ABSTRACT

This paper presents a technique for creating a 3D object by using 2D images by calibration at different angles. The goal of this study is to investigate the construction of 3D graphical models of real objects in a controlled imaging environment and present the work done based on Eigen values and Eigen vectors. Although many parts of the whole system have been well known in the literature and in practice. Based on a multi-images calibration, an algorithm is designed to extract the common features of the images. Furthermore, this can be extended to estimate robustly the initial Eigen values of the object to be modeled. This algorithm has a simpler visibility check. Besides, in order to construct the appearance, we use the application of images based rendering. The reconstruction results are shown both on real world objects.

Keywords: 3D object modeling, Image based rendering, Eigen values and Eigen vector

1. INTRODUCTION

We present a new approach that facilitates the creation of 3D model directly from 2D images of objects. Our work is motivated by observing the workflow of traditional CAD, where design usually proceeds in many phases. First load data set of an object which is in 2D in MATLAB, often from front, side, except top and bottom view and then uses these images as a reference in constructing a 3D model on a computer. While most of the creative designing occurs in 2D, the process of translating those 2D images into a 3D model is tedious, and often takes some time. The 3D position for every element of the model must be specified, and when referring to images, the user must be constantly translating a desired change in a 2D view to the sequence of 3D operations that will produce that change.

The core idea that makes our approach tractable, from both a user interface and implementation perspective, is to focus on 2D object silhouettes. The user specifies a part by drawing its silhouettes in two or more orthogonal views and a 3D shape that matches those silhouettes is automatically generated and displayed. A user can focus on iterating the 2D images, allowing for fast experimentation and modelling[1].

With the help of an algorithmic standpoint, we show that using silhouettes dramatically simplifies the computation of one of the most challenging features in CAD-CAM[1] and MAYA: Eigen values and Eigen vectors, Constructive Solid Geometry (CSG). Building on ideas from the visual hull and silhouette intersection, we introduce simple algorithms that operate entirely in 2D but can generate complex 3D CSG objects. We also enable the creation of smooth shapes that approximate the least variation-of-curvature surfaces that match their input silhouettes[1] with the extraction of common features.

After loading the data set there is a task to normalize the images to calculate the mean image. Normalization is necessary to reduce the artifacts occurred due to light, after normalization calculate Eigen values if the Eigen value is zero then it can be eliminated because it will plot directly which it find that's why we can ignore it. When all the Eigen values are calculated then it will be sort in ascending order and calculate Eigen vectors which will detect the direction after that calculate the Euclidean distance. After extracting all the values reconstruct the wireframe model.

2. RELATED WORK

In this section, we give a brief overview of the previously proposed methods for create a 3D model by using various techniques and methods with the help of Image based Rendering. During 3D reconstruction solutions, the problem can be simplified by using controlled imaging environments around the object, and background surface and lighting are selected to reduce the specularities on the acquired image. A setup consisting of a rotary table with a fixed camera is generally used in order to obtain a controlled camera motion around the object[1].
The camera has to be calibrated in such a setup to obtain the internal and external parameters defining the physical properties of the camera and the camera imaging positions with respect to the rotary table turn angles. Due to acquisition setup, the rotation axis and distance from the camera center to this rotation axis remain the same during the turns of the table. Based on this idea a vision-based geometrical calibration algorithm have been developed for the rotary table. One of the advantages of this algorithm is that it is more robust than the single image calibration methods[1].

It can compute very easily the distance between the rotation axis of the table with respect to the camera centre which in fact facilitates the calculation of the bounding box. The initial estimation of the bounding box is not an essential step but leads to more efficient computations in the subsequent steps. Keeping the number of voxels constant, the size of the bounding box affects the resolution of the final model: smaller the box, less the quantization effects. There are many disadvantages of the silhouette-based reconstruction algorithm. In this context, they have implemented an algorithm which removes the disadvantages using photoconsistency[1].

In another approach sketch-based modelling is designed by drawing their 2D silhouettes from different views. Complex models can be constructed by assembling them out of parts defined by their silhouettes, which can be combined using CSG operations. Algorithm is used to compute CSG solids that have special properties of silhouette cylinders to convert the 3D CSG problem into one that can be handled entirely with 2D operations, making implementation simpler and more robust. In this designer first sketches an object in 2D, often from front, side, and top views, and then uses these sketches as a reference in constructing a 3D model on a computer. It takes the most time[2].

The user specifies a part by drawing its silhouettes in two or more orthographic views, and a 3D shape that matches those silhouettes is automatically generated and displayed. Silhouettes dramatically simplify the computation of one of the most challenging features in CAD-CAM: Boolean operations, Constructive Solid Geometry (CSG). Basically target is for modeling of man-made objects, as they typically can be decomposed into axis-aligned subparts. Organic models are not well suited to this approach[2].

In another approach, user consider 3D-shape descriptors generated by using functions on a sphere. The descriptors are engaged for retrieving polygonal mesh models. Model can be created either by using the Principle Component Analysis (PCA) or defining features in which the invariance exists, firstly, they define a new rotation invariant feature vector based on functions on concentric spheres. secondly, we compare the two approaches for achieving rotation invariance as well as options to use a single function or several functions on concentric spheres to generate feature vectors. [3]

This algorithms consist of three steps: normalization, feature extraction, and similarity search. During the normalization, a 3D-object is translated, rotated, scaled, and flipped into a canonical position and orientation. [3]

3. PROPOSED METHOD

In our approach, a 3D model is created out of parts. Each part is specified by two silhouettes from front, side views. As the user creates and refines the silhouettes of a part, the corresponding 3D model is automatically generated and displayed.

For example, if the user sketches a triangle in the front and side views, and a square in the top, a pyramid will appear in the 3D view. The calculation is fast enough to provide immediate feedback [1].

A shape with complex or occluded segments is built up by composing it out of multiple parts. Each part has its silhouettes match separately, and our interface makes associations between parts and silhouettes explicit by only allowing the MATLAB to match silhouettes for the currently selected part. One image is match with another image, extract the common features from both of the images [1].

After extracting the common features calculate the mean image, and from the mean image the need to calculate the Eigen values and the Eigen vectors which gives us the directions and then sort all Eigen values in the ascending order then we need to calculate Euclidean distance to create a 3D model.

In this method, we use a man made object that is Teapot having 3 different degree angles : 3º, 5º and 7º around 360º. To construct a 3D model we have many steps:

3.1. Calculate Eigen Values

Eigen values depend on the concepts of vectors and linear transformations.

The eigen values of A are precisely the solutions λ to the equation

\[ A \mathbf{v} = \lambda \mathbf{v} \]

The solutions to this equation are the eigen values \( \lambda_i = a_{i,i} \) (i = 1, ..., n).

Eigen value equation for a matrix A can be expressed as

\[ A \mathbf{x} - \lambda \mathbf{x} = 0 \]

I is the \( n \times n \) identity matrix. This equation is called the characteristic equation. For example, if A is the following matrix (diagonal matrix),
3.2. Calculate Eigen Vectors
The eigenvectors of a square matrix are the non-zero vectors that, after being multiplied by the matrix, remain parallel to the original vector. Eliminate those values whose Eigen value is zero.

Geometric symmetric having less number of features to extract, like sheer & tear and scaling.

3.3. Calculate Euclidean Distance
It is the ordinary distance between two points.

In one dimension the distance between two points on the real line is the absolute value.

\[\sqrt{(x - y)^2} = |x - y|\]

In two dimension if \(p = (p_1, p_2, ..., p_n)\) and \(q = (q_1, q_2, ..., q_n)\) are two points in Euclidean n-space, then the distance from \(p\) to \(q\), or from \(q\) to \(p\) is given by:

\[d(q, p) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}\]

4. RESULTS
This section presents the image of an object which is in 3° difference upto 360°.

In Figure 3 the image of an object show the result which is in 5° difference upto 360°.

In Figure 4 the image of an object show the result which is in 7° difference upto 360°.

These are the results with the different degree images to create a 3D wireframe model.

4.1. Accuracy Graph
We have tried three different angle images to convert 3D model and take different results for transform. 3° and 5° images have good results as compared to 7° images.
4.2. Time graph

We have tried three different angle images to convert 3D model and take different time for transform. 3º images takes more time because of 120 images, 5º images take less time as compared to 3º because of 72 images but in the case of 7º images take lesser time because of 52 images, less number of common features are extracted.

5. CONCLUSION

By discussing the above results we conclude that where the accuracy of 3º & 5º images is greater there the time taken by 3º image is more & 5º images is less as compared to 3º and the accuracy and time taken by 7º images are lesser because of lesser number of images.

REFERENCES
