matrix using the RBF model, such that it gives a minimum RMS error. This can be achieved by directly computing a point correspondence on the RBF zero-isosurface for each point on the average face template using the following equation:

$$\mathbf{x_{s0}} = \mathbf{x_t} - s(\mathbf{x_t})\frac{\nabla s(\mathbf{x_t})}{||\nabla s(\mathbf{x_t})||} \quad (10)$$

where $\mathbf{x_t}$ is a 3D face template point, $\mathbf{x_{s0}}$ is its corresponding point on the RBF zero isosurface, where $s(\mathbf{x}) = 0$. The set of point correspondences yields a rotation matrix, as previously described, to rotate the average face template and the process can be iterated to yield a refined rotation matrix. This process is a variant of ICP, but there is no requirement to search for correspondences. Rather, they can be computed directly from the RBF, even in areas where the raw face data has missing parts. We find that we only need 3-4 iterations before rotational adjustments fall below 1 degrees, 4-7 iterations to fall below 0.5 degrees and 7-11 iterations to fall below 0.1 degrees. Evaluations of these pose checking and refinement processes are given in section 7.3.

5.4 Pose-normalised depth map generation

Generation of an RBF model has provided mechanisms to localise the nose tip and determine facial orientation. It also provides a futher step, namely a flexible way of generating arbitrary resolution depth maps. The method we use is a gridded coarse-to-fine search for the RBF zero-isosurface. To extract an $n \times m$ depth map, with 8 bit depth resolution, we execute the following procedure.

1. Generate a 3D grid of size $(n \times m \times 17)$, which is sufficiently large to encase all 2.5D head data.
2. Translate the grid so that the nose tip is localised at the centre of (nxm) in the X-Y plane and on the 16th row of the Z plane. (Using the 16th row rather than the 17th gives room for a sign change in the RBF at the nose tip).
3. Rotate the 3D grid about the nose tip using the rotation matrix generated by the IRAD alignment process and any RBF based pose refinements.
4. Use the RBF model to determine (nxm) sign changes in RBF evaluations along the z-dimension (local depth dimension) of the rotated grid.
5. Populate each sign change with another (evenly spaced) 15 RBF evaluations to execute a fine-scale search for the RBF sign change. This gives an equivalent eight-bit resolution i.e. 256 depth possible values.

5.5 Average timing of our processes

We have avoided algorithms with high computational complexity in order to allow a 3D face to be processed in reasonable time. However, our prototype system is implemented in MATLAB and we have emphasized correctness rather than speed optimizations that would be used in a live application. The time to process a face is dependent on the raw data size, the complexity of the surface (for example clothing in the chest and shoulder areas), and parameter settings, such as the size of the size of spherical neighborhoods and the density of spherical sampling in SSR descriptors. In the University of York 3D face dataset, we typically have 5000-10000 useful vertices after the automatic filtering process, which is a similar order of magnitude to FRGC data when downsampled by a factor of 4 (in two directions). To give an idea of the speed of our system, we averaged the processing times over 100 facial scans. The results are as follows: (i) Normals and DLP descriptors (10mm radius neighbourhood): 4.8s; (ii) RBF model fitting: 12.1s; (iii) SSR values 40.7s (128 spherical samples); (iv) SSR value local maxima 0.0003s; (v) SSR histogram generatation (4096 spherical samples) and comparison 6.5s (vi) 30mm isoradius contour extraction (1mm step length) 32.5s (vii) depth map generation (60x90 pixels, 8 bit depth) 9.9s. This gives an average processing time of around 107s per facial scan for our basic one-to-one face alignment process. These times were obtained from a PC with the following specification: AMD Athlon 64x2 Dual core 4200+ 2.20 Ghz, 4Gb RAM, running Windows XP and MATLAB R2006a.

There are two time consuming stages in our process: computatation of SSR values and generation of isoradius contours. The time to compute SSR values is large because there are many nose tip candidates in the DLP filter output, generated from clothing in the chest and shoulder area of the scan. Typically we have to compute around 400 SSR values, but if the face is framed well, this falls to around 100 values, reducing the processing time by 30s.

6 Evaluation of nose tip identification

We have evaluated our RBF derived shape descriptors on both the UoY 3D face dataset and the FRGC 3D dataset. The UoY dataset has 1736 3D faces of 280 different people (subjects) and contains facial expression variations (38% of scans), pose variations (12% of scans) predominantly in the up/down tilt direction, and missing parts, due facial hair, shiny skin and spectacles. The modal mesh resolution in the dataset is around 4mm.

We have found it convenient to split our evaluation into two categories of performance metric, namely: (i) A feature *identification* metric, measured as the percentage of correctly identified nose tip features. This metric measures the performance of SSR shape histograms in a simple classification scheme, when compared to three variants of spin images (see section 6.1 for UoY data, section 6.2 for FRGC data evaluations); (ii) A feature *localisation metric*, measured as the RMS repeatability of the localisation of the nose tip. This metric measures the performance of the SSR value in providing a repeatable nose localisation (see section 6.3 for UoY evaluations only).

6.1 Nose tip vertex identification: UoY data

Examining the filtering stages in figure 12, one might reasonably ask: why not just take the nose candidate outputs from filter 3 (the local maxima of SSR value), compute the Mahalanobis distance to the training set of SSR values and select the minimum distance as the identified nose vertex? This is a good question, because if we can not improve on this nose identification performance, then filter 4 (using balloon images or spin images) is, at best, a waste of processing time and may even be detrimental to the overall identification performance. Therefore, we apply this metric in place of filter 4 as a baseline test (control).

Overall, we have applied five nose identification methods, each of which uses the minimum Mahalanobis distance as the nose identification metric. The training and testing data, however, is different in each case, and is as follows: (1) Baseline test using SSR values. (2) Standard spin images (spin-image type 1), where cylindrical polar coordinates, (r, h), of local vertices are binned. (3) Our own variant of spin image (spin-image type 2), which bins a radius and angle above/below the local tangent plane $(r, tan^{-1}(\frac{h}{r}))$. (4) A spin image which bins $(log(r), h)$ (spin-image type 3). This is often used to give higher weight to closer vertices. (5) SSR shape histograms (balloon images). Our experimental methodology was:

1. A registered bitmap for each of the 1736 images was displayed and a human operator was asked to click their best estimate of the nose tip position using a mouse, and the 2D mouse clicks were stored on disk.
2. Our nose vertex identification process, described by the filters in figure 12, was applied to the dataset, such that we found a set of candidate nose positions (filter 3 outputs), which were locally maximal values of SSR values. Our process uses weak thresholding and hence always finds the nose tip vertex (this was manually verified), but there are typically up to 10 other false positives, which occur on the chin, Adam's apple, shirt collars, quiffs of hair and spectacle frames.
3. We mapped each of these 3D nose candidates into their associated, registered 2D bitmap images and the bitmap position closest to the manual nose click (in step 1), was stored on disk as the correct nose vertex. This allowed us to collect training data for nose features and allowed us to establish a ground truth for the testing phase of nose identification.
4. We randomly selected 100 subjects (of the 280) and for each of these persons, we randomly selected a capture condition to give 100 training 3D images.
5. For each of these 100 training 3D images, we constructed a SSR shape histogram, using 8 radii of 10mm to 45mm in steps of 5mm and 23 bins for normalised RBF values. This gave SSR shape histograms (or balloon images) of dimension 8x23. We also constructed three variants of spin images, as described above. These were constructed to the same resolution as the balloon images, namely 8x23 resolution, using a maximum radius of 45mm and a height of $\pm$ 45mm.
6. We applied principal components analysis (PCA) to all four sets of training data, reducing the shape descriptor dimensionality from 184 to 64.
7. For all nose candidates (filter 3 outputs) on all test images, we calculated the Mahalanobis distance to the trained data for all five methods above. For each test image, the vertex with the minimum Mahalanobis distance was identified as the nose and stored.
8. We then counted, for each of the five methods, what percentage of noses were correctly identified.

In our dataset of 1736 3D images, we used 100 images of 100 individuals as training data, leaving a *test set A*, of 515 3D images, which contains the remaining images of these 100 individuals, not used in the training set, and *test set B*, which contains 1121 3D images of individuals who never appear in the 3D training set.

The results of nose identification are given in table 2. Note that we obtained a 91.7% rate of successful nose identification by using the SSR values. Using SSR histograms improved this figure to 99.6%, whereas use of spin images degraded the system performance to around 70% and hence should be considered unsuitable for the UoY dataset.

There are several reasons why SSR histograms outperformed spin images on the UoY dataset. (i) Spin images require a local normal estimate and this normal varies greatly close to the nose tip, due to the high surface curvature. Any significant error in the local normal

	SSR values		Spin image 1		Spin image 2		Spin image 3		SSR histograms	
Test set	Fails	% Pass	Fails	% Pass	Fails	% Pass	Fails	% Pass	Fails	% Pass
test A (515 images)	48	90.7%	185	64%	153	70.3%	152	70.5%	3	99.4%
test B (1121 images)	93	91.7%	400	64%	316	70.8%	339	70%	4	99.6%

Table 2 Nose identification results using five different methods applied to the UoY dataset

estimate, for example due to sparse data, causes the whole spin image to be corrupted, because the whole spin image is computed relative to this normal. In contrast, the RBF is a global fit significantly influenced by a whole group of normals in the vicinity of the sparse data region. Thus, although a single noisy local normal can locally distort the RBF, we do not encode our descriptor in a local frame relative to this, and so the effect of the noisy normal is contained within a limited region of the SSR descriptor. (ii) The data in our data set has missing parts, particularly around the eyes, when the subject is wearing spectacles. These missing parts corrupt spin images, but have little effect on SSR histograms, because the RBF is defined everywhere in 3D space; (iii) Spin images, in the form used here, use raw vertices and so the data density is a function of the raw mesh resolution. In contrast a SSR histogram can sample the RBF to any required density. (Here we used 512 samples on each of 8 spheres, giving 4096 data elements in each SSR histogram). In order to use spin images effectively on this dataset, we would need to generate a global zero isosurface of the RBF at a sufficiently high resolution. To do this we would evaluate the RBF everywhere on a voxel grid enclosing the full head and then use a 'marching cubes' [39] style of algorithm to find the zero isosurface, alternatively we could use some form of surface following approach. However, global isosurfacing introduces significant additional complexity and processing time.

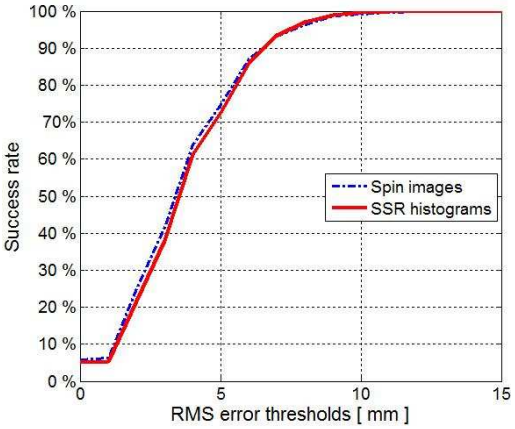


Fig. 14 Nose tip identification performance in the FRGC data for varying thresholds. The performance of SSR histograms and spin images is almost identical

6.2 Nose tip identification: FRGC data

In order to test our nose tip identification method on a significantly larger dataset, we used the FRGC dataset [48] which contains registered 3D shape and 2D intensity (texture) information. Approximate ground truth locations for the nose tip were collected by very carefully manually clicking on enlarged 2D intensity images and then computing the corresponding 3D point using the registered 3D shape information. A dual 2D/3D view was used to verify 2D-3D landmark correspondences and only those with an accurate visual correspondence were retained. This gave us a total of 3780 scans from the 4950 in the dataset and we used 100 of these for training and 3680 for testing. Identical parameters were used in the UoY dataset experimentation, in both training and testing stages.

We gathered results by computing the root mean square (RMS) error of the automatically localised 3D landmarks with respect to the 3D landmarks manually labelled in our ground truth. Remember that localisation is done at the 3D vertex level and we are using a down-sample factor of four on the FRGC dataset, which gives a typical distance between vertices of around 3-5mm. This has implications on the achievable localisation accuracy. We set a distance threshold (specified in millimetres) and if the RMS error is below this threshold, then we label our result as a successful localisation. This allows us to present a performance curve indicating the percentage of successful feature localisations against the RMS distance metric threshold used to indicate a successful location. These results have the nice property that they are not dependent on a single threshold and, in general, these performance curves show two distinct phases: (i) a rising phase where an increased RMS distance threshold masks small localisation errors, and (ii) a plateau in the success rate, where an increased RMS threshold does not give a significant increase in the success rate of localisation. If the plateau is not at 100% success rate, this indicates the presence of some gross errors in landmark localisation. This performance curve is presented in figure 14 and indicates that our system performance is excellent, using either SSR histograms or spin images.

Of course, it is useful to choose some RMS threshold value to quote performance figures. A sensible place to choose the threshold is close to where the graph switches from the rising region to the plateau region, which is around 12mm, indicating that the nose is localised within 3 vertices of the ground truth. This threshold gives a SSR histogram system performance of 99.92% (3 errors) amd a spin image performance of 99.7% (11 errors). We visually observed the three failed cases for the system using the SSR histograms and found that the first fail contained a facial scan with a missing nose, the second selected a vertex within the subject's hair that was nose shaped and the third selected a vertex on the subject's lips due to a non-neutral facial expression.

A valid question to ask is why should we extract an RBF surface model and use RBF based descriptors, if spin images can perform just as well as SSR histograms when the surface data is high quality with no significant areas of missing data due to specular reflections or self occlusions. The answer to this is that the avantages of SSR histograms over spin images is certainly reduced, but the performance of both systems is high as a result of the SSR value descriptor selecting only a small number of candidate vertices for each of these shape histograms to test. For example, if we apply spin images directly to the much larger number of candidates extracted from the 'distance to local plane' (DLP) filter, nose tip identification performance falls below 70%.

6.3 Nose tip localisation refinement: UoY data

To make a preliminary evaluation of our nose localisation refinement (inter-vertex interpolation) approach, we used 80 UoY 3D facial scans in arbitrary poses, each of which had a registered 2D image. We compared our approach both with a simple automatic method and a manual method, in which a user was asked to select a raw 3D coordinate for each of the 80 images, by viewing the surface and rotating it in 3D. In the simple automatic method, the face is rotated through a raster scan of pan and tilt angles within a 45 degree cone and the *nearest point* to the camera acquires a vote. The vertex with the highest number of votes is chosen as the nose coordinate. This is called the NPH (nearest point histogram) method. Our experimental procedure was as follows:

1. Manually locate (by cursor click) three 2D features in the 2D bitmap image: we use the outer corner (*exocanthion*) of the left and right eyes and the midpoint of the upper vermillion line, which is the upper lip's junction with the face (*labiale superius*).

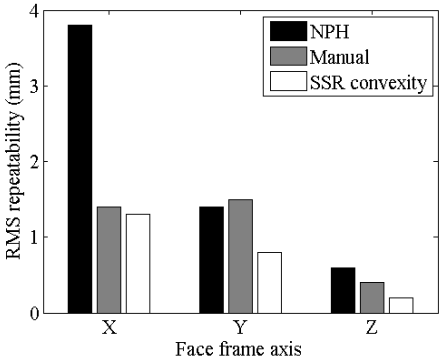


Fig. 15 Nose localisation repeatability RMS(mm) in the three face frame dimensions for the UoY dataset

2. Interpolate to determine the corresponding 3D coordinates, using texture coordinates in the raw 3D file, and use these 3D locations to define a face frame (i.e object centred rather than camera centred frame).
3. Transform the computed nose position from the camera frame to the face frame.
4. Examine the within-class (single subject) repeatability of nose localisation in the face frame, using an RMS metric.
5. Use the average within-class RMS value to compare with the manual method and NPH methods.

The repeatability results of the three methods are given in figure 15. We can clearly see that the NPH method is poor and that our SSR method slightly outperforms the manual method. In part, that is to be expected, since the manual method operates on raw vertices at the original mesh resolution (3-4mm), whereas the nose refinement method interpolates a higher density (2mm resolution) zero isosurface using the RBF model. The results do, however, inspire confidence in the method, and give repeatable results in the presence of noise. Finally, one has to remember that errors in manually locating face frame features and in 2D-to-3D registration appear across all of these results.

7 Evaluation of pose alignment

The evaluation of the isoradius contour (IRAD) method of rotational alignment, in the context of a comparison with ICP, consists of three experiments: (i) How reliably can IRAD/ICP reorientate a facial scan, when that scan is rotationally displaced (synthetically) through a range of angles (0-100 degrees) in the pan, tilt and roll directions. This is a medium scale test using 11 subjects and a total of 660 alignments; (ii) How accurate is IRAD/ICP alignment under real head pose variations of up to 60 degrees? This is a small scale test of

28 alignments and uses manual mark up of eight head poses; (iii) How reliable is IRAD as an alignment mechanism when using a single face template to align a set of faces to a common alignment? This is a large scale test, using both UoY and FRGC data. These three experiments are described and the results are presented in the following three sub-sections.

7.1 IRAD/ICP robustness on synthetic alignment

We have conducted a partly synthetic experiment to illustrate the use of IRAD and ICP in 3D face alignment. The experiment is relatively small-scale (660 alignment experiments) and does not represent a definitive performance of these approaches for face scans, but it does hint at some interesting properties of the algorithms when used in this context. The basic idea is to take a 3D face scan in a frontal pose, rotate it by some angle (0-100 degrees) in some direction (pan, tilt or roll) about the nose tip and then see if IRAD/ICP can re-align the 3D face with the rotated version of itself. This is done for 11 3D images in 5 degree steps across pan, tilt and roll. For each experiment, we determine how many faces are correctly re-aligned, by measuring the RMS error between a set of three reference points.

Firstly, we applied the IRAD method, using a single IRAD of 30mm and we found that the method found the correct alignment in each of the 660 experiments, due to point correspondences being computed explicitly. For ICP we observed, for each experiment, how many faces fail to converge and the number of steps for convergence for those that do. Data points within a spherical neighbourhood (r=54mm) of the nose tip are used to exclude areas of hair, collar and so on.

We apply ICP, such that the nose tips of the two data sets are always locked together, with no translation component allowed (we found that this performed better than standard ICP, where the data means are initially aligned). In this case, ICP computes the rotation matrix (only) that successively minimizes the least squares distance between correspondences. The results are shown in figure 16. Using the overall shape of the graphs in 16, we conclude that ICP performs best in the roll dimension, followed by the tilt dimension and finally, it performs worst in the pan direction. The average number of iterations to reach convergence for the 11 subjects is shown in figure 17. Here we notice that the reverse order in terms of performance, in that the most stable results (roll) take longest to reach convergence, wheras the most unstable are quicker to converge (when successful). It is likely that these results provide an upper initial estimate of the range of angles over which an ICP based facial alignment system could perform,

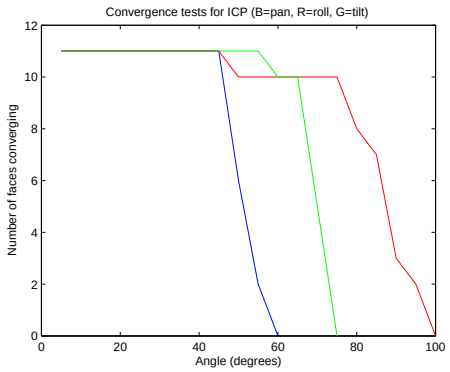


Fig. 16 ICP rotational alignment: Number of faces converging against angle (degrees). Blue=pan, Red=roll, Green=tilt.

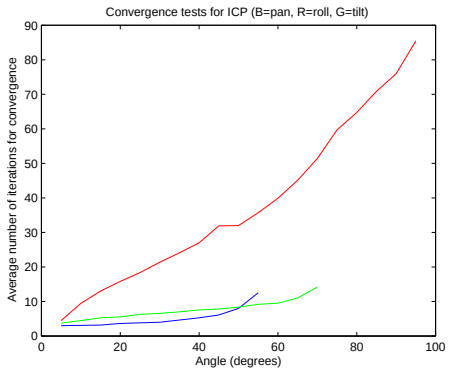


Fig. 17 ICP rotational alignment: Average number of iterations for convergence against angle (degrees). Blue=pan, Red=roll, Green=tilt

because real head pose variations cause changes in the 3D image that are more complex than rigid Euclidean transformations (due to self-occlusion, for example).

7.2 Accuracy test for IRAD/ICP alignment

We now experiment with real head pose variations, rather than synthetic ones, and so the data is subject to self occlusion, such as the nose ocluding the cheek area. In this test, a single subject adopted eight different poses, as indicated in figure 18. Three markers were applied to rigid parts of the face and the centre of these markers was manually clicked, allowing us to localise three 3D coordinates using the known 2D-to-3D registration. This allowed us to compute the rotational (and translational) displacement using three 3D correspondences across any pair of 3D images.

We conducted 28 alignment experiments, one alignment for every pair of 3D images. Firstly the 3D point clouds were aligned by translation, such that both extracted nose tips were coincident. We then rotationally aligned the faces, using the following methods: (i) ICP

Fig. 18 Data used in the pose alignment accuracy test

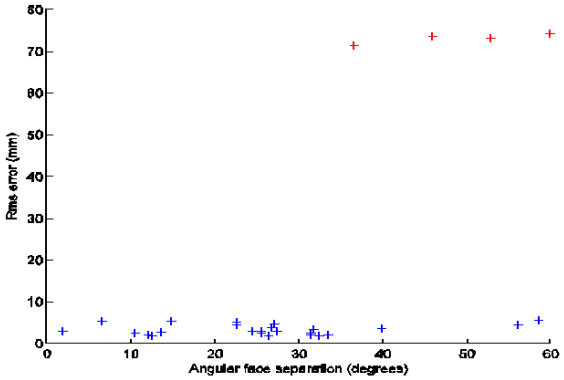


Fig. 19 ICP rotational alignment: residual RMS error (mm) after 20 iterations against initial angular face separation (degrees). Convergence failures are shown in red and occur above 35 degrees

with 20 iterations on a point cloud within a spherical neighbourhood (radius 54mm) of the nose tip; (ii) Isoradius contours using a single extracted 30mm IRAD contour. At the end of each alignment process, we compute the residual RMS error in the alignment of the three 3D marker locations.

Figure 19 shows the results of ICP performance. RMS error is plotted against the angular separation in pose (degrees in an axis-angle formulation), between two 3D images, as measured by the three known 3D correspondences. Clearly, in four of the 28 experiments, ICP has failed, and it appears that, for this subject, convergence to the incorrect solution can occur for angular separations of over 35 degrees.

Figure 20 shows the RMS error of IRAD based alignment (blue trace) with ICP based alignment (red trace). In the instances where ICP fails, IRAD succeeds, as it has determined accurate 3D correspondences over the pair of 3D images, whereas ICP has not. In the case where ICP is successful, it can be seen that the accuracy performance is very similar.

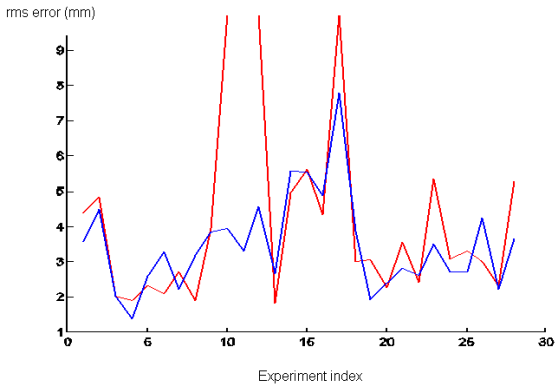


Fig. 20 A comparison of IRAD (blue) and ICP (red) residual RMS alignment error

7.3 Pose normalization: Large scale robustness tests.

Of course, a pair of IRAD signals is going to have a sharp, high correlation peak if they are generated from the same subject. In this sense, we can see that our basic method is highly useful for one-to-one pose alignment and matching, particularly when IRADs in a large 3D face dataset can be computed and stored in an off-line batch process, since only the IRAD from live probe data needs to be extracted on-line. However, other recognition approaches do not align data on a one-to-one basis, but require a common alignment, derived from a pose-normalization process, for all data. Such methods include the popular sub-space based methods, such as PCA and LDA. To test if the IRAD method was capable of pose normalization to a common alignment for a large 3D face dataset, we conducted large scale robustness tests using both UoY and FRGC data.

For every 3D scan in both UoY and FRGC datasets, a single isoradius contour was generated, using an intersecting sphere of $R = 30mm$ from the localised (RBF interpolated) nose tip. One hundred of these were selected from the UoY dataset and one hundred from the FRGC dataset. These contours and associated curvature signals were cropped to $\pm 16mm$ of a manually marked nose bridge location, allowing average contours and signals to be created for the nose bridge area, one for the UoY dataset and one for the FRGC dataset. The nose bridge area is a rigid part of the face, which, intuitively, should be useful for locking IRAD curvature signals into the correct rotational phase when maximising cross-correlation. In addition, the sets of 100 face scans were used to generate upper face templates, comprising a grid of 3D points for fine alignment, as described in section 5.3.1. Both sets of 100 scans were excluded from the testing phase.

Dataset	Method PN1	Method PN2	Method PN3
UoY	98.3%	96.8%	99.1%
FRGC	94.5%	98.7%	99.6%

Table 3 Pose normalisation success rates. Method PN1 is our standard method using the maximum peak in IRAD correlation signal. Method PN2 selects the best of all IRAD correlation peaks. Method PN3 is the similar to PN2, but additionally allows RBF based pose refinement using an upper face template.

We implemented three variants of pose-normalisation system: in the first, our standard method (PN1), we normalise pose using the largest peak in the IRAD curvature correlation signal. In the second method (PN2), we check the rotations associated with all significant correlation peaks (those which are more than 50% of the maximum local peak, typically 4-6) and select the one that has the minimum RMS of RBF evaluations, where these evaluations are at the 3D points that make up the average upper face template. In the third method (PN3), we allow 10 cycles of RBF based pose refinement, as described in section 5.3.1, and again, we selected the pose with the minimum RMS of RBF evaluations over the points comprising the average upper face template. To evaluate our three methods, we manually marked up the intersection of the IRAD contour with the nose bridge on each 3D scan in both UoY and FRGC datasets and measured the rotational shift error (in millimeters) along the IRAD contour for the correlation peak used to determine the head pose. A threshold of 6mm was used to define a successful pose normalisation (success rates reach a plateau at this threshold level), and our results are given in table 3, showing that method PN3 clearly performs best for pose normalisation.

After pose alignment, 60x90 depth maps with 8 bit resolution were generated, as described in section 5.4. Figure 21 shows a sample of the results from the UoY dataset, for those 3D scans that have a significant initial pose variation from frontal. The top row shows depth maps generated without pose normalisation, the middle row shows depth maps from the 3D scans after IRAD based alignment (methods PN1 and PN2, which produce the same result when both are successful) and the third row shows depth maps from the same 3D scans when additional pose refinement using an upper face template is employed (method PN3). Qualitatively, we feel that our system works best when correcting roll angles, where there is no self-occlusion, then tilt angles, and pan angles are the most difficult, due to the significant self occlusion caused by the nose. In figure 21, we can see that, for the last two scans, the part of the face pointing away from the 3D camera is poorly defined in the aligned depth map. To deal with this, further developments to our system are required, such as PCA based reconstruction of the large areas of missing data, which occur due to self occlusion.

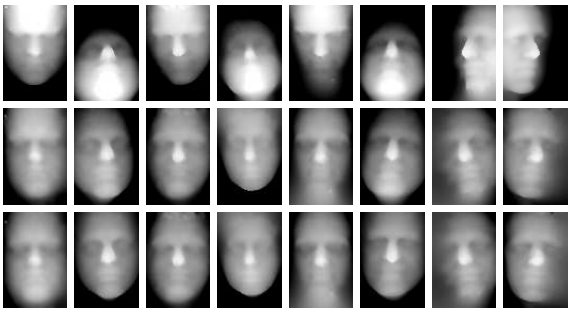


Fig. 21 Sample of UoY depth maps, when the subject is asked to move head 45 degrees relative to frontal pose. The top row shows depth maps in the original pose. The middle row shows pose normalised depth maps without the refinement process (methods PN1 and PN2). The bottom row shows pose normalised depth maps after the refinement process (method PN3)

8 Conclusions

We have presented an RBF-based system to map noisy 3D point clouds to pose aligned or pose normalised depth maps. In doing so, we have developed a system with light viewing constraints that can handle missing parts in a robust way. Several novel 3D pose invariant features have been presented. The first of these is the spherically-sampled RBF (SSR) histogram, which is based on sampling RBFs on concentric spheres, at arbitrary resolutions in 3D space. These representations are pose invariant and they are relatively immune to missing parts, as the RBF is defined everywhere in 3D space. Our experiments on nose vertex identification indicate that these factors appear to be important when characterising high curvature surfaces in the presence of noise and missing parts. We have shown that it is possible to derive an SSR value, which describes the volumetric intersection between a sphere and the object of interest (face), thus providing a useful measure of convexity. A notable issue here is that this feature, in essence, is derived as a summation, which has the effect of suppressing (averaging) noise, where many 3D surface features are based on differencing, whose effect is to amplify noise. The second novel 3D pose invariant feature is the isoradius contour curvature signal, which has been demonstrated to be effective in 3D face alignment. Our future work will focus on developing our methods to deal with extreme poses, such as pure profile facial views.

References

1. M. Ankerst, G. Kastenmüller, H.-P. Kriegel, and T. Seidl. 3d shape histograms for similarity search and classification in spatial databases. In *SSD*, pages 207–226, 1999.
2. K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3d point sets. *IEEE Trans. Pattern Analysis and Machine Intell.*, 9(5):698–700, 1987.
3. J. Assfalg, A. D. Bimbo, and P. Pala. Spin images for retrieval of 3d objects by local and global similarity. In *Proc. 17th Int. Conf. on Pattern Recognition (ICPR'04), volume 3*, pages 906–909, 2004.
4. P. N. Belhumeur, J. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.
5. P. Besl and R. C. Jain. Three-dimensional object recognition. *ACM Computing Surveys*, 17(1):75–145, 1985.
6. P. Besl and N. D. McKay. A method for registration of 3D shapes. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 14(2):239–256, 1992.
7. V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(9):1063–1074, 2003.
8. K. W. Bowyer, K. I. Chang, and P. J. Flynn. A survey of approaches and challenges in 3d and multi-modal 3d+2d face recognition. *Computer Vision and Image Understanding*, 101(1):1–15, 2006.
9. A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Expression-invariant representation of faces. *IEEE Trans. Image Processing*, 16(1):188–197, 2007.
10. J. Carr and W. R. F. amd R. K. Beatson. Surface interpolation with radial basis functions for medical imaging. *IEEE Transactions on Medical Imaging*, 16(1):96–107, 1997.
11. J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. Evans. Reconstruction and representation of 3d objects with radial basis functions. In *Proc. ACM Siggraph 2001*, pages 67–76, 2001.
12. C.Conde, R. Cipolla, L. J. Rodriguez-Aragon, A. Serrano, and E. Cabello. 3d facial feature loction with spin images. In *IAPR Conf. on Machine Vision Applications (MVA'05)*, pages 418–421, 2005.
13. K. I. Chang, K. W. Bowyer, and P. J. Flynn. An evaluation of multimodal 2d+3d face biometrics. *IEEE Trans. PAMI*, 27(4):619–624, 2005.
14. K. I. Chang, K. W. Bowyer, and P. J. Flynn. Multiple nose region matching for 3d face recognition under varying facial expression. *IEEE Trans. PAMI*, 28(10):1695–1700, 2006.
15. C. Chen and E. Prakash. Face personalization: Animated face modeling approach using radial basis function. In *TENCON 2005 2005 IEEE Region 10*, pages 1–6, Nov. 2005.
16. D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung. On visual similarity based 3d model retrieval. *Eurographics 2003*, 22(3), 2003.
17. D. Chetverikov, D. Stepanov, and P. Krsek. Robust euclidean alignment of 3d point sets: the trimmed iterative closest point algorithm. *Image and Vision Computing*, 23(3):299–309, 2005.
18. F. H. Chin-Seng Chua and Y.-K. Ho. 3d human face recognition using point signature. In *4th IEEE Int. Conf. on Automatic Face and Gesture Recognition 2000*, pages 233–238, 2001.
19. C. S. Chua and R. Jarvis. Point signatures: A new representation for 3D object recognition. *Int. Journal of Computer Vision*, 25(1):63–85, 1997.
20. D. Colbry, D. Stockman, and A. Jain. Detection of anchor points for 3d face verification. In *cvpr*, 2005.
21. H. Q. Dinh and S. Kropac. Multi-resolution spin-images. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'06)*, pages 863–870, 2006.
22. C. Dorai and A. K. Jain. Cosmos-a representation scheme for 3d free-form objects. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 19(10):1115–1130, 1997.
23. O. D. Faugeras and M. Hebert. The representation, recognition and locating of 3d objects. *Int. Journal of Robotics Research*, 5(3):27–52, 1986.
24. R. Franke. Scattered data interpolation: Tests of some methods. *Mathematics of Computation*, 38(157):181–200, 1982.
25. T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs. A search engine for 3d models. *ACM Transactions on Graphics*, 22:83–105, 2003.
26. G. G. Gordon. Face recognition based on depth and curvature features. In *Proc. IEEE Computer Society Conf. on: Computer Vision and Pattern Recognition*, pages 808–810, 1992.
27. L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journ. Comput. Phys.*, 73:325–348, 1987.
28. R. M. Haralick, H. Joo, C.-N. Lee, X. Zhuang, V. G. Vaidya, and M. B. Kim. Pose estimation from corresponding point data. *IEEE Trans. Sys. Man. Cybernetics*, 19(6):1426–1446, 1989.
29. T. Heseltine, N. E. Pears, and J. Austin. Three-dimensional face recognition: A fishersurface approach. In *Proc. Int. Conf. Image Analysis and Recognition. LCNS 3212, part II*, pages 684–691, 2004.
30. T. Heseltine, N. E. Pears, and J. Austin. Three-dimensional face recognition: An eigensurface approach. In *Proc. IEEE Int. Conf. Image Processing*, pages 1–2, 2004.
31. T. Heseltine, N. E. Pears, and J. Austin. Three-dimensional face recognition using combinations of surface feature map subspace components. *Image and Vision Computing*, 26(3):382–396, 2008.
32. B. K. P. Horn. Extended gaussian images. *Proceedings of the IEEE*, 72(2):1671–1686, 1984.
33. Q. Hou and L. Bai. Line feature detection from 3d point clouds via adaptive cs-rbfs shape reconstruction and multi-step vertex normal manipulation. In *Computer Graphics, Imaging and Vision: New Trends, 2005. International Conference on*, pages 79–83, July 2005.
34. X. Hu, Y. Tang, and Z. Zhang. Video object matching based on sift algorithm. In *IEEE Int. Conference Neural Networks and Signal Processing*, pages 412–415, June 2008.
35. A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. PAMI*, 21(5):433–449, 1997.
36. I. Kakadiaris, G. Passalis, G. Toderici, N. Murtuza, and T. Theoharis. 3d face recognition. In *British Machine Vision Conference (BMVC'06)*, 2006.
37. M. M. Kazhdan, T. A. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *Symposium on Geometry Processing*, pages 156–165, 2003.
38. R. Kimmel, A. M. Bronstein, and M. M. Bronstein. Three-dimensional face recognition. *Int. Journal of Computer Vision*, 64(1):5–30, 2005.
39. W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, 1987.
40. D. G. Lowe. Object recognition from local scale-invariant features. In *7th IEEE Int. Conf. Computer Vision*, volume 2, pages 1150–1157, September 1999.
41. D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.
42. X. Lu, A. K. Jain, and D. Colbry. Matching 2.5d face scans to 3d models. *IEEE Trans. PAMI*, 28(1):31–43, 2006.

43. A. S. Mian, M. Bennamoun, and R. Owens. An efficient multimodal 2d-3d hybrid approach to automatic face recognition. *IEEE Trans. Pattern Analysis and Machine Intell.*, 29(11):1927–1943, 2007.
44. A. S. Mian, M. Bennamoun, and R. Owens. Keypoint detection and local feature matching for textured 3d face recognition. *Int. Journal of Computer Vision*, 79(1):1–12, 2008.
45. P. Papadakis, I. Pratikakis, S. Perantonis, and T. Theoharis. Efficient 3d shape matching and retrieval using a concrete radialized spherical projection representation. *Pattern Recogn.*, 40(9):2437–2452, 2007.
46. N. E. Pears. Rbf shape histograms and their application to 3d face processing. In *8th IEEE Int. Conf. On Automatic Face and Gesture Recognition (FG'08), Amsterdam, Netherlands*, 2008.
47. N. E. Pears and T. D. Heseltine. Isoradius contours: New representations and techniques for 3d face matching and registration. In *3rd Int. Symposium on 3D Data Processing, Visualization and Transmission (3DPVT'06), University of North Carolina, USA*, pages 176–183, 2006.
48. P.J.Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 947–954, 2005.
49. R. Rohling, A. Gee, L. Berman, and G. Treece. Radial basis function interpolation for freehand 3d ultrasound. In *Information Processing in Medical Imaging*, volume 1613 of *Lecture Notes in Computer Science*, pages 478–483. Springer Berlin/Heidelberg, 1999.
50. D. Saupe and D. V. Vranic. 3d model retrieval with spherical harmonics and moments. In *Proceedings of the DAGM symposium on Pattern Recognition*, pages 392–397. Springer, 2001.
51. V. V. Savchenko, A. Pasko, O. G. Okunev, and T. L. Kunii. Function representation of solids reconstructed from scattered surface points and contours. *Computer Graphics Forum*, 14(4):181–188, 1985.
52. S. Se, D. G. .Lowe, and J. Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *International Journal of Robotics Research*, 21(8):735–758, 2002.
53. M. Segundo, C. Queirolo, O. Bellon, and L. Silva. Automatic 3d facial segmentation and landmark detection. In *Proc. 14th Int. Conf. Image Analysis and Processing*, pages 431–436, 2007.
54. P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The princeton shape benchmark. In *Shape Modeling and Applications*, pages 167–178, 2004.
55. F. Stein and G. Medioni. Structural indexing: Efficient 3-d object recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2):125–145, 1992.
56. T. Theoharis. 3d object retrieval. inter-class vs. intra-class. In *Artificial Intelligence Techniques for Computer Graphics*, pages 55–66. Springer Berlin / Heidelberg, 2008.
57. T. Theoharis, G. Passalis, G. Toderici, and I. A. Kakadiaris. Unified 3d face and ear recognition using wavelets on geometry images. *Pattern Recogn.*, 41(3):796–804, 2008.
58. G. Turk and J. O'Brien. Shape transformation using variational implicit surfaces. *Computer Graphics (Proc, ACM SIGGRAPH 1999*, pages 335–342, 1999.
59. M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.
60. Y. Wang, C. Chua, and Y. Ho. Facial feature detection and face recognition from 2d and 3d images. *Pattern Recognition Letters*, 23(10):1191–1202, 2002.
61. T. Whitmarsh, R. C. Veltkamp, M. Spagnuolo, S. Marini, and F. T. Harr. Landmark detection on 3d face scans by facial model registration. In *1st International Symposium on Shapes and Semantics*, pages 71–75, 2006.
62. C. Xu, T. Tan, Y. Wang, and L. Quan. Combining local features for robust nose location in 3d facial data. *Pattern Recognition Letters*, 27:1487–1494, 2006.