Feature-Based Part Similarity Assessment

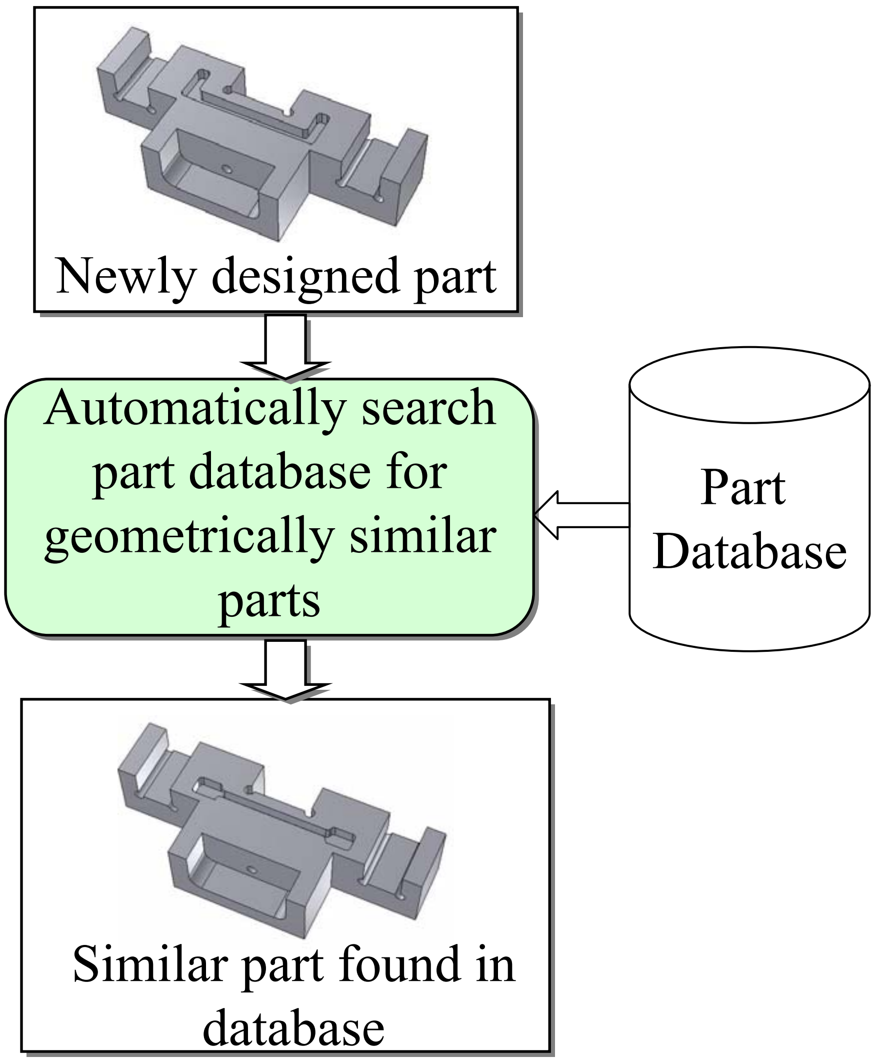
S.K. Gupta
Mechanical Engineering Department and
Institute for Systems Research
University of Maryland
College Park, MD

Students: Antonio Cardone, Abhijit Deshmukh, and Mukul Karnik

Sponsor: Center for Energetic Concepts Development and Naval Surface Warfare Center

Motivation

- Many organizations are archiving 3D geometric models
- Ability to automatically locate geometrically similar parts will be very useful
  - » Considerable reduction in time in locating similar parts
  - » No need for designers to remember file names
- Possible applications
  - » Cost estimation
  - » Tool maker selection
  - » Redesign suggestion generation

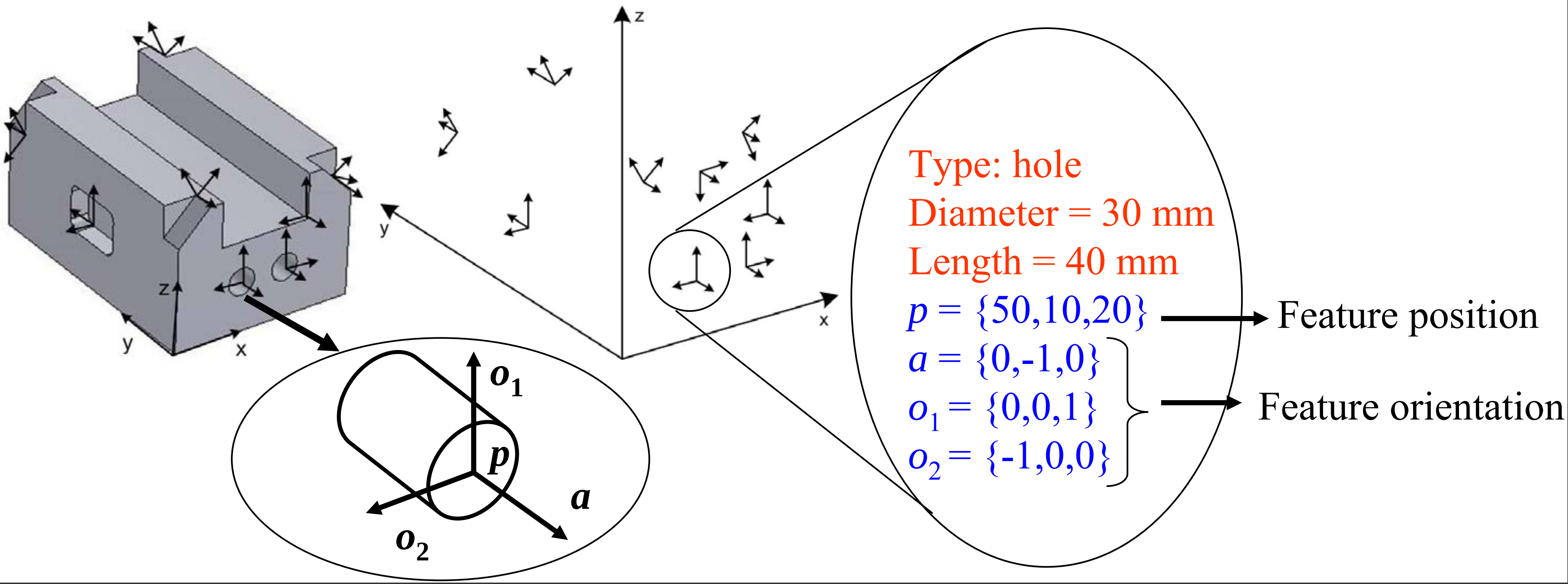


Observations

- Techniques based on gross shape of the parts may not have a good performance in manufacturing applications
  - They may use irrelevant details of the part in comparison
  - They may ignore important details if they are relatively small in size
  - Attributes assigned to geometry such as tolerances cannot be utilized
- Feature-based techniques that ignore feature position and relative orientation may not have a good performance in manufacturing applications

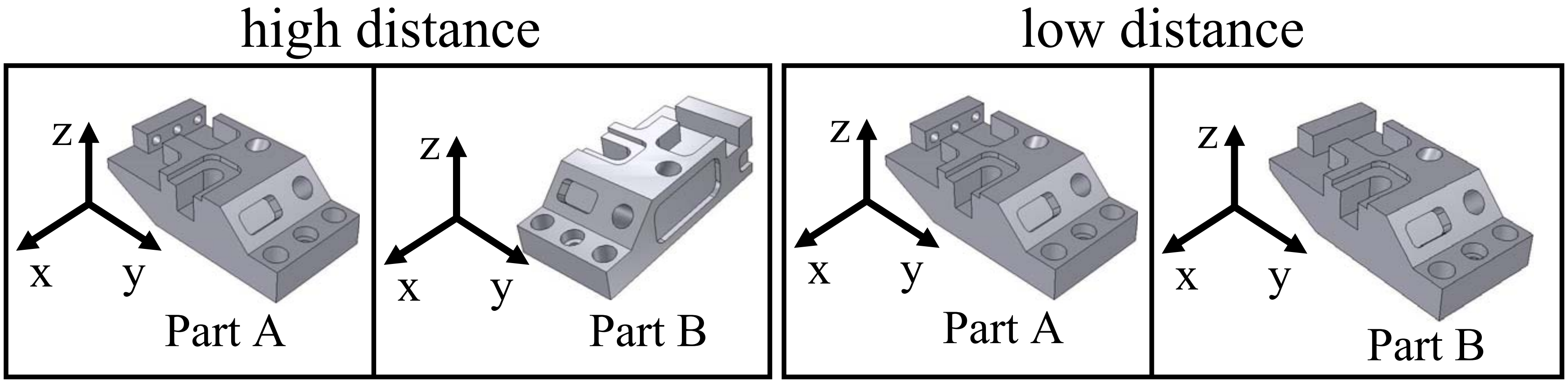
Basic Idea

- Mechanical parts are represented by their shape features
- The shape features of two similar parts should not only be of similar types and sizes but should also be located and oriented in a similar way
  - » This accounts for similarity of spatial feature interactions as well



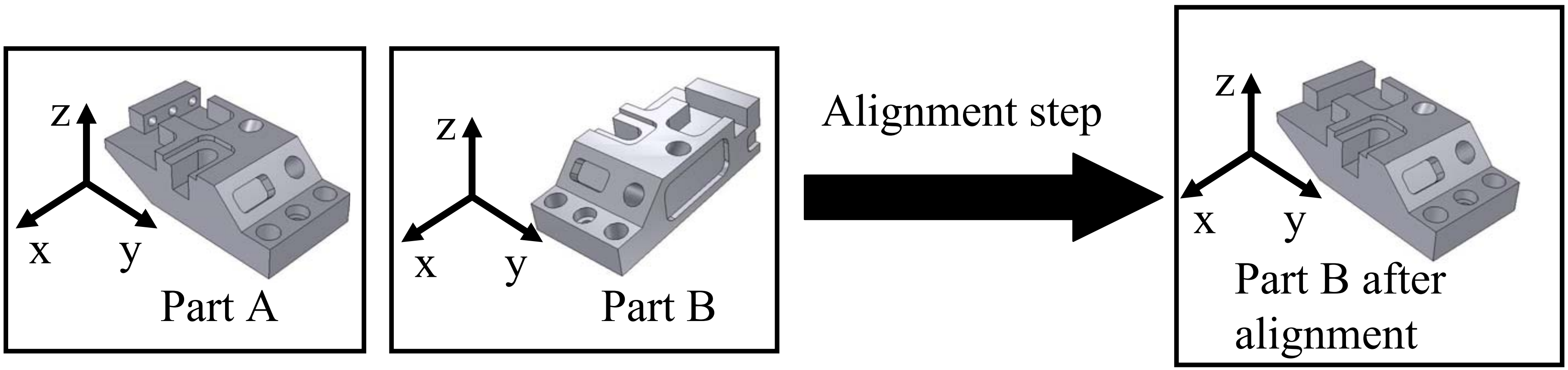
Basic Idea (Continued)

- Similarity can be assessed by computing an *abstract distance function* between two feature sets
  - » Smaller distance means higher degree of similarity
- Distance function should account for relevant feature characteristics – location, type, size
  - » The relative importance of feature characteristics should be determined by user-assigned weights based on application
- Distance value depends on coordinate frames



Basic Idea (Continued)

- To assess similarity correctly, features between two parts need to be aligned
  - » Transform one set of features with respect to others using appropriate transformations



- We are interested in finding transformations that minimize the distance function
  - » Similarity will be assessed using distance resulting from optimal alignment

Overall Methodology

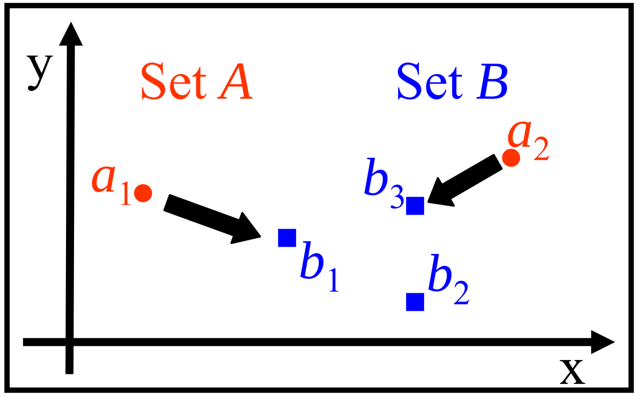
- Develop computational foundations for feature alignment algorithms
  - Develop optimal alignment algorithms based on partitioning of transformation spaces
    - Useful for alignment problems involving lower dimension transformations (e.g., 3 DOF or lower)
  - Develop alignment algorithms based on iterative strategies
    - Useful for alignment problems involving higher dimension transformations (e.g., 4 DOF or higher)
- Develop algorithms for two applications
  - Develop a machining feature-based shape similarity assessment algorithm
  - Develop a surface feature-based shape similarity assessment algorithm

Feature Alignment Problem

- Feature-based shape similarity assessment reduces to the feature alignment problem
- Distance function needs to have the following characteristics
  - » It should consist of a sum of distances between individual features
    - – Distances that are based on worst matches such as Hausdorff one may not perform well in manufacturing applications
  - » It needs to account for feature correspondence
    - – Using closest features results in similar feature distribution in space
- The following general form has been selected

$$D(\mathbf{T}A, B) = \frac{\sum_{i=1}^{n} \min_{b \in B} d(\mathbf{T}a_i, b)}{n}$$

Arrows identify closest neighbor

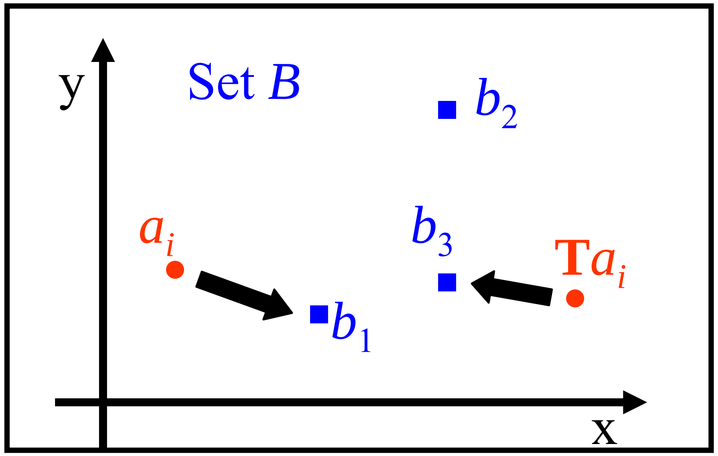


Feature Alignment Problem (Continued)

- General form of distance function

$$D(\mathbf{T}A, B) = \frac{\sum_{i=1}^{n} \min_{b \in B} d(\mathbf{T}a_i, b)}{n}$$

  - » Computing distance functions between two feature sets involves finding the closest neighbor to each feature
- The closest neighbor to each feature changes with the aligning transformation
- Simple enumeration leads to exponentially many possible different closest neighbor combinations



Arrows identify closest neighbor to a_i

- We need a new approach to feature alignment problem

A representative formal problem statement for R^2 using pure translations

- Consider features that are represented as attributed 2D points
- Find the aligning transformation **T**=(Δx,Δy) that minimizes the distance function defined as follows

$$D(\mathbf{T}A, B) = \frac{\sum_{i=1}^{n} \min_{b \in B} d(\mathbf{T}a_i, b)}{n}$$

  - » The distance between feature *a* of set *A* and its closest neighbor *b* of set *B* is defined as follows

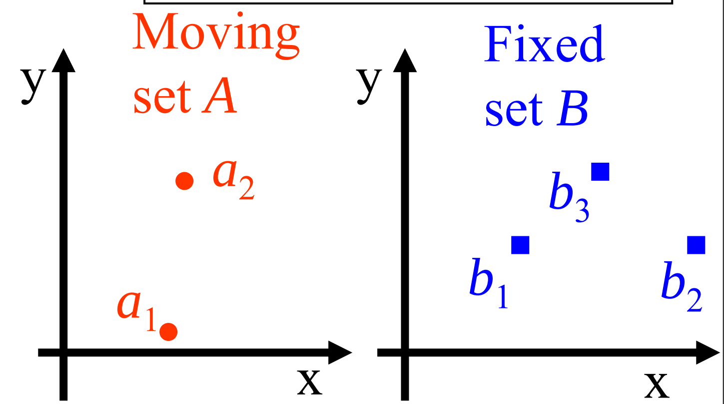
$$d(a,b) = (x_a - x_b)^2 + (y_a - y_b)^2 + \sum_{i=1}^{h} (w_i^a - w_i^b)^2$$

  - » The first two terms account for feature location
  - » The last *h* terms account for feature attributes
  - » The distance function is customizable
    - – Feature attributes can be added and weights can be assigned

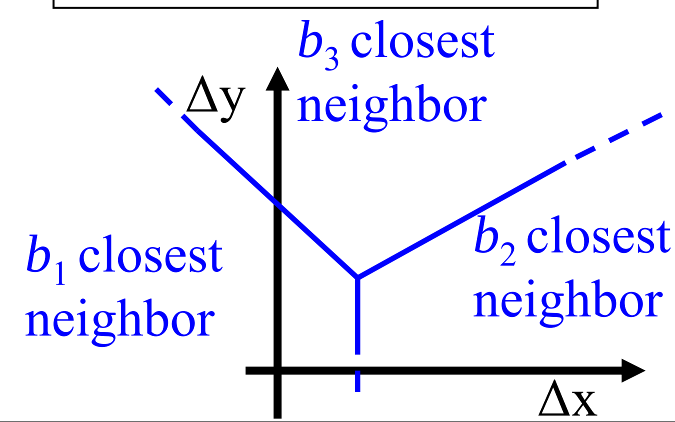
Idea for solving the problem

- If closest neighbors are known then techniques from classical optimization can be used
- Consider a fixed set *B* and moving set *A* with a known **T** applied to *A*
  - » If the transformation is known then nearest neighbors can be easily computed
- Apply a very small transformation to *A*
  - » It is very unlikely that closest neighbors will change
- Closest neighbors only change at certain discrete locations in transformation space
  - » There are regions within transformation space where closest neighbors remain invariant

Feature sets *A* and *B*



2 DOF transformation space for feature a_1



Characteristics of Alignment Algorithms

- We showed feasibility of developing optimal feature alignment algorithms based on partitioning of transformation spaces into closest neighbor invariant regions
  - We developed mathematical foundations for partitioning transformation spaces into closest neighbor invariant regions
  - We showed that the resulting numbers of regions are bounded by a lower order polynomial
  - The distance function used in our algorithms is customizable
    - As many feature attributes as desired can be considered
    - The relative importance of feature attributes is determined by user defined weights

Alignment Involving Higher Dimensional Spaces

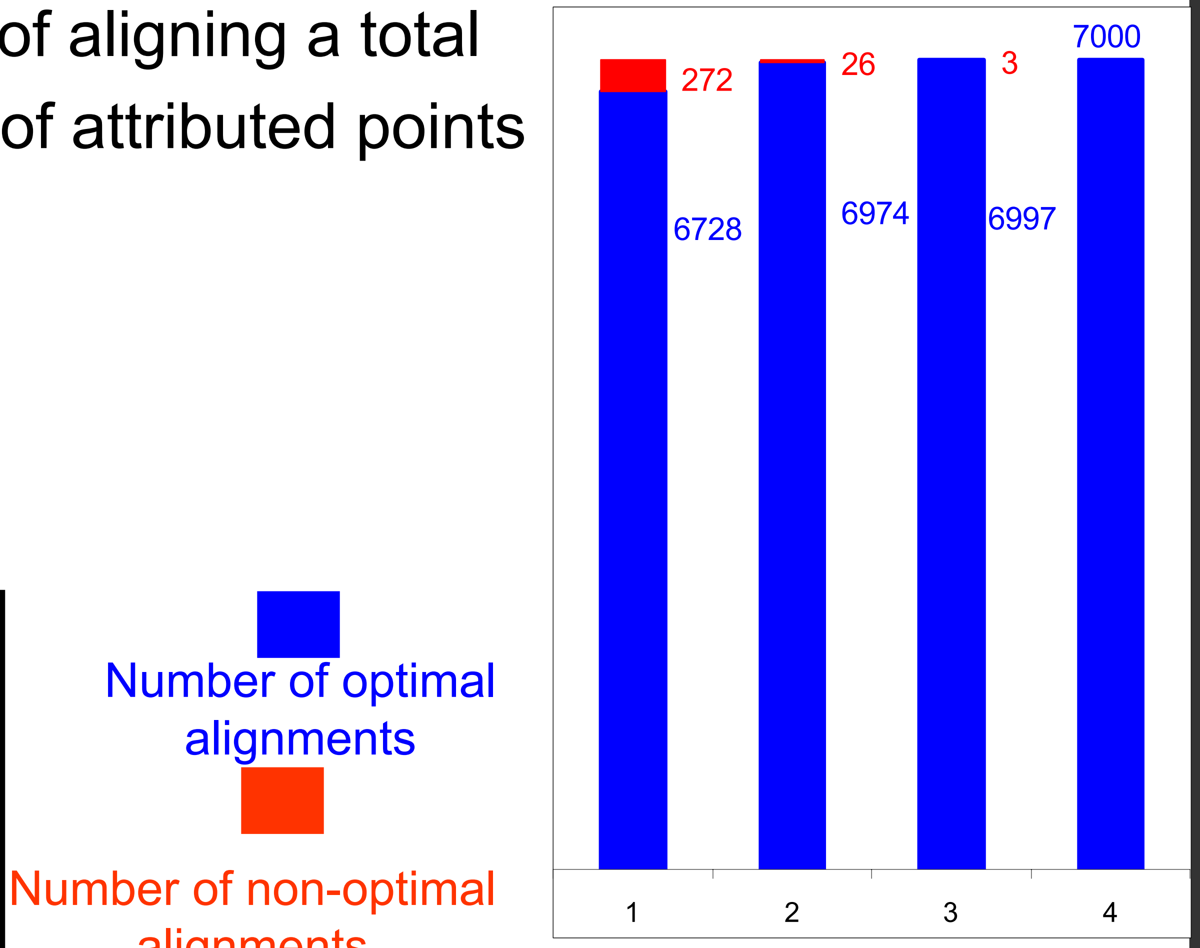
- Implementing partitioning of transformation spaces of higher than three dimensions is difficult
  - » Libraries for higher dimensional geometric entities do not exist
    - – Developing such libraries will be a time-consuming task
- Many problems of practical interests will involve higher dimension transformation spaces
  - » Some of the dimensions can be eliminated by pre-aligning certain attributes of at least one feature pair
- We are interested in exploring iterative strategies for solving higher dimension alignment problems
  - » Transform a higher dimension alignment problem into a sequence of lower dimension alignment problems by fixing certain dimensions
- Key research issues
  - » Will iterative strategies lead to the optimal solution?
  - » How will initial location of feature affect the performance?

Alignment Involving Higher Dimensional Spaces (Cont.)

- Consider two feature sets *A* and *B* and a transformation **T**
- Consider a set of optimal alignment algorithms ALIGN-$\mathbf{T}_i$ based on partitioning of lower dimension transformation spaces
  - » Each lower dimension alignment problem uses a transformation $\mathbf{T}_i$ obtained by fixing certain components of the transformation **T**
- Develop an iterative strategy $\mathbf{S}_i$ that is based on sequences of applications of the optimal alignment algorithms ALIGN-$\mathbf{T}_i$ until the distance between *A* and *B* reaches its minimum value
  - » This requires identifying an appropriate sequence of application and finding an appropriate number of initial conditions

Experimental results for iterative strategy

- Experiments consist of aligning a total of 7000 pairs of sets of attributed points in R^3



Number of initial conditions used

For iterative strategies that use successive translations and rotations we obtained the optimal alignment in all 7000 instances considered by using four initial conditions

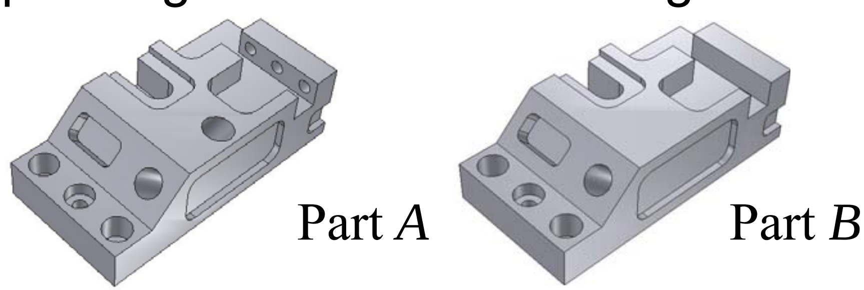
Alignment Involving Higher Dimensional Spaces

- We have developed alignment algorithms based on iterative strategies in R^2 and R^3
  - » The iterative strategies use optimal alignment algorithms based on partitioning of transformation spaces
- Based on the empirical evidence, we can conclude the following
  - » Alignment algorithms based on iterative strategies in R^2 and R^3 can provide optimal solutions for the corresponding alignment problems
  - » The number of initial conditions necessary to obtain the optimal solutions using iterative strategies is low

Machining Feature Application

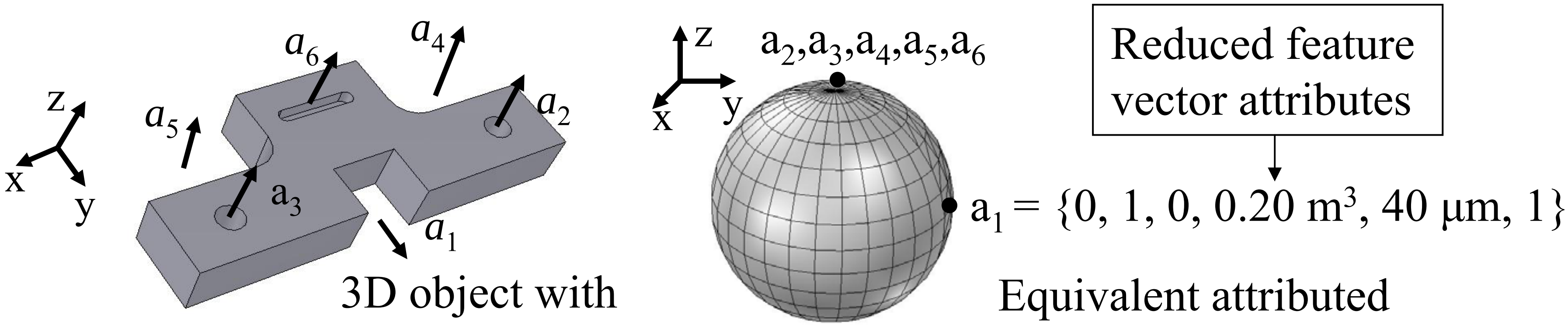
- **Motivation:** Machining features of a part are related to its machining cost
  - » Feature access directions $\Rightarrow$ setup changes
  - » Feature types $\Rightarrow$ tool changes
  - » Feature volumes and tolerances $\Rightarrow$ machining time
  - » Shape similarity assessment based on machining features of parts can be potentially useful to assist human cost estimators
- **Goal:** Build a machining feature-based similarity assessment framework
  - » Define a distance function that accounts for significant machining feature characteristics and develop an algorithm for feature alignment

Parts *A* and *B* are similar in machining features, therefore they are expected to have comparable machining costs



Background

- Parts are represented using reduced feature vectors (RFV), which are the feature components significant from machining effort point of view
- RFVs can be mapped to attributed points on the unit sphere because feature access directions and not feature positions are significant from machining setup point of view



3D object with feature accesses

Equivalent attributed points on unit sphere

- RFV sets are aligned so that distance function is minimized

Problem statement

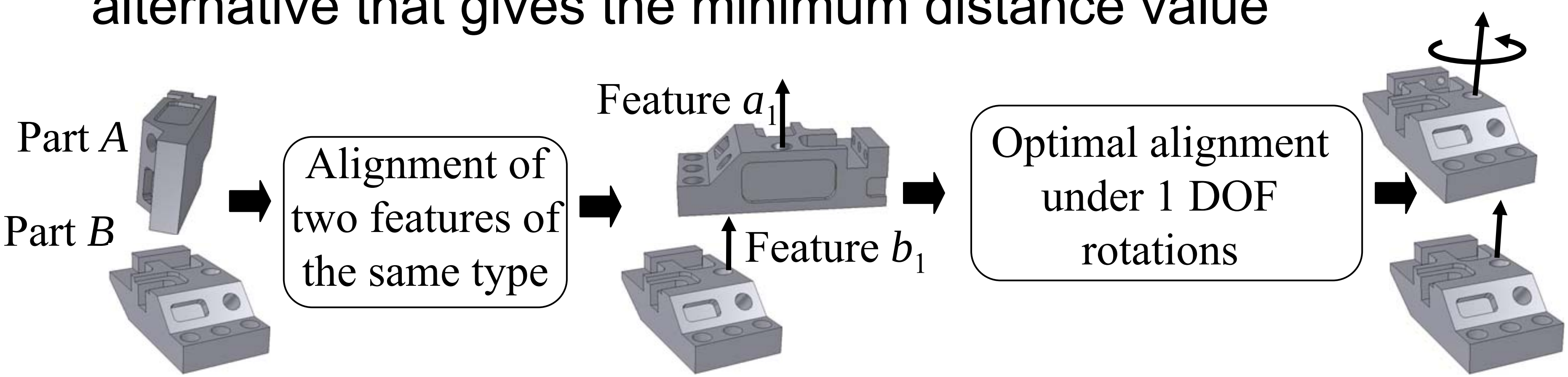
- Find the 3 DOF transformation **T** that aligns feature sets *F* and *F'* to minimize the following distance function

$$D(\mathbf{T}(F), F') = \frac{\sum_{i=1}^{n} \min_{q \in F'} d\left(\mathbf{T}(p_i), q\right)}{n}$$

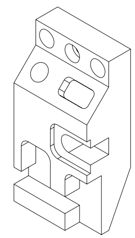
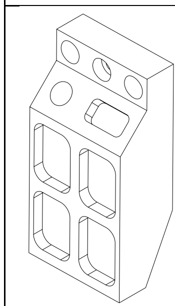
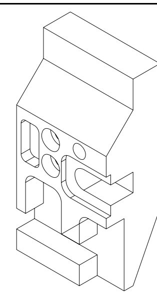
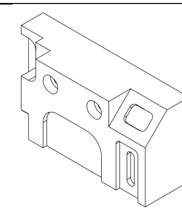
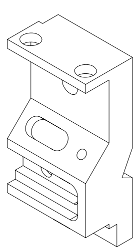
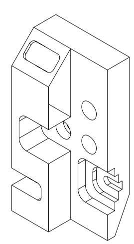
  - » The distance between two features *p* and *q* accounts for the following feature characteristics:
    - – Feature orientation, volume, tolerance, group cardinality, and type
  - » Each feature characteristic is assigned a user-defined weight that determines its relative importance with respect to the other characteristics

Overview of algorithm

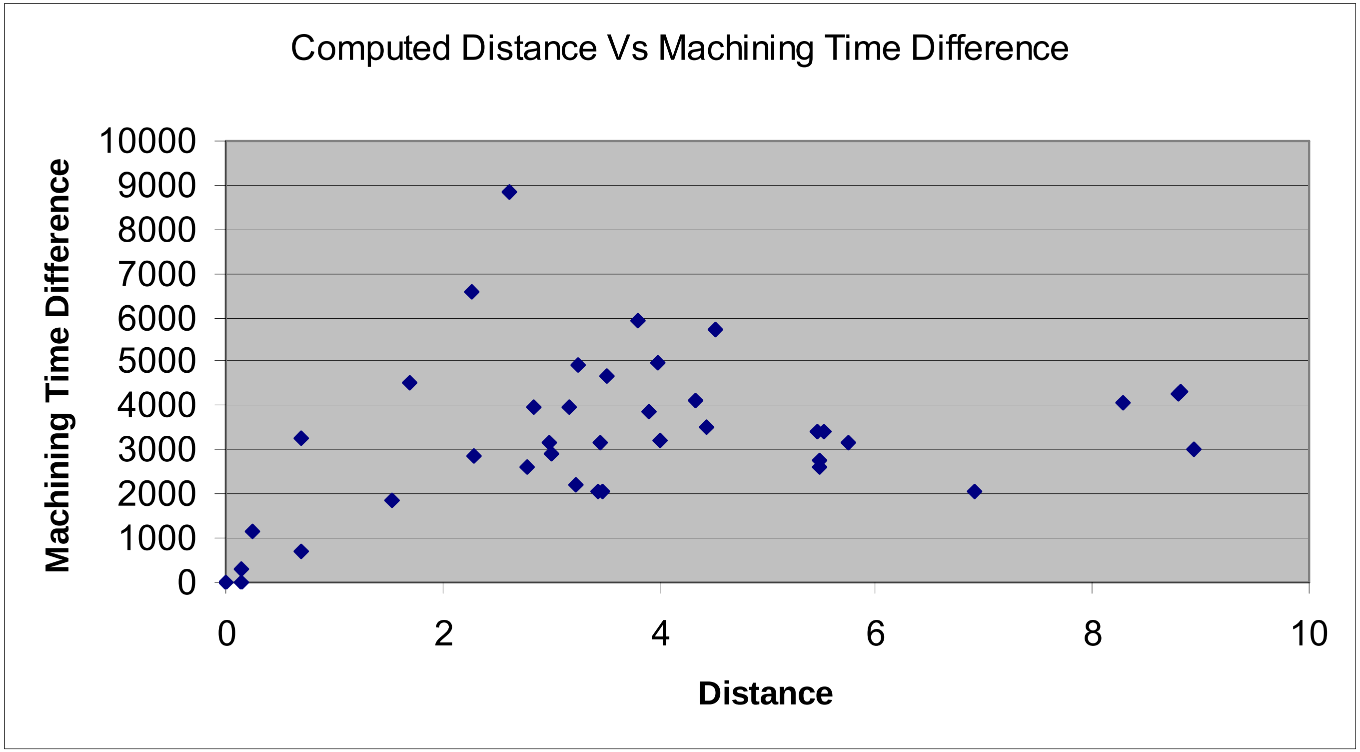
- Only parts that have at least a pair of features whose types match are compared
- Step 1: For each pair of features (*p*,*q*) from *F* and *F'* of the same type
  - » Align *p* and *q* (therefore 2 out of the 3 DOF involved are constrained)
  - » Find the optimal alignment between *F* and *F'* using 1 DOF rotations
- Step 2: Among all the alternatives considered return the alternative that gives the minimum distance value



Example

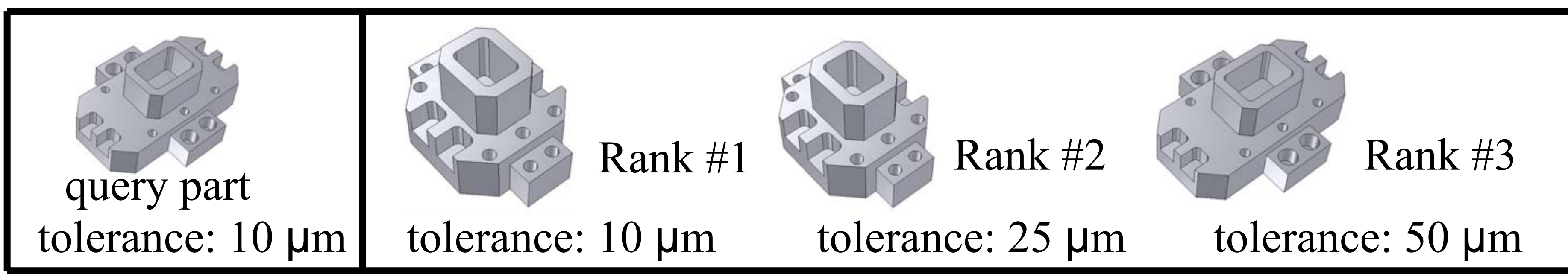
 Query: Part #1		
 Part #20 Distance: 0.148	 Part #15 Distance: 0.149	 Part #14 Distance: 0.245
Rank 1	Rank 2	Rank 3
 Part #13 Distance: 0.696	 Part #16 Distance: 0.697	
Rank 4	Rank 5	

Example (Cont.)



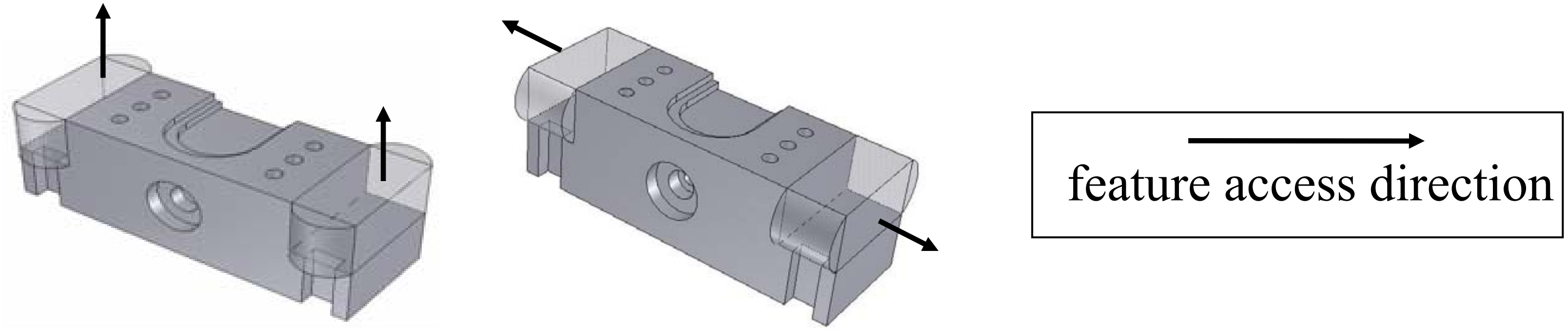
Using Tolerance Attributes

- Database parts are rank-ordered based on their similarity to query part whose feature tolerances are 10 μm
  - » Tolerance and orientation attributes are assigned higher weights than other feature characteristics
- Parts in database were assigned many different tolerance levels
  - » Retrieved parts are similar to query in feature orientation and tolerance

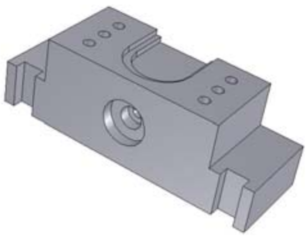


Considering Multiple Feature Interpretations

- Different feature interpretations correspond to different possible ways of machining the same feature
- It is necessary to account for them in performing shape similarity assessment based on machining features

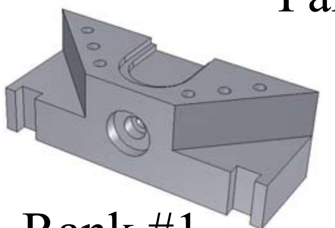


Example

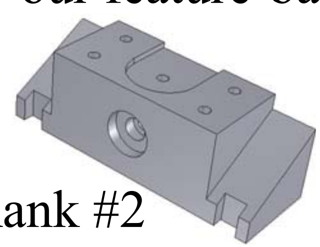


query part

Parts retrieved by our feature-based algorithm



Rank #1

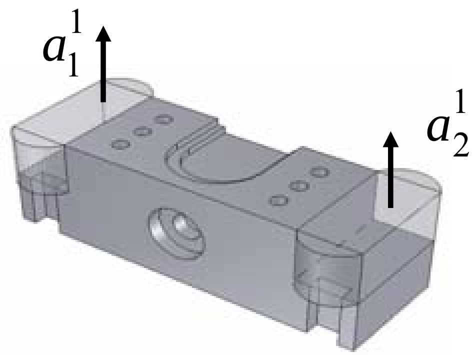


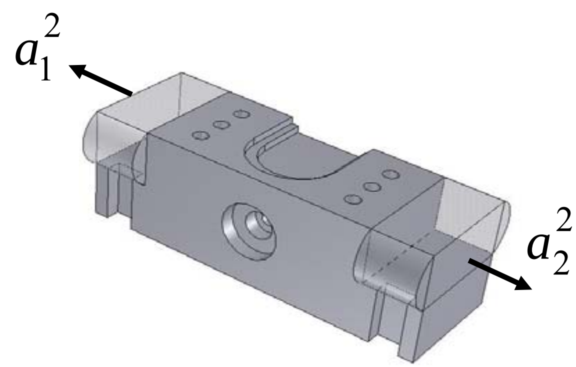
Rank #2

Rank #3

Arrows identify which feature interpretation of query part corresponds to each retrieved part

Query part has two possible interpretations for the features whose corresponding access vectors are shown



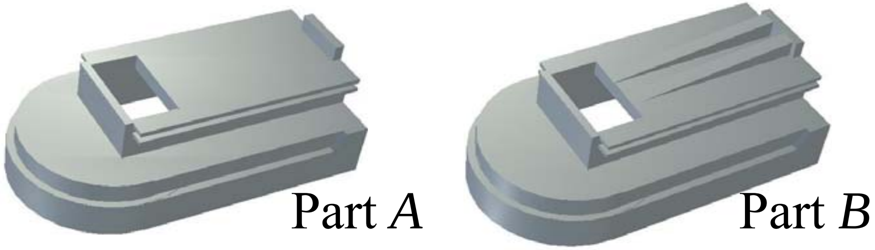


Surface Feature Based Part Alignment

- **Motivation:** Surface features of a part are related to the tooling needed to manufacture it
  - » Surface area ⇨ dimension of tool
  - » Surface position/orientation
  - » Surface type/curvature

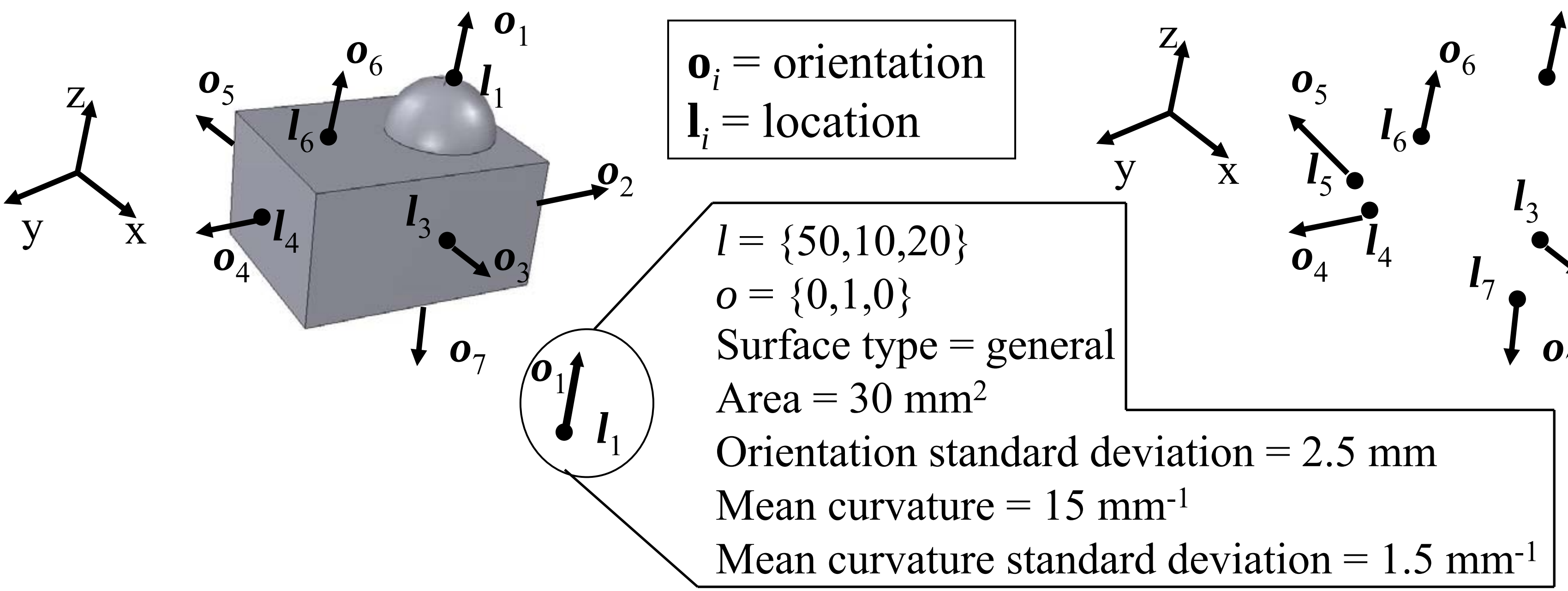
    } ⇨ type and complexity of tool
  - » Shape similarity assessment based on surface features of parts can be potentially useful to assist human tool maker selectors
- **Goal:** Build a surface feature-based similarity assessment framework
  - » Define a distance function that accounts for significant surface feature characteristics and develop an algorithm for feature alignment

Parts *A* and *B* are similar in surface features, therefore they are expected to have similar toolings



Background

- Parts are represented by reduced surface feature vectors (RSFV), which are the surface components significant from tooling point of view
- Map RSFVs to attributed applied vectors in Υ^3



- RSFV sets are aligned so that distance function is minimized

Problem statement

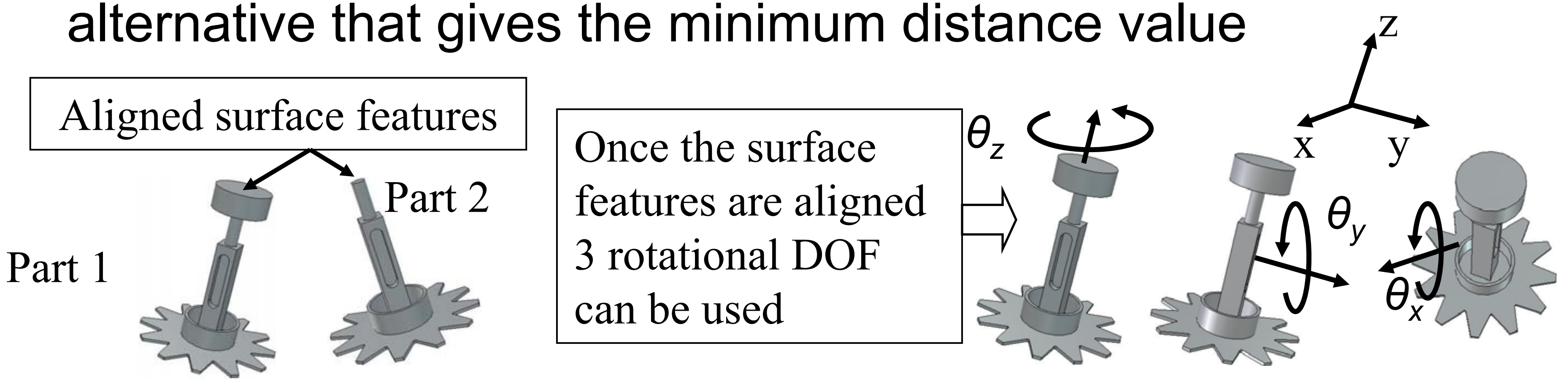
- Find the 6 DOF transformation **T** that aligns feature sets *F* and *F'* to minimize the following distance function

$$D(\mathbf{T}(F), F') = \frac{\sum_{i=1}^{n} \min_{q \in F'} d\left(\mathbf{T}(p_i), q\right)}{n}$$

  - » The distance between two features *p* and *q* accounts for the following surface characteristics
    - – Surface location, area, orientation, curvature, and type
  - » Each surface characteristic is assigned a user-defined weight that determines its relative importance with respect to the other characteristics

Overview of algorithm

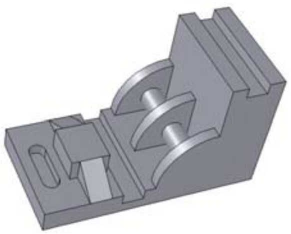
- Only parts that have at least a pair of features whose types match are compared
- Step1: For each pair of features (p,q) from F and F' of the same type
  - » Align positions of p and q (3 out of the 6 DOF involved are constrained)
  - » Find the optimal alignment between F and F' using an iterative strategy under three 1 DOF rotations
- Step 2: Among all the alternatives considered return alternative that gives the minimum distance value



Database and examples

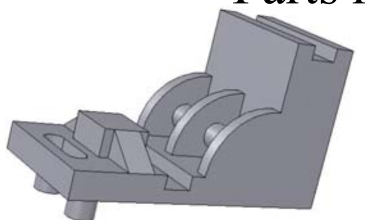
- A database of 1000 parts has been used to test our algorithm
  - The database parts have been collected from real part databases or modeled after real parts
- We compared our algorithm with the Fourier transformation based similarity assessment technique described in [Chakraborty et al. 2005]
- We also tested the customizability of the distance function used

Example #1

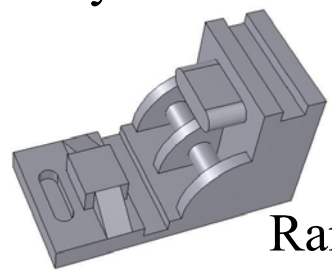


query part *A*

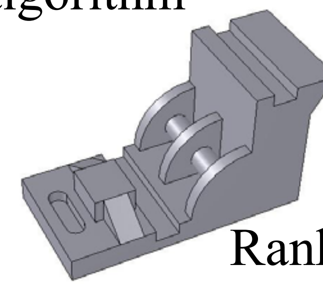
Parts retrieved by our feature-based algorithm



Rank #1
(similar surface features to *A*)

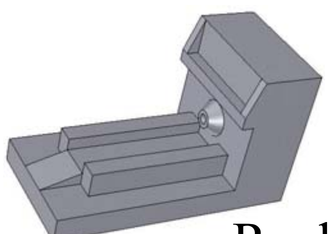


Rank #2
(similar surface features to *A*)

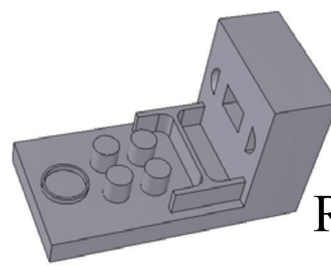


Rank #3
(similar surface features to *A*)

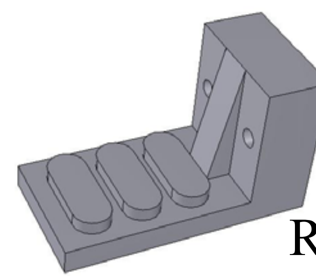
Parts retrieved by the Fourier and Harr based technique



Rank #1
(different surface features than *A*)

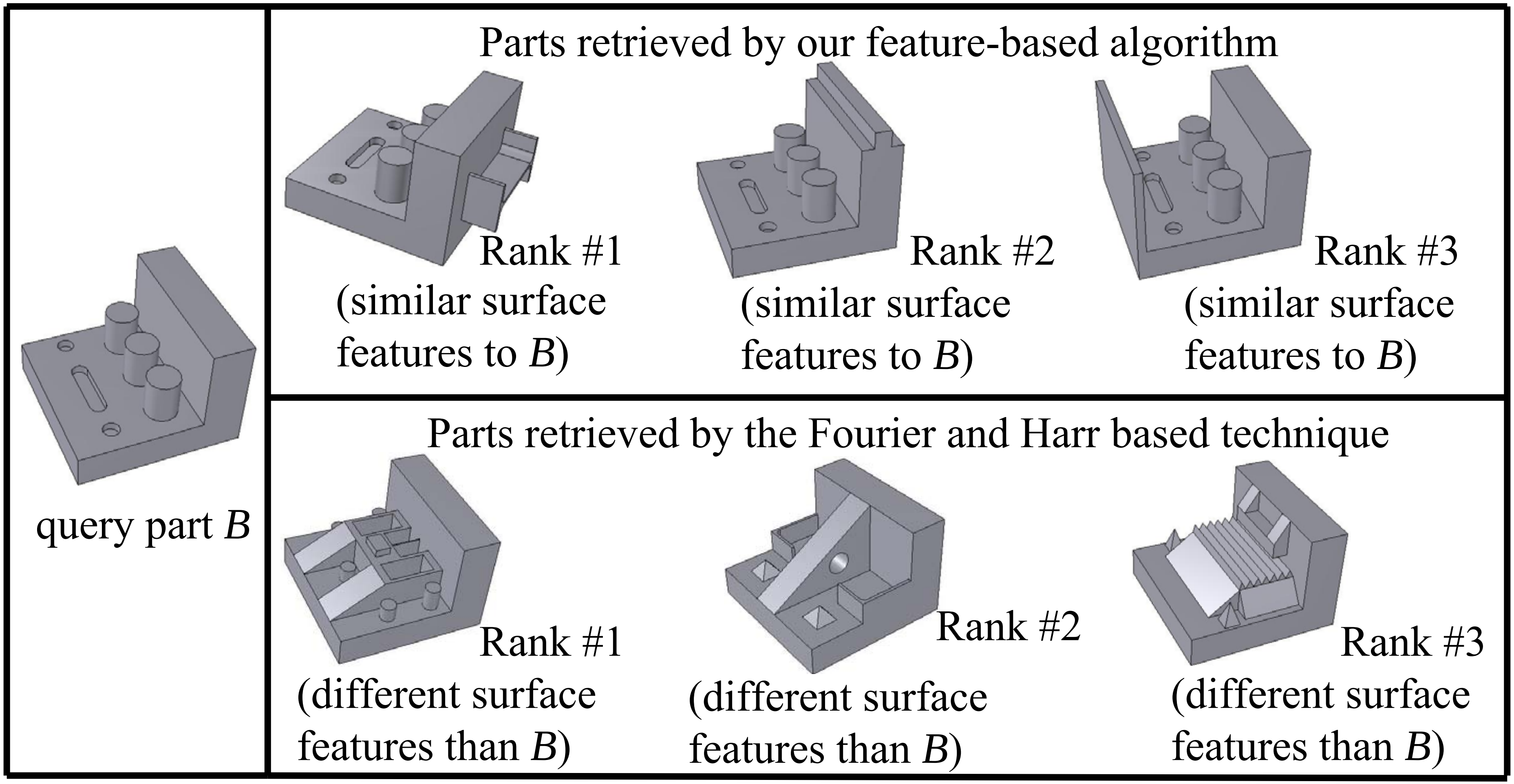


Rank #2
(different surface features than *A*)



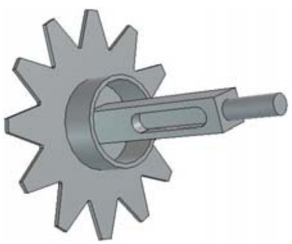
Rank #3
(different surface features than *A*)

Example #2



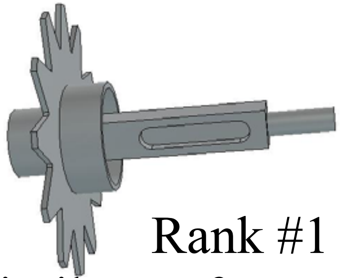
query part B
Parts retrieved by our feature-based algorithm
Rank #1
(similar surface features to B)
Rank #2
(similar surface features to B)
Rank #3
(similar surface features to B)
Parts retrieved by the Fourier and Harr based technique
Rank #1
(different surface features than B)
Rank #2
(different surface features than B)
Rank #3
(different surface features than B)

Example #3

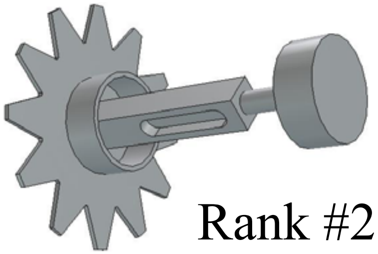


query part *C*

Parts retrieved by our feature-based algorithm



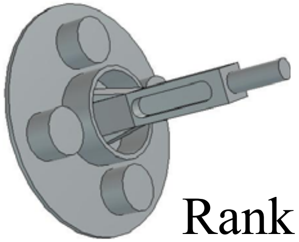
Rank #1
(similar surface features to *C*)



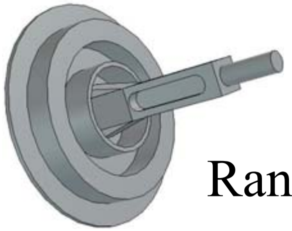
Rank #2
(similar surface features to *C*)

Rank #3
(similar surface features to *C*)

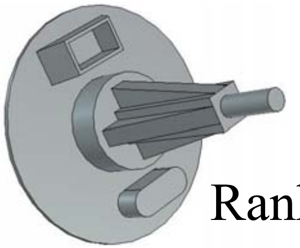
Parts retrieved by the Fourier and Harr based technique



Rank #1
(different surface features than *C*)



Rank #2
(different surface features than *C*)

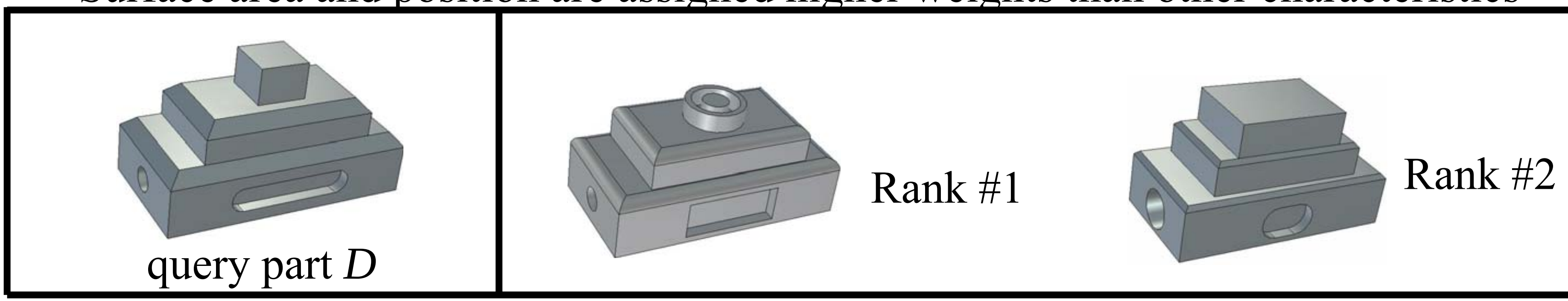


Rank #3
(different surface features than *C*)

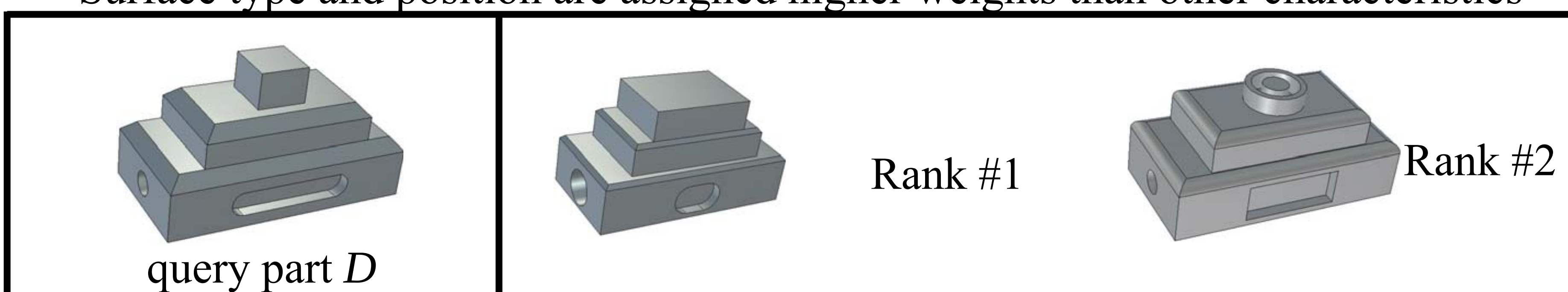
Example showing customizability of distance function

- Database parts are rank-ordered based on their similarity to query part *D*

Surface area and position are assigned higher weights than other characteristics



Surface type and position are assigned higher weights than other characteristics



Research contributions

- Optimal feature alignment algorithms based on partitioning of transformation spaces
  - Algorithms are customizable and have polynomial complexity
- Feature alignment algorithms based on iterative strategies
  - Iterative strategies use optimal algorithms based on partitioning of lower dimension transformation spaces
  - Empirical evidence that they can find optimal solution was provided
- Surface feature-based similarity assessment algorithms
  - The idea works both for explicit as well as implicit feature parameters
- Incorporation of alternative interpretations of volumetric features in similarity assessment algorithms
  - Our algorithms eliminate the need for considering the combinatorial enumeration of various alternative feature interpretations

How to Select a Method?

- Design reuse
  - False positives can be tolerated
    - Designer would be able to sort these out
  - False negatives are not acceptable
  - A suitable gross-shape signature based method will work fine
- Design information reuse
  - False positives are not acceptable
    - Users won't be able to resolve false positives
    - Using false positive can lead to serious problems
  - False negatives can be tolerated
  - A feature based method appears to be a better solution