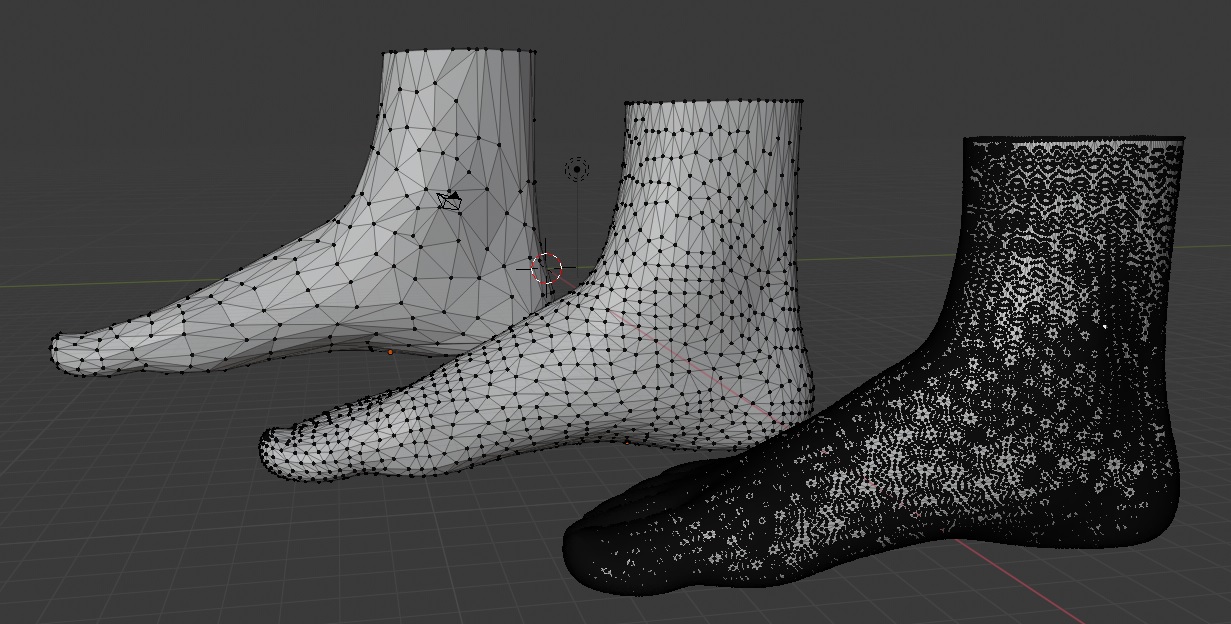
Normalising Geometry

When we scan in foot data, the geometry is somewhat arbitrary. It is dependent on scan points and is both different and random for each model and each scan. This can present some challenges, like too much or too little detail, or detail that isn't proportional to the importance of certain features. It can also lead to unnecessary use of computing resources.

It is important to note that geometry can be independent of model volume and features. The same model can have different geometry, and conversely, the same geometry structure can be used to model different objects.

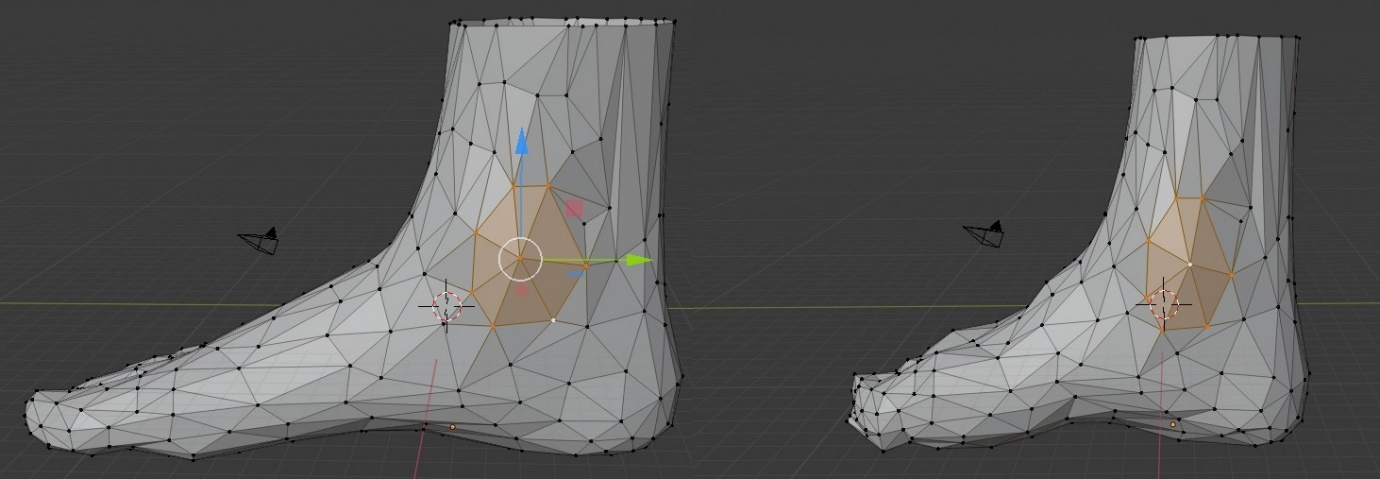
Here is an example of three identical models with different geometries:



I've used different levels of detail here to illustrate that the geometry is different, but it’s the structural topology of the points and edges that can vary despite the underlying model shape being the same.

Because the geometry was different for each scan, I was finding it a challenge to identify exact features. At the same time, I was looking for a way to transition between different models whilst preserving feature integrity and realised a lot of the calculation and transitions would be somewhat trivial if the geometry was normalised, if we applied a standard geometry to all foot models.

Here is an example of two different models with the same geometry:



Notice how in the above example - the ankle protrusion is highlighted, as it would be the same in all models, or at least the same geometry would correspond to that feature. We can actually customise the geometry to better describe particular features.

The geometry would need a high level of detail so it could describe features on many different feet, but we would have direct control of the geometry structure, and be able to give different parts of the feet more or less vertices depending on how important those features were.

These are some of the advantages of having a normalised geometry:

1. Feature detection would be trivial, and essentially come with the model.

2. We could control how detailed and resource heavy the models are.

3. Measurements would be easier, and we could align the geometry with the proportions we want to measure.

4. Animating the growth of the foot over time would be both trivial to implement and aesthetically pleasing.

5. The normalised geometry could even come with details approximated and not included in the scan, like toe geometry under the sock and bone positions.

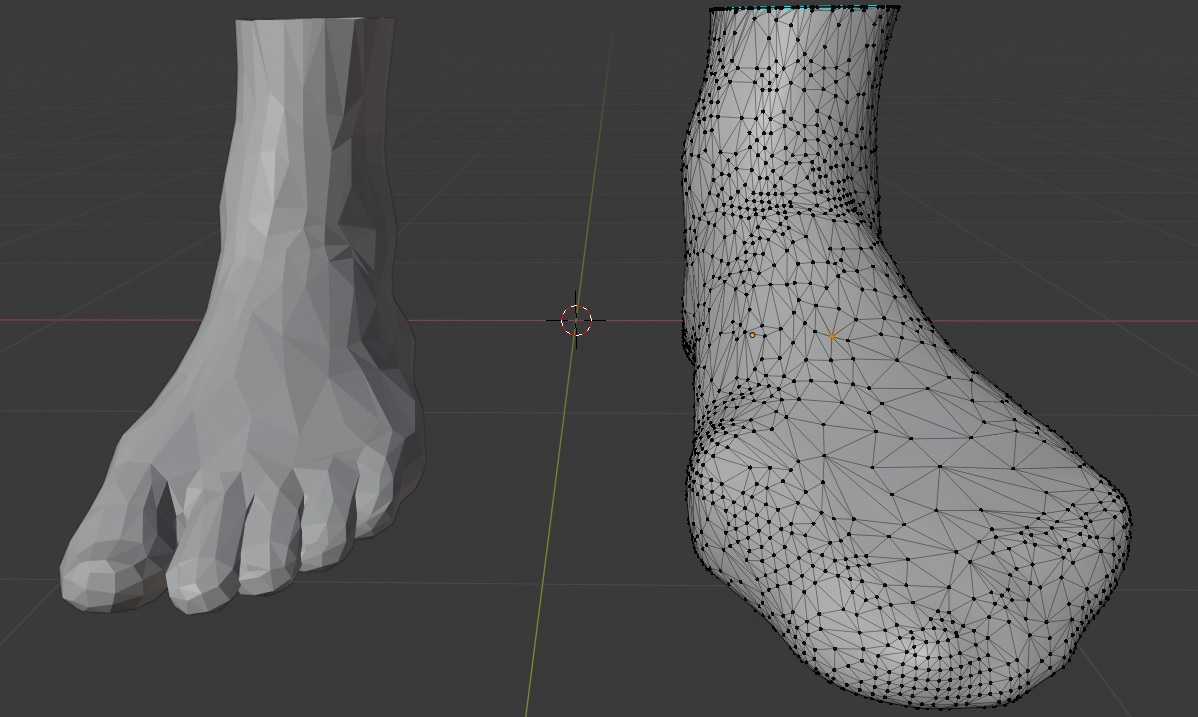
6. If geometry is kept constant, changes in the geometry could be used for other purposes, like having different levels of detail.

7. Comparing foot models would be easier - as each point would correspond with its alias in each model and actually describe useful difference information.

The downside of this approach is the question of how and when to convert/normalise the geometry. Using a normalised geometry may be possible when taking the scan or even help with calibration, it may be best to convert the model on the back end, and deliver the normalised geometry to the front end, or it may be necessary to load in the raw scan and normalise it on the client. I recommend keeping our options open - it's not yet completely clear when it would be best to use the raw scan and when to use a normalised version. It could be that we still need to take measurements before normalising the geometry, and it could well be the case that a normalised version of the foot is only useful when visualising.

Some of the challenges we appear at first to have circumvented would still need to be tackled at the point of conversion, feature detection, for example, would now be a part of the normalisation process, and finding corresponding points and making sure they are standardised for each model would still require an algorithmic approach.

What may help is that we can start with a reference model of an average foot, from which we take the normalised geometry and use that as a guide for aligning with anatomical features. Looking for similarity between a reference model and the scan data will at least give us a head start.



Credit to the following paper for the inspiration:

*(This paper’s focus is on recognition of 3D objects in 2D images, but some of their methods still apply to our work.  
 It was really the wording of the abstract that got me thinking about this approach.)*

NORMALIZATION AND SHAPE RECOGNITION OF THREE-DIMENSIONAL OBJECTS BY 3D MOMENTS

J. M. GALVEZ $ and M. CANTON t Departamento de Fisica Fundamental y Experimental, Centro Superior de Informfitica, Universidad de La Laguna, 38297 La Laguna, Tenerife, Canary Islands, Spain :~ Departamento de Fisica, Facultad de Ciencias del Mar, Universidad de Las Palmas de Gran Canaria, 35017 Gran Canaria, Aptdo. 550, 35080, Canary Islands, Spain

(Received 8 January 1992; in revised.form 12 August 1992; received for publication 18 August 1992)

**Abstract**

*In this paper we are primarily concerned with the recognition of three-dimensional (3D) objects on the basis of the geometry of their physical surface. The 3D moments are defined on the object's surface with two main aims: (1) to derive a normalized version irrespective of position, size, and orientation for each imaged object; and (2) to establish global descriptors, extracted from normalized shapes, in order to make up a representative feature vector to be used in recognition tasks. The source data are multiple orthographic views of 3D objects with known viewpoint specifications. A volume intersection procedure is used in the recovery of 3D object surface from two-dimensional (2D) data. The problem of normalization to achieve recognition of 3D objects is dealt with and a heuristic solution is proposed to lessen the inherent ambiguity of the principal axes method for object orientation. Experimental results are described with ten object categories, showing excellent percentage classification success rates by using only a small number of normalized moments as elements of the feature vector.*

**Keywords:**

*Computer vision, 3D shape recognition, Shape normalization, 3D moments, Surface representation, Shape-from-contour*

I. INTRODUCTION

The recognition of three-dimensional objects is a very important task which needs to be performed in many industrial applications of machine vision and in research areas, such as robotics and computer vision. A suitable way to tackle this problem is to carry it out in two stages. First, the three-dimensional (3D) surface of an object contained in a scene is recovered from a small number of two-dimensional (2D) images. To this end, several approaches were proposed. Stereopsis, motion, and analysis of shading stand in the forefront among the shape-from-x approaches. Shape-from-contour methods have been found to be effective in determining the shape of a visible surface. In the second stage, once the object's surface is properly reconstructed so that it can provide the relevant details, a pattern classification technique is accomplished. Here, the pattern features extracted from recovered surfaces are compared with a stored set of references until a match is found. The matching process is carried out by applying one of several well established similarity measure algorithms. Figure 1 shows a typical block diagram of the two- stage 3D object recognition procedure from multiple 2D views.

In the previous scheme there are two crucial points to take into account. The first point is related to the reconstruction method employed to obtain 3D surface models from 2D data. In this paper the source of shape information consists of the contours of multi- ple views of a 3D object, assuming the orthographic projection approximation in the sensing imaging device. A procedure based on volume intersection presented in papers by Martin and Aggarwa111'2~ has been developed 13~ to derive a surface description, in contrast to the volumetric representation described in reference (2). The primary reason for surface rep- resentation is directly related to the application of shape recognition: that is, 3D object recognition based exclusively on the geometry of an object's surface. The second point is that the application of pattern recognition techniques to shape recognition requires the definition of a suitable normalized feature vector, representative of both the general and the more subtle properties of a given shape. As the primary concern of this paper is shape recognition regardless of spatial position, orientation, and size, the associated feature vector for each 3D object must be normalized against shift, rotation, and scale change transformations. Moment invariants were introduced by HH (4) in 2D pattern recognition as features having the property of being invariant under the aforementioned transform- ations. Such moments were derived, in a limited and reduced number, upon the invariant algebra theory. Satisfactory applications, improvements, extensions and new kinds of normalizations, using 2D moments, have been proposed since then. ~5-16~

The use of 3D moment invariants was first suggested by Sadjadi and Hall, ~ extending the concept of 2D moment invariants to three dimensions. The reduced number of invariants derived (two) may not be sufficient to identify objects having a complex shape. In this paper the 3D moments are defined and evaluated on the object's surface for two specific purposes: (1) to derive a normalized version independent of position, size and spatial orientation for each viewed 3D object; (2) as global descriptors ex- tracted from normalized shapes, to make up the representative feature vector used in 3D object classification.

Full paper continued here: <galvez1993.pdf>