Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes

Andrew Johnson
Martial Herbert
Outline

- Introduction
- Surface Matching
- Spin Images
- Spin Image Parameters
- Surface Matching Engine
- Object Recognition
- Analysis of Recognition
- Conclusion
Introduction

- Surface Matching is used to recognize an object in a scene to obtain information.
- Surface alignment is used to determine transformation between coordinate systems.
- Two possible coordinate systems for surface representation: viewer-centered and object-centered.
Viewer-centered

• Dependent on surface view
• Easy to construct
• Surface description changes with the viewpoint
• Surfaces have to be aligned before comparison
• Separate representations must be stored for each viewpoint
Object-centered

- Object is described in a coordinate System fixed to the object
- View-independent
- No alignment required
- More compact: single representation
- Finding a coordinate system is tough
Clutter

• Real-world surface data has multiple objects
• Can cause clutter
• Segment the scene into components (object/non-object)
• Object-centered coordinate systems using local features
Occlusion

• Surface data can have missing components
• Alter global properties of surfaces
• Complicate object-centered system construction
• Real-life object-centered representation must be robust to clutter and occlusion
Outline

- Introduction
- **Surface Matching**
- Spin Images
- Spin Image Parameters
- Surface Matching Engine
- Object Recognition
- Analysis of Recognition
- Conclusion
Surface matching representation

- Dense collection of 3D points and surface normals
- Object centered coordinate system
- Descriptive image that encodes global properties
- Independent of transformation between surfaces
Surface representation components

Fig. 1. Components of our surface representation. A surface described by a polygonal surface mesh can be represented for matching as a set of (a) 3D points and surface normals and (b) spin images.
Surface matching

- Match individual points to match complete surfaces
- Localized problems
- Can handle clutter and occlusion without segmenting the scene
- Clutter: no matching points
- Occlusion: will not be searched for
Surface Matching

- 2D images associated with each point
- Local basis at an oriented point on an object surface
- Positions can be described by two parameters
- Create a descriptive image by using these parameters in a 2D histogram
- Apply 2D template and pattern matching techniques
Previous work

- Splash representation
- Point signatures
- 1D representations that accumulate surface information along a 3D curve
- 2D images are more descriptive
- Oriented-point basis is robust
- Techniques from image processing can be used
Ideas in this paper

• Description and analysis of use of spin images
• Multi-model object recognition
• Scenes contain clutter and occlusion
• Localization of spin images by reducing spin image generation patterns
• Using statistical eigen-analysis to reduce image dimensionality
Outline

• Introduction
• Surface Matching
• Spin Images
• Spin Image Parameters
• Surface Matching Engine
• Object Recognition
• Analysis of Recognition
• Conclusion
Spin Images

- An oriented point defines a partial object-centered coordinate system
- Radial coordinate
- Elevation coordinate
- Computed for a vertex in the surface mesh
- The bin that indexes this is incremented
- In the accumulator, dark areas correspond to bins with several projected points
Spin images for three oriented points

Fig. 2. Spin images of large support for three oriented points on the surface of a rubber duck model.
Spin images from different surfaces

- Spin images from two different surfaces for the same object will be similar
- Allowances for noise and sampling variations
- Uniform sampling: linearly related spin images
- Use the linear correlation coefficient to compare
- Similarity measure: $l_2$ distance
Outline

- Introduction
- Surface Matching
- Spin Images
- Spin Image Parameters: Bin Size, Image Width, Support Angle
- Surface Matching Engine
- Object Recognition
- Analysis of Recognition
- Conclusion
Spin Image Generation Parameters

- **Bin Size**: Determines storage size of spin images, averaging in spin images, descriptiveness
  - Set as a multiple of mesh resolution
  - Four times: not descriptive enough
  - One-fourth: Not enough averaging
  - Ideal bin size = Mesh resolution
Bin size

Fig. 3. The effect of bin size on spin image appearance. Three spin images of decreasing bin size for a point on the duck model are shown. Setting the bin size to the model resolution creates descriptive spin images while averaging during point accumulation to eliminate the effect of individual vertex positions. (a) 4x mesh resolution. (b) 1x mesh resolution. (c) 1/4x mesh resolution.
Effect of Bin Size

Match distance: Median Euclidean distance

Lower match distance: better matches
Image Width

- Equal number of rows and columns
- *Image width*
- Image width * bin size = Support Distance
- Amount of space swept by a spin image
- Lower image width, lower descriptiveness
- Lower image width, reduced chance of cluttering
Image width

Image width decreases, match distance decreases

Here, image width = 15
Support Angle

• Maximum angle between direction of the basis of the oriented point and the surface normal of contributing points

• **A** has position and normal \((p_a, n_a)\)

• **B** has position and normal \((p_b, n_b)\)

• B is in the spin image of A if \(\cos(n_a, n_b) < \text{support angle}\)

• Support angle is used to minimize self-occlusion and clutter
Support Angle

Here, support angle = 60 degrees

Lower support angle, lower descriptiveness, higher match distance

Support angle necessary for robustness to clutter and occlusion
Spin Image localization

**Large Support**
- Scene spin-image
- Scene points accumulated
- Model spin-image
- Model points accumulated

**Spin-Image Scatter Plot**
\[ A_s = 180, D_s = 20 \]

\[ \rho = 0.636 \]

**Small Support**
- Scene spin-image
- Scene points accumulated
- Model spin-image
- Model points accumulated

**Spin-Image Scatter Plot**
\[ A_s = 60, D_s = 5 \]

\[ \rho = 0.958 \]
Outline

• Introduction
• Surface Matching
• Spin Images
• Spin Image Parameters
• Surface Matching Engine
• Object Recognition
• Analysis of Recognition
• Conclusion
Surface Matching Engine

- Compare spin images from points on two surfaces
- Establish point correlation
- All spin images from the model are stored in a stack
- The spin image of a vertex on the scene is compared to a best-match
- Point correlation is made between the two points
- Point correspondences are grouped, outliers are eliminated
Fig. 8. Surface matching block diagram.
Outline

- Introduction
- Surface Matching
- Spin Images
- Spin Image Parameters
- Surface Matching Engine
- Object Recognition
  - Spin Image Compression
  - Matching compressed images
- Results
- Analysis of Recognition
- Conclusion
Object Recognition

- Every model can be represented as a polygonal mesh
- Spin images for all vertices are created and stored for every model
- The best match at recognition time leads to simultaneous recognition and localization
- Inefficient: too many bins and a linear growth rate during matching
Spin images of points close to each other will be correlated, as will those that are equal and on opposite sides.
PCA

- The eigenspace is computed using PCA.
- The $l_2$ distance in spin image space is best approximated to the $l_2$ in eigenspace.
- $N$ spin images $x_i$ of size $D$.
- Mean is $x_m$.
- Subtracting the mean from each, to make it more effective,
  \[ \hat{a}_i = x_i - x_m \]
Spin Image Compression

- \( S^m = [\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_n] \)
- Covariance matrix: \( C^m = S^m(S^m)^t \)
- The eigenvectors are obtained from:
  \( \phi^m_i e^m_i = C^m e^m_i \)
- The eigenvectors can be considered spin images (eigenspin images)
- We determine the model projection dimension, \( s \)
- The \( s \)-tuple is the representation
  \( p_j = (\hat{a}_j e^m_1, \hat{a}_j e^m_2, \hat{a}_j e^m_3, \ldots, \hat{a}_j e^m_s) \)
Matching Compressed Spin Images

• A scene must be generated for matching
• To generate the s-tuple
  \[ \hat{y} = y_j - x_m \]
• \( Q_j = (\hat{y}_j e^m_1, \hat{y}_j e^m_2, \ldots \hat{y}_j e^m_s) \)
• The \( l_2 \) distance between the model and scene tuples is used
• We find the closest points and store s-tuples in an efficient closest-point data structure
Results

• Model library of 20 objects

Cluttered scenes were created, followed by mesh smoothing and resampling
Recognition

Seven models were simultaneously recognized.

Models have been closely packed and the scenes include clutter and occlusion.
Fig. 13. Simultaneous recognition of six models from a library of 20 models in a cluttered scene.
Analysis in Complex Scenes

• Experimental verification of performance in scenes with clutter and occlusion

• Experiment:
  – Consider several scene data sets
  – Run the recognition algorithm
  – Interactively measure clutter and occlusion along with success in recognition
Recognition success states

- Model exists and is recognized: true-positive
- Model does not exist, but algorithm recognizes it: false-positive
- Model exists, but algorithm cannot recognize it: false-negative
- No true-negative state in their experiment
Recognition Trial

- Model is placed on the scene with other objects
- Scene with occlusion and clutter is imaged
- The algorithm matches the object to the data
- User then separates the surface patch of the model from the rest of the surface
- Amount of occlusion and clutter are calculated
Analysis

• Occlusion: 1 - (model surface patch area/total model surface area)
• Clutter: 1 - (clutter points in relevant volume/total points in relevant volume)
• 100 scenes were created using 4 models: bunny, faucet, Mr. Potato head and y-split
• Randomly placed models
Analysis

Fig. 15. Recognition states versus clutter and occlusion for compressed and uncompressed spin images.
Results

• Low occlusion: recognition failures are non-existent
• High occlusion: failures are many
• After about 70% of occlusion, spin imaging no longer works well
• Effect of clutter is uniform until a high level of clutter is reached
• Spin image matching is fairly independent of clutter
Results for occlusion

Recognition is high until occlusion is 70%, after which it is no longer successful.
Results for clutter

Spin image matching is independent of clutter
Recognized models

True-positive models: Matching without compression matches more than with compression
Matching Time

- Images with compression match slower than those without compression

- The number of points in the scene influence matching time
Conclusion

• Algorithm handles all general shapes
• Through compression using PCA, the spin image representation is made efficient for large model libraries
• Robust to occlusion and clutter
• Spin images are applicable to 3D computer vision problems
• Can be made more efficient and robust with extensions like automated learning and better parameterizations
Thank You